Managing Affective-learning THrough Intelligent atoms and Smart InteractionS

D4.1 MaTHiSiS sensorial component

Workpackage	WP4 - Affective and Natural Interaction Instruments
Editor(s):	Nicholas VRETOS, CERTH Dorothea TSATSOU, CERTH Dimitrios ANTONARAS, CERTH Martha POLITOU, CERTH Enrique HORTAL, UM Esam GHALEB, UM Dimitris KOUTSOUKOS, UM Evaggelos SPYROU, NCSR Dimitrios SGOUROPOULOS, NCSR
Responsible Partner:	*CERTH*
Quality Reviewers	Robert Houghton (UoN), Georgina Cosma (NTU), Andy Burton (NTU)
Status-Version:	Final – v1.0
Date:	Project Start Date: 01/01/2016; Duration: 36 months Deliverable Due Date: 31/12/2016 Submission Date: 09/01/2017
EC Distribution:	Report, Public

This document is issued within the frame and for the purpose of the *MATHISIS* project. This project has received funding from the European Union's Horizon 2020 Programme (H2020-ICT-2015) under Grant Agreement No. 687772

Abstract:	This deliverable presents the different modalities employed within MaTHiSiS in order to extract learners' affective cues from different sensors, i.e. from depth and RGB cameras, microphones and inertial sensors embedded in mobile devices. The modalities that take advantage of this sensorial input in order to understand the affective state of the user include facial expression analysis, gaze estimation, speech recognition and speech-based affect recognition, skeleton motion analysis and inertia sensor-based affect recognition in mobile devices.
Keywords:	Sensorial Component, affect recognition, facial expression analysis, gaze estimation, speech recognition, speech-based affect recognition, skeleton motion analysis, mobile inertia sensors-based affect recognition
Related Deliverable(s)	D2.3 *Full System Architecture* D2.6 *Framework for impact assessment of MaTHiSiS against LEPOSA requirements* D5.1 *Description of the robotic layer* D5.4 *Description of the mobile layer* D5.7 *Description of the interactive whiteboards layer* D7.2 *MaTHiSiS Platform, 1st release*

Document History

Version	Date	Change editors	Changes
0.1	09/11/2016	Dimitrios ANTONARAS, CERTH Dorothea TSATSOU, CERTH Nicholas Vretos, CERTH	Initial version
0.2	07/12/2016	Dimitrios ANTONARAS, CERTH Martha POLITOU, CERTH Dorothea TSATSOU, CERTH Nicholas Vretos, CERTH	Added CERTH's contribution
0.3	12/12/2016	Dimitrios ANTONARAS, CERTH Martha POLITOU, CERTH Dorothea TSATSOU, CERTH Nicholas Vretos, CERTH Enrique HORTAL, UM Esam GHALEB, UM Dimitris KOUTSOUKOS, UM	Added UM contribution, additional CERTH contribution
0.4	13/12/2016	Dimitrios ANTONARAS, CERTH Dorothea TSATSOU, CERTH Nicholas Vretos, CERTH Enrique HORTAL, UM Esam GHALEB, UM Vaggelis SPYROU, NCSR Dimitrios	Added NSCR contribution, additional UM contribution

Version	Date	Change editors	Changes
		SGOUROPOULOS, NCSR	
0.5	19/12/2016	Dimitrios ANTONARAS, CERTH Dorothea TSATSOU, CERTH Nicholas Vretos, CERTH	Proof-reading, homogenization, last inputs (CERTH)
0.6	21/12/2016	Andy Burton, NTU Georgina Cosma, NTU	Quality Review by tracked changes. Quality Review of Sections 1,2,3.
0.6.5	22/12/2016	Robert Houghton, NTU	Quality Review (tracked changes and internal review sheet).
0.7	29/12/2016	Dorothea Tsatsou, CERTH, Nicholas Vretos, CERTH, Vaggelis Spyrou, NSCR,	Addressed peer review comments.
0.8	05/01/2017	Dimitrios Antonaras, CERTH, Dorothea Tsatsou, CERTH, Enrique Hortal, UM	Addressed peer review comments. Document sent to second review (reviewers: NTU)
0.9	05/01/2017	Dimitrios Antonaras, CERTH, Dorothea Tsatsou, CERTH	Incorporated final review feedback from NTU's second revision (Andy Burton, Georgina Cosma)
1.0	09/01/2017	Dimitrios Antonaras, CERTH, Dorothea Tsatsou, CERTH	FINAL VERSION TO BE SUBMITTED

Table of Contents

List of Tables

List of Figures

List of Acronyms

Abbreviation / acronym	Description
AFEW	Acted Faces Emotion In The Wild
ASC	Autism Spectrum Case
ASD	Accumulated Squared Derivatives
CGDLC	Career Guidance Distance Learning Case
CLS	Cloud-based Learner's Space
CNN	Convolutional Neural Network
DSS	Decision Support System
D-SFA	Discriminant Slow Feature Analysis
D-SIFT	Dense Scale Invariant Feature Transform
EM	Expectation Maximization
GMM	Gaussian Mixture Models
HCI	Human-Computer Interaction
HMA	Human Motion analysis
HoF	Histogram of optic Flows
HoG	Histogram of Gradients
ITC	Industrial Training Case
LSTM	Long Short-Term Memory
MEC	Mainstream Education Case
PA	Platform Agent
PMLDC	Profound and Multiple Learning Disabilities Case
RNN	Recurrent Neural Network
SC	Sensorial Component
SDM	Supervised Descent Method
SD-SFA	Spatially-discriminative Slow Feature Analysis
SFA	Slow Feature Analysis
SIFT	Scale Invariant Feature Transform
SLA	Smart Learning Atoms

Abbreviation / acronym	Description
SVM	Support Vector Machine
S-SFA	Supervised Slow Feature Analysis
U-SFA	Unsupervised Slow Feature Analysis

Table 1: Definitions, Acronyms and Abbreviations

Project Description

The MaTHiSiS learning vision is to provide a novel advanced digital ecosystem for vocational training, and special needs and mainstream education for individuals with an intellectual disability (ID), autism and neuro-typical learners in school-based and adult education learning contexts. This ecosystem consists of an integrated platform, along with a set of re-usable learning components with capabilities for: i) adaptive learning, ii) automatic feedback, iii) automatic assessment of learners' progress and behavioural state, iv) affective learning, and v) game-based learning.

In addition to a learning ecosystem capable of responding to a learner's affective state, the MaTHiSiS project will introduce a novel approach to structuring the learning goals for each learner. Learning graphs act as a novel educational structural tool. The building materials of these graphs are drawn from a set of Smart Learning Atoms (SLAs) and a set of specific learning goals which will constitute the vertices of these graphs, while relations between SLAs and learning goals constitute the edges of the graphs. SLAs are atomic and complete pieces of knowledge which can be learned and assessed in a single, short-term iteration, targeting certain problems. More than one SLA, working together on the same graph, will enable individuals to reach their learning and training goals. Learning goals and SLAs will be scoped in collaboration with learners themselves, teachers and trainers in formal and non-formal education contexts (general education, vocational training, lifelong training and specific skills learning).

MaTHiSiS is a 36 month long project co-funded by the European Commission Horizon 2020 Programme (H2020-ICT-2015), under Grant Agreement No. 687772.

Executive Summary

This document reports on the progress, methodologies followed, technologies developed and first results of the MaTHiSiS Sensorial Component (SC), i.e. Task 4.1, as well as the ground truth data collection and the implementation strategy for the SC within MaTHiSiS. This is broken down for each of the five Sensorial Component modalities, namely facial expression analysis, gaze detection, skeleton motion analysis, audio analysis and mobile-based inertia sensors analysis, while the interconnections between the SC modalities towards multimodal affect recognition are presented (Section 3). Furthermore, the document follows through the strategy and tools developed for gathering ground truth data, in order to train the affect recognition algorithms of WP4 (Section 4), and lastly it details the implementation strategy of the Sensorial Component on the MaTHiSiS platform (Section 5).

SC development follows initial technical and user requirements, driven by the system architecture (Deliverable 2.3 *Full System Architecture*) and the first scenarios developed for MaTHiSiS (Deliverable D2.2 *Full Scenarios of All Use Cases*) respectively. This document (Deliverable 4.1 *MaTHiSiS Sensorial Component*) is a result of an iterative process and will be refined and expanded after gathering and training with real MaTHiSiS data (cf. Section4) and after the evaluation of the proposed solutions in the MaTHiSiS *driver pilots*. The objective is to produce interoperable technical solutions that efficiently extract affect-related features of the MaTHiSiS learners from sensory input, that being image/video, audio and mobile-based inertia sensors.

To this end, this document provides a detailed description of the implemented SC algorithms for the first release of the MaTHiSiS platform for all SC modalities, along with first scientific results. The requirements/specifications in terms of processing of raw sensory data received from the MaTHiSiS Platform Agents (PAs) in the SC and the subsequent SC output homogenization and ingestion strategy for multimodal affect recognition in the MaTHiSiS Decision Support System (DSS) is also described. Lastly, the document presents specifications and implementation of the MaTHiSiS Data Acquisition Tool (DAT), which is going to be the vehicle for collecting training data in real world MaTHiSiS settings, with respect to all MaTHiSiS use cases.

1. Introduction

Effective, hyper-personalised learning within MaTHiSiS relies on the adaptation of parameters in the learning experience, based upon each learner's affective state. That is, the physical behaviour of a learner, expressed by a number of cues in their facial, bodily and vocal expressions, their gaze locality and their physical manipulation of PAs, can shed light to their uptake of the learning process. The role of MaTHiSiS WP4 is to effectively translate this physical behaviour into comprehensive affect states that adaptation will take into account in order to understand each learner's uptake of knowledge during the learning process.

The Sensorial Component (SC) is the basis of the recognition of the learners' affect states. Its goal is to gather (physical) behavioural cues of the learner from the MaTHiSiS Platform Agents (PAs) and apply innovative machine learning techniques in order to interpret them into comprehensive affective cues that tell the story of the learner's uptake of the learning objective(s). Its role is to provide these – fragmented per sensor – affective cues to the MaTHiSiS Decision Support System (DSS), which in turn will provide a holistic overview of the learner's affective state from all sensory resources available.

The Sensorial Component implements state-of-the-art technologies from various fields, spanning from computer vision to artificial intelligence, to extract and represent affect-related features stemming from the learners' face, gaze, body postures, speech and inertia-sensor captured interaction with mobile PAs (Section 2).

The list of affect-related features that the SC is extracting, along with a unification and synchronization schema of the features is discussed in Section 3. Moreover, while part of the SC algorithms is going to be deployed on the MaTHiSiS cloud, pre-processing of the raw sensorial features is going to be initially done on each PA and then communicated to the cloud. This implementation strategy of the Sensorial Component on the MaTHiSiS Platform Agents is reported in Section 5.

Finally, an inherent part of the SC (and all of WP4) is the ground truth data that are going to be used to train the respective machine learning algorithms. Although for most of the SC modalities, public datasets are available, and the first results of developed algorithms reported in Section2, were implemented on these public datasets, the particularities of the MaTHiSiS use cases (wide range of age groups and learners with special needs) cannot be met through these datasets. It is also worth noting that there are no datasets available for data captured through the newest sensors, such as mobile inertial sensors. Therefore, there is a need to capture and train the WP4 algorithms through real-life data, for all types of MaTHiSiS learners and for all use cases. To this end, the MaTHiSiS Data Acquisition Tool (DAT) has been developed. Section 4 analyses the requirements, specification and implementation of the DAT. A conclusion is given in section 6 and references visible in Section 7.

2. Sensorial component modalities

2.1 Facial expressions analysis

The face is often considered as the strongest indicator of human emotions. It displays people's feelings and mood state, from simple spontaneous emotions like happiness and disgust to time-dependent affective expressions like anxiety, boredom and engagement in a current task and/or situation. This allows the person's interaction counterpart to understand their state of mind and adjust its behaviour according to the person's feelings. In the MaTHiSiS context, facial images are one of the data cues that will be captured through the Sensorial Component by means of different types of cameras across devices. Due to its high impact, facial images will play a central role along with other data channels to understand learner's affective states while using the MaTHiSiS system.

2.1.1 Facial expression analysis relative work

Facial expressions are an important component of non-verbal human communication. A facial expression can be physically perceived as one or more specific movements of the muscles of the face. These facial motions indicate a person's spontaneous emotions and time-evolving affective state. By understanding a person's facial expressions, computers or robots can respond appropriately to users' behaviour, which is useful in many applications such as psychology, computer technology, medicine, security and education.

The interest in this research area has attracted a lot of attention in the past twenty five years (1), while in most recent years efforts have increased in many fields, making facial expression recognition a well-researched area (1) (2) (3) (4) (5). An automatic facial expression recognition system consists of four main steps: face detection, face registration, feature extraction and expression recognition (5). Many feature extraction approaches have been proposed, depending on the source of data which can be 2D image, 3D and 4D data, thermal images and multimodal combination of those. Facial expression recognition comprises in most cases a task of classifying raw features, such as the aforementioned muscle movements, extracted from facial images.

Much of the previous work in face-based affect detection has focused on the so-called basic (Eckmanian) emotions: anger, fear, sadness, happiness, disgust, and surprise. The muscle movements necessary to express these primary emotions are found to be universally common (6) (7) (8), which means that there is a high degree of consistency in the facial musculature placement among people around the world. However, in an educational domain context, these basic emotions do not appear frequently, according to a review and meta-analysis of 24 studies (9). Engaged concentration, boredom, confusion, frustration, happiness, and anxiety are the main learning-centred affective states instead of basic emotions (10).

2.1.2 MaTHiSiS Facial expression extraction in the wild - Graph Based Method

In this approach, MaTHiSiS takes advantage of the ability of face representation as a graph. The face is located using points tracing specific areas of the face, which are then used to create a graph. The variation of muscle movement on the face, during the expression of different emotions, leads to different positions of points on the image and generates different graphs. MaTHiSiS algorithm uses this graph variation to predict the different emotions.

The proposed algorithm is illustrated graphically in Figure 1. The MaTHiSiS algorithm takes as input an image, then it detects facial landmarks using the Supervised Descent Method (SDM) (11) technique. These landmarks are used for feature extraction (see section 2.1.2.1), wherein a pre-

trained classifier takes into account the extracted features to make a decision about the emotion portrayed in the given image.

In order to extract facial landmarks, the MaTHiSiS system uses an SDM landmark detector. The classifier used throughout the whole process is a Support Vector Machine (SVM) (12), a widely used multi-class classifier (13), (14), (15). Multi-class SVMs classify test datum instances (features) into one of multiple pre-defined target classes, choosing the class that classifies an instance with the greatest margin from other classes.

The aforementioned procedure will be detailed below.

Figure 1: MaTHiSiS facial expression analysis - Graph Based Method: pipeline

2.1.2.1 Feature Extraction

Facial landmarks are points on specific part of the facial image, which indicate for instance the location of the nose, the eyes, the brows and the mouth within an image. These points are tracked to follow the facial muscles' movements in time. If all facial landmarks are considered as a connected graph, we assume that the density of the graph differs in each facial expression (e.g., the density of connected landmarks around different areas of the graph differs due to an emotion response, differently for each of the examined emotions). Graphs are a very useful mathematical tool that can provide a wealth of information regarding the interrelationships of spatial points – in this case, of the facial landmarks. In order to extract features from these facial landmarks, spectral graph analysis is used, through which a characteristic vector, depicting areas of density in a graph, is extracted. To do so, the Laplacian matrix of the graph is calculated (cf. Formula (1)) and the eigendecomposition problem for the eigenvectors corresponding to the 1st and 2nd greatest eigenvalues is solved. This eigenvector holds information regarding the different density areas of the initial graph. In this modality's case, these areas are the characteristic areas of eyes, mouth and nose, thus, the areas that are more expressive when an emotion response is triggered.

Given a graph, its combinatorial Laplacian matrix can be defined as:

$$L = D - A \quad (1)$$

Where D is the degree matrix and A is the adjacency matrix of the graph computed as:

$$\boldsymbol{A = 1 - e^{\frac{(-|x_i - x_j|)}{d}}} \quad (2)$$

$|\boldsymbol{x}_i - \boldsymbol{x}_j|$ is the Euclidean distance between landmark points, $\boldsymbol{x}_i = (a, b)$ is a landmark point where a, b are the corresponding coordinates on Cartesian coordinate system.

d is a constant depicting the variance of the overall distance between the facial landmarks.

In order to normalize between different facial image scales and sizes, a more robust version of the Laplacian matrix is used, the so-called symmetric normalized Laplacian matrix which can be calculated as:

$$L^{Sym} = D^{-1/2}\, L\, D^{-1/2} \quad (3)$$

Once the symmetric normalized Laplacian matrix is calculated, its eigen-decomposition is considered.

$$L^{Sym} v_i = \lambda_i v_i \quad (4)$$

The corresponding eigenvectors of first and second largest eigenvalues are used as the feature of a specific frame.

2.1.2.2 Facial expressions classification

Support Vector Machines (SVM) are one of the most popular supervised learning models for classification that are used in machine learning. The proposed method uses SVM as a classifier, using the RBF kernel function to firstly undergo a training procedure, where labelled ground truth data is used in order to train the algorithm to classify pre-defined labels, based on the association of labelled data with features (in this case facial landmarks) within the training set. In the case of MaTHiSiS, and since it is an early stage of the project, there are no real-life data sets available yet to properly train the developed algorithms. A first collection of data will be gathered through the Data Acquisition event (cf. Section 4), while a larger collection of data is expected to be gathered at the first piloting phase (driver pilots). Nevertheless, for the first release, a publicly available database was used in order to train and test the first implementation, i.e. the Cohn-Kanade database [13]. This dataset is limited to labelling images with the well-known 6 spontaneous emotions of Ekman (16), as there is no existing dataset trained after time-dependent affective states, such as engagement, boredom, etc. The Eckmanian emotions are Anger, Disgust, Fear, Happiness, Sadness and Surprise. These emotions, according to Ekman, are the most basic emotions that can be expressed through facial expressions. A correlation of the aforementioned emotions with affective states however was retrieved in Russell's Core Affect Framework, (17), allowing a direct mapping of the spontaneous emotions to affect states (Table 2).

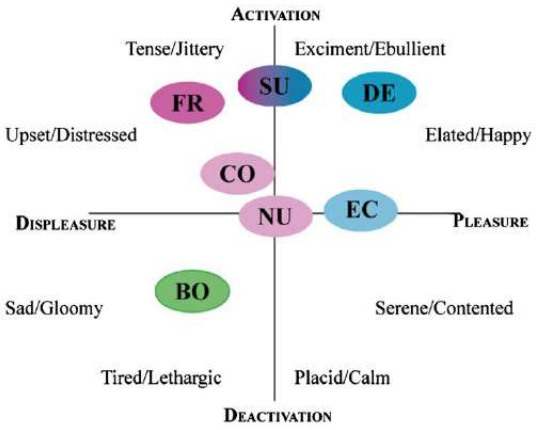


Figure 2: Learning-centred cognitive–affective states mapped onto Russell's Core Affect Framework (2003)

Table 2: Emotion with affect states mapping associations

Ekmanian Emotion	Time-dependent affect state
Sad	Boredom
Happy	Engagement
Surprise, Anger, Fear	Frustration

Through these associations the aforementioned algorithm was trained in order to predict affective states related to the theory of flow (boredom, frustration and engagement).

2.1.2.3 Initial results: classification accuracy

Using 'peaked'[1] images from the Cohn-Kanade database to train the SVM classifier in the method described in the previous sections (around 80% of all 'peaked' images were used as the training set) and the remaining 20% of 'peaked' images as the test set, the proposed approach yielded the results seen in Table 3 (representing the values in the confusion matrix's diagonal). A confusion matrix was used to evaluate the performance of the classifier over the test set, resulting to an overall accuracy (calculated as the sum of the diagonal of the matrix divided by the entire matrix's sum) of 91.67% for the proposed algorithm.

Table 3: First results using CK database for Ekman emotion labels

Emotion	Accuracy
Anger	100%
Disgust	86.37%
Fear	60%
Happiness	100%
Sadness	75%
Surprise	100%

For the second experiment, the SVM was trained after removing data correlated to disgust emotion, due to the fact that the disgust emotion does not map to any of the Theory of Flow states, as can be observed in Figure 2 and Table 2. Then the labels are mapped to flow theory as shown in Table 2.

In the specific experiment, in the Cohn-Kanade database, the label 'Sadness' is more accurate after removing the 'Disgust' label, to which the classifier falsely assigns some of 'sadness' data.

In addition, grouping Surprise Anger and Fear into the same affective cue group of Ekmanian emotions to map to flow affect states according to the mapping rules of Table 2, further boosts the prediction accuracy of the algorithm when it comes to recognising affect states, since Fear images

[1] Each emotion instance in Cohn-Kanade is represented by a series of images, starting from neutral, peaking to the most representative state of the emotion and then returning back to neutral. Peaked images are the ones, roughly in the centre of each series, in which the expression of the emotion has peaked to the most representative state for each test subject.

falsely classified as Anger are now merged into the same affect state category. After these, the precision of the proposed algorithm, when it moves from plain emotion classification to affect state classification, rounds up close to 100%.

2.1.2.4 Future plans

Although, this facial analysis approach gives quite good results as seen in the previous section, in the MaTHiSiS context there is more demand for accuracy of emotion prediction, using landmark detection with different head poses and a requirement for dealing with situations when facial occlusion appears.

To cope with these issues, several alternatives have been tested including the one depicted below, making use of a new alternative of convolution neural networks (CNN) (18), the spectral CNN (Figure 3).

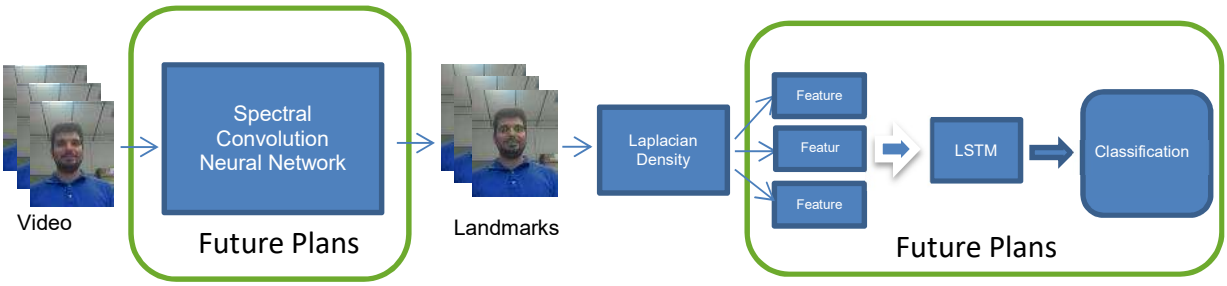


Figure 3: Facial Analysis Algorithm Graph Based Method - Future plans

CNN has been extremely successful in machine learning problems as well as in facial landmark detection (19). In the approached future solution, the Spectral CNN Tracker replaces the current SDM landmark detector and the SVM classifier is replaced by a Long Short-Term Memory (LSTM) (20) network.

In MaTHiSiS, a spectral convolution layer will be implemented instead of the typical convolution operator due to two main characteristics. The first one is the use of graphs, instead of regular grid structure of the image, to define the notion of "locality". In this way neighbors of points can be defined on an image by taking advantage of face structure represented as a graph in the image. The second aspect, is that spectral convolution can greatly reduce the number of parameters in a neural network giving the possibility to create more complex networks while retaining acceptable training time.

The input of the spectral CNN is not only an image (as is the case for classical CNNs) but also a graph which represents the mean facial structure, among all emotion expressions, in an image. Therefore, all layers (convolution operator, pooling layer which is a form of non-linear down-sampling, and linear layers which are the high-level reasoning in the neural network) are adjusted to take into account this new architecture.

From a classification point of view and in order to cope mainly with facial occlusions, MaTHiSiS will replace the SVM classifier with a Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN). LSTM neural networks take advantage of deep neural network characteristics and add some useful properties. LSTMs help preserve the recognized error, which can be back-propagated through time and through the different layers of the network. So, if for a frame, facial landmarks are not detected correctly, the classification will detect this error and provide more accurate results through LSTM, since the RNN prevents errors transiting through time.

The input of the LSTM is the sequence of the features described in Section 2.1.2.1, but, as previously mentioned, using the spectral convolution neural network instead of SDM for landmark detection.

2.1.3 MaTHiSiS facial expression extraction in the wild - Appearance Based Method

A second, appearance-based, approach of facial expression analysis is also considered within MaTHiSiS.

2.1.3.1 Preprocessing

In facial expression analysis, the pipeline starts with face detection, facial landmark detection and face tracking. Feature extraction and emotion prediction can then be outputted.

Face detection and tracking: For face detection, two universal algorithms trained appropriately are used. OpenCV (21) provides the Haar Cascade classifier, while Dlib uses a HoG-based detector (22). While the former is employed more frequently, the latter performs faster and more reliable face detection. Dlib's detector was trained on 3000 images from the Labeled Faces in the Wild database using a SVM max-margin classifier (23). For preprocessing the available data, dlib's classifier was primarily used, with OpenCV's only being employed as a fallback when the former would not yield a confident detection. For better real-time performance, Dlib provides a correlation tracker (24) implementation. It tracks objects in images reliably and performs iteration on the aforementioned in real-time. Once a face has successfully been detected, the tracker can take over, thus increasing the detection performance and rendering the algorithm suitable for real-time applications.

Facial Landmark Detection: Succeeding the step of face detection, facial landmark detection algorithms can be applied. The method of Kazemi and Sullivan (2014) (25) is a successful face shape regression technique, which begins with an initial S0 face shape and progressively predicts the final shape of the facial landmarks in an iterative way. Comparing to other techniques, this method provides robust and accurate landmark positions in challenging conditions, such as varying illumination and pose, and low quality images. In addition, it gives a reliable and robust tracking of facial landmarks in the wild in real-time. Figure 4 provides examples of faces with accurate land-mark detection.

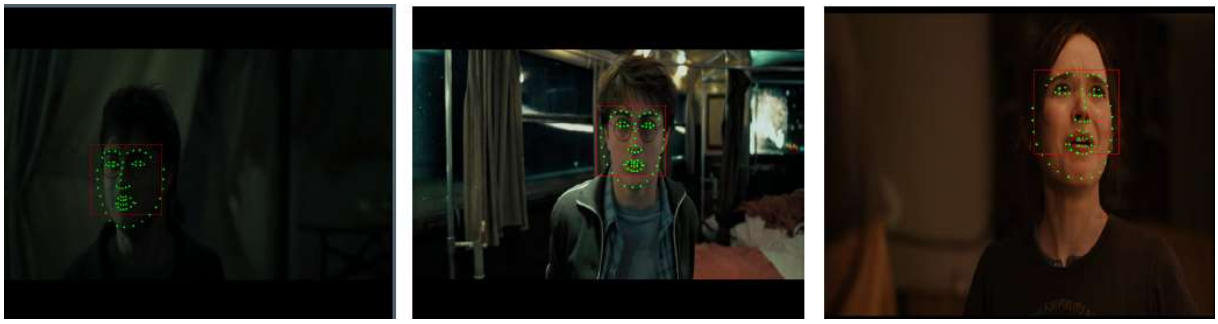


Figure 4: Face and facial landmark detection

Face alignment: Face alignment is an essential step in facial emotion recognition. It is the process of registering faces with respect to facial landmark (e.g. eyes, nose, mouth, and chin) to canonical frame. This process fixes the landmark positions in aligned images and it is carried out by similarity transformation. MaTHiSiS uses facial landmarks provided by (25) landmark detector and performs a affine transformation that aligns faces to the fixed canonical frame based on eye centres positions and mouth. In addition, facial images are cropped and re-sized to a fixed resolution: 224 × 224. In Figure 5, examples of aligned and cropped faces are depicted.

Figure 5: Face Alignment

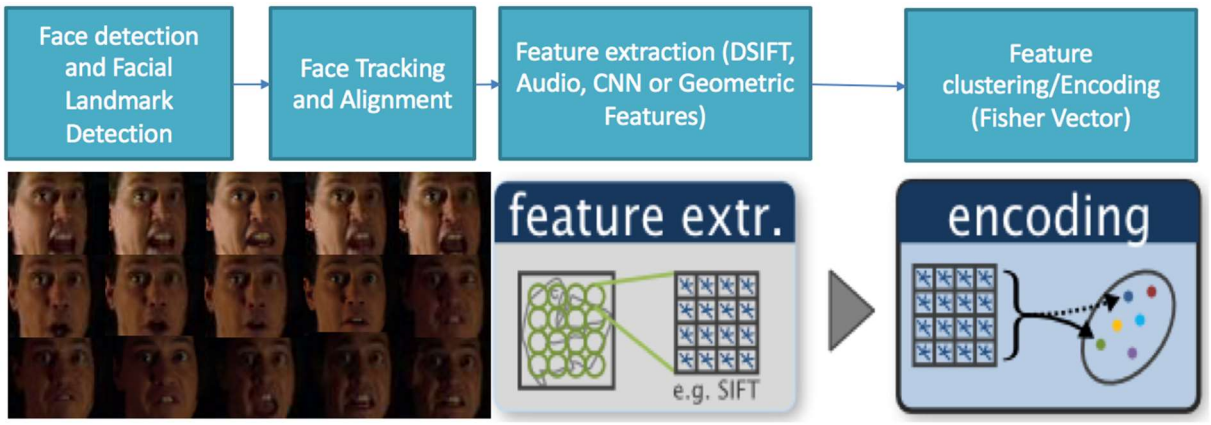


Figure 6: Face pre-processing and feature extraction and encoding.

2.1.3.2 Feature Extraction and Encoding

Emotion recognition relies on representative data along with accurate and discriminative descriptors. This type of information contributes to obtaining enhanced recognition and classification accuracy. Accordingly, low-level descriptors for the visual modality are extracted. Then, Fisher vectors are used for feature encoding and video modelling.

Dense SIFT: Dense Scale Invariant Feature Transform (SIFT) has been widely used for image representation in the last decade in many computer vision recognition tasks (26), (27). In dense SIFT, instead of sparsely detecting and selecting the facial key-points, the SIFT histogram is computed densely over a given image with a scale and a space. This has an advantage since it does not rely on a facial landmark detector. The facial images are divided into a grid of overlapping blocks with a step size equal to 1. Specifically, the block size is 24× 24. Later, a SIFT histogram is computed for each block. This step is repeated in 5 scales, with a scale factor equal to √ 2. D-SIFT is computed on the whole facial image.

Feature Encoding and Video Modelling: MaTHiSiS adopts the usage of Fisher vector for encoding and clustering different level features for face modality. The features are not only pooled from one still image, instead they are pooled from all the frames across a face track. As suggested in (26), video pooling is used, where a single fisher vector is computed over the whole face track by pooling together low-level features (e.g. D-SIFT, or CNN features) from all facial images in a track. This kind of representation has many advantages compared to still image based representation for various reasons: (i) it encodes the spatio-temporal information in a face track, (ii) it captures the motion of the face over time which leads to a better description of the different low-level features; and (iii) it dramatically reduces the dimensionality of data by producing a single discriminative descriptor for a

video. Figure 6 shows the whole pipeline of the appearance-based facial analysis process, namely the pre-processing sub-pipeline (top), examples of tracked facial images from the AFEW dataset (bottom left), feature extraction (bottom centre) and feature encoding (bottom right).

2.1.3.3 Dataset

Acted Faces Emotion In The Wild (AFEW): There are several facial expressions datasets gathered in controlled environments, which mainly contain still images or videos of frontal faces. Furthermore, the facial expressions are posed, limiting the capacity of the data to reflect real-world challenging conditions. Therefore, the AFEW dataset was chosen for several reasons: (i) AFEW is a challenging dataset with occlusions, varying illumination and head poses, which meet real-world conditions; (ii) it provides baseline results and an evaluation protocol which is useful to evaluate the efficiency of the MaTHiSiS scheme and (iii) it is currently studied by the research community, as it was the subject of several competitions over the last few years.

The Acted Faces Emotion in the Wild dataset is divided into three subsets: Train (723 samples), Validation (383 samples) and Test (539 samples). It has both audio and video modalities. In this dataset, the task is to classify a sample audio-video clip into one of the seven categories: Anger, Disgust, Fear, Happiness, Neutral, Sadness and Surprise.

The dataset has in the wild settings, containing wide pose, expression and illumination variation which reflect real world challenging conditions. Figure 7 illustrates examples of still images and a face track, where the various challenging illumination and pose conditions can be noticed.

Video Modelling: Video-pooling is used, where the low-level features are pooled from all the frames across a face track in each video of the AFEW train and validation sets. Then, a single fisher vector is computed over the whole face track by aggregating and encoding low-level features (e.g. D-SIFT) of all frames.

Evaluations Metrics. In the validation experiments for this approach, two evaluation techniques were taken into consideration: (i) Accuracy: which is the number of correctly classified video samples; (ii) Confusion Matrix between the ground truth and the predicted emotion labels.

Figure 7: Example of still images of affective states and a face track from AFEW dataset

2.2 Gaze estimation

Gaze estimation refers to an emerging computer vision research topic that is defined as the process of determining the eye's point of regard, usually with respect to a specific plane such as a computer screen but also in the more general case of the eye's orientation or a person's "look at" direction.

Recent gaze estimation methods aspire to estimate a person's gaze accurately, invariant of the head pose, the lighting conditions and the eye's appearance, using low cost commodity hardware and

simple set-ups. Gaze based methods can be classified into three general categories according to Hansen and Ji (28):

- Shape (Feature) based ones,
- Appearance-based and
- Hybrid ones, combining elements from the two previously mentioned ones.

Shape-based methods (28) are a-priori methods, utilizing a known model of eye shape and its surrounding structures. Usually these methods consist of two components: a geometric eye model and a similarity measure and are further categorized as either 3D model-based or 2D regression-based.

Appearance-based methods do not extract local features but use the input eye images to estimate gaze directions. Then a mapping function is learned from the eye images to gaze directions. Appearance-based methods need large amounts of training data compared to the shape -based ones. Various methods have been applied such as: Gaussian processes (29), manifold learning (30) , adaptive linear regression (31) and support vector regression (32). Last but not least, CNN are used in many recent works with very good performance, as can be seen in (33), (34).

Finally, hybrid methods combine feature, shape, and appearance approaches to exploit their respective benefits (28). Depth sensors (Kinect) are also used in various combinations to estimate a person's point of regard. Mora and Odobez (35) use a 3D Morphable Model (3DMM) to form a person specific 3D face template (3D mesh), then the head pose is extracted from the depth sensor and the head image is transformed to a frontal pose. Eye images are cropped by the frontal pose and used to estimate gaze. Gaze is estimated using adaptive linear regression (ALR) first in head coordinate system and then in world coordinate system. Although the system does not achieve state-of-the-art results gaze direction is estimated using low resolution eye images.

"Deep learning refers to a class of machine learning techniques, where many layers of information-processing stages in hierarchical architectures are exploited for pattern classification and for feature or representation learning" (36). Deep learning methods have been applied in speech recognition, visual object recognition, object detection, natural language processing and recently in identifying molecules that could lead to new drugs and many other domains. The progress in computational power, the lower cost of computer hardware, new methods of splitting computing tasks among many machines and the availability of great amounts of data have been some of the main enablers that contributed to the evolution of deep learning. Human vision, artificial intelligence, and natural language understanding are areas where deep learning is expected to make a large impact over the next few years (37).

In the context of gaze estimation, one of the most recent works by Krafka et all (34), has introduced GazeCapture, a dataset created by mobile devices (mobile phones and tablets) consisting of 1500 subjects with the use of crowdsourcing. The benefit of using crowdsourcing is that recording conditions vary in illumination and head pose. A convolutional neural network (CNN), namely iTracker, was trained with this dataset, using the images of both eyes, the image of the face and the position of the face in a grid as the input to the network. The estimated prediction error is 1.71cm on mobile phones and 2.53cm on tablets (the prediction error is estimated with the use of the average Euclidean distance from the target in centimetres) without calibration. With calibration, the estimated prediction error decreases to 1.34cm and 2.12cm respectively. Further, in the same paper, it is demonstrated that the features learned by iTracker can also be used with other datasets for generalisation purposes, achieving state-of-the-art results.

In another recent work utilizing deep networks to estimate a user's gaze direction, Zhang et all (33) published the MPIIGaze dataset containing images by 15 users. Data was recorded by users applying the data collection software on their laptops in-situ without any special guidelines given about the head pose, the time of execution or the illumination conditions. The data collection procedure lasted more than three months. In this way The MPIIGaze dataset collected 213,659 images with large variation in illumination and head pose. Head poses and eye images are the input to a CNN. The

output is the gaze direction in the camera coordinate system. Data collection was performed by the rgb cameras (web-cams) of laptops. A special calibration procedure was performed in each laptop before the data collection. The MPII estimation model outperforms state-of-the-art methods in person- and pose-independent training scenarios.

Following the success of data driven methods, MaTHiSiS has trained a gaze estimating regression network in a similar way to these state-of-the-art deep learning gaze estimation techniques. Initially, the MPIIGaze dataset is used since in the MaTHiSiS project gaze will be estimated in uncontrolled illumination conditions without expecting from the user any special head pose. Moreover, in several use cases of MaTHiSiS such as the autism case and people with profound and multiple learning disabilities case, no special head pose for gaze estimation can be expected by users. Additionally, in the MPII case we aspire to estimate gaze by a single RGB camera. For the above mentioned reasons the MPIIGaze dataset is the most similar to the MaTHiSiS use cases, and the accompanying method, where the input of the CNN is an eye-image and the accompanying head pose seems most appropriate. At this initial stage the concept is similar to the MPIIGaze method as an effort to thoroughly understand state of the art algorithms and subsequently place them within the MaTHiSiS context. The algorithmic pipeline of estimating the user's gaze is presented in Figure 8 and is outlined below:

i) Face detection is initially applied to locate the user's face region.
ii) Then, facial landmark detection follows to locate the user's landmarks from a calibrated monocular RGB camera.
iii) Following, these are used to establish 2D-3D correspondences and estimate the user's head pose.
iv) Next, the frontal facing view of the user's face is synthesized.
v) Finally, the eye image is detected in the frontalised image using a second pass of the facial landmark detection algorithm.
vi) The eye images are size-normalized and contrast enhanced and fed as input to the CNN along with the extracted head pose.

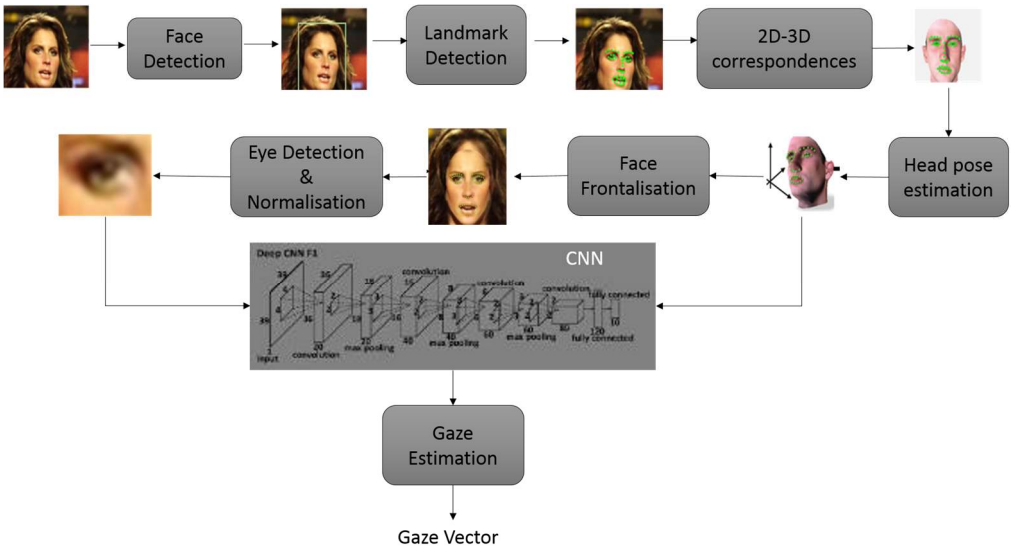


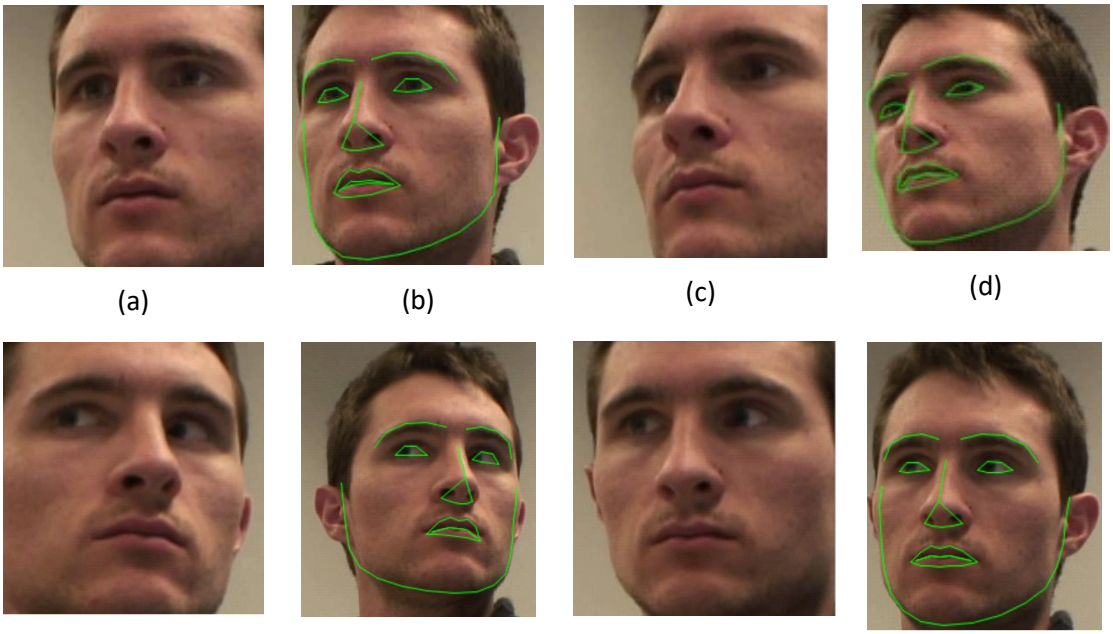
Figure 8: The overall gaze estimation pipeline.

2.2.1 Face and landmark detection

The user's face is detected using a standard cascade detection method, more specifically, Li et al.'s SURF cascade method (38). Assuming a single user, the largest face is selected and all images without

faces are naturally discarded. Afterwards, a state-of-the-art landmark detector is utilized (25) to locate the specific landmarks. The selected method was selected due to the fast processing speed offered by the highly parallelizable regression trees.

Figure 9: Examples of the utilized landmark detector results from the HELEN database (left most) and from a monologue contest video.



(a) (b) (c) (d)

(e) (f) (g) (h)

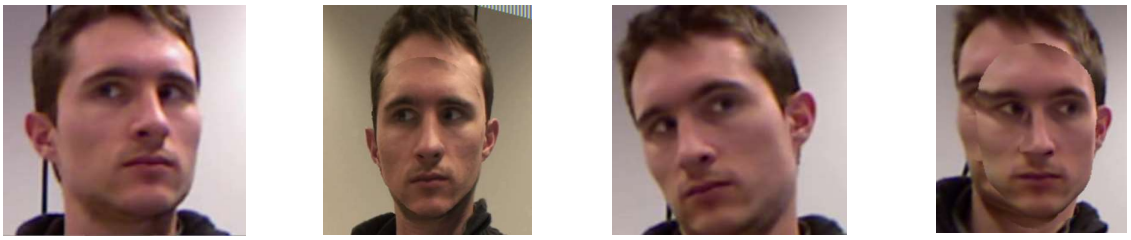
Figure 10: The landmark detector results on the EYEDIAP database (1_A_DS_M example).

2.2.2 Head pose estimation

Utilizing the 2D detected landmarks, correspondences with pre-annotated 3D positions can be established. These are annotated on a generic 3D mean facial shape head model and are used to estimate the user's head pose by fitting (39) the 3D model data to the 2D image correspondences via non-linear optimisation. The result is the head's transform (rotation & translation) with respect to the coordinate system defined by the camera.

2.2.3 Face Frontalisation

Using the estimated head pose and the generic facial shape model, the head's pose can now be removed to help in simplifying the problem of gaze estimation. Frontal face views are synthesized using the method of Hassner et al. (40) through bi-linear interpolation. This procedure results in a newly rendered image where the user's head pose has been removed.



(a) (b) (c) (d)

Figure 11: Examples of the face frontalisation method from the EYEDIAP database. (a) and (c) are the original images, while (b) and (d) are the frontalised ones respectively.

2.2.4 Gaze estimation implementation and initil results

After detecting the eyes (second pass of landmark detection) in the frontalised image, they are fed into a pre-trained deep CNN along with the user's head pose after normalizing them both (size and contrast for the eye images and 2D angle parameterization for the head pose). This regression network then produces a feature vector representing the user's gaze direction. The current MaTHiSiS trained model mean angular error is 14.5 degrees using the MPII dataset, where around 90% of the dataset was used for training and the remaining, randomly selected, 10% was used for testing.

2.3 Mobile device-based emotion recognition

Recently there have been many studies that support the potential usage of recognizing users' emotional states through various mobile embedded sensors such as accelerometer and gyroscope (41), (42), (43), and (44). Inspired by research on this subject by Coutrix (41), identification of emotion by exploiting the expression through 2D and 3D gesture using aforementioned sensors can be attempted in MaTHiSiS. As stated in (41), 2D and 3D descriptors contribute to emotion expression while interacting and using mobile phone.

In order to perform the experiments of emotion recognition through mobile based inertia-sensors, the streamed data should be gathered and annotated properly. It is important to design this framework accordingly to store sensory data in a meaningful and time interval based way.

In the MaTHiSiS context, data will be collected from mobile-based inertial sensors and users' touch behaviour. As a result, regardless of the application and the learning material running on the mobile device, the launcher will monitor sensory data from the gyroscope, accelerometer and touch panel.

Data gathering will be in sessions for multiple users in an adjusted environment such as a classroom or at home (it is not possible to analyse this kind of data if the learner is using the devices in a moving vehicle). Diversity of the users should be considered and the users must have different profiles such as age, disabilities, skills and gender.

The acquired data must be properly annotated. The annotation model used in the MaTHiSiS project indicates the emotion state of the learner based on the basic theory of flow (boredom, engagement, and anxiety). These three-dimensional and continues space can be accurately mapped to affective states from the theory of flow.

2.3.1 Data Pre-processing

Following the step of data collection with the proper annotation, data analysis is applied on the raw gesture logs from the input of accelerometers, gyroscopes and touch screen and also from the standard Z-Scores (standard deviation either above or below the mean) for each subject that will participate in the data collection step. The feature extraction from these raw data (2D touch input and 3D motion) descriptors will be used for the data analysis and emotion recognition.

2.3.1.1 3D motion gesture descriptors

For feature extraction from 3D motion raw data, as suggested in (41), extraction of (1) acceleration values and acceleration's derivative (jerk), and (2) spectrum will be considered. For acceleration and jerk, the projection of the 3D components is considered as such: **X:** left-right movement of the device, **Y:** bottom-top movement of the device, and **Z:** back-front movement of the device. In addition, minimum, maximum, amplitude, median, mean and variance of the signed and absolute values will be obtained. These descriptors of the acceleration and jerk are based on both raw data, and high-pass filtered values.

For spectrum, the gap (G) that maximizes the difference between most and least important frequencies in the spectrum of the acceleration signal will be computed, to show the gesture periodicity. In addition, the number (N) of the important frequencies in the spectrum, and the most important frequency (M) will be obtained.

As mentioned previously, all the three features are computed from all the 3D motion components (X, Y, Z) projections of raw acceleration values, and on the high-pass and low-pass filtered acceleration values.

2.3.1.2 Feature extraction from 2D gestures input

In order to obtain the descriptors of 2D input, as suggested in (41), (1) touches and (2) strokes level will be considered.

Firstly, touches' locations, touches' slopes (angle between two consecutive elementary drawn segments), touches' speed (e.g. instantaneousness speed between two consecutive elementary drawn segments) will be analysed. Secondly, the number of strokes, strokes' total length, strokes' length, strokes' locations, strokes' speeds and strokes' slopes will be examined. These features must be extracted for the projections on the X and Y axes and also for the vectors' length. In addition, similarly to 3D motion gestures' descriptors, minimum, maximum, amplitude, median, mean and variance of their signed and absolute values will be computed.

2.3.1.3 Basic rules

Due to the lack of available public datasets, no specific algorithms have been implemented for the time being. During the first pilots, a set of rules will be used to establish the affective state of the learners based on inertial sensors data and users' touch behaviour. The rules defined to that end are the following:

- Frustration: Movements registered by the gyroscope and/or accelerometer exceed a certain threshold. Abrupt movements of the mobile device could represent hand movement (possibly) owing to frustration, anxiety or anger.
- Boredom: No touches in the screen during a long period of time (e.g. 10 seconds). The lack of activity could be a symptom of distraction and/or boredom.
- Boredom: Lack of (or slight) data changes registered by the gyroscope/accelerometer. Similarly, the lack of movement can be a result of distraction, boredom or of abandoning the device, for example, on the table.
- Engagement: Short time between consecutive touches. Usually, the learning activities in this kind of devices will be conducted by touching the device's screen. For this reason, a constant use of the touch screen could represent that the learner is concentrated in the learning process and it could be considered that they are engaged.
- Engagement: Smooth movements (using a threshold in the gyroscope/accelerometer data). Likewise, smooth movements registered by the inertial sensors could represent a state of concentration during the learning experience.

Since the field of recognising user emotions or affect states from mobile inertia sensors is new, there is no precedent approach in the literature to base these rules on. Therefore the rules presented above rely on intuition and they will be verified and revised, if needed, using real data.

2.4 Skeleton Motion Analysis

During the last decades, Human Motion analysis (HMA) became an active research domain. Applications like Sport Motion Analysis, Human-Computer Interaction (HCI) and Smart or Automated Surveillance highlight the significance of HMA. A variety of surveys have been published, each one suggesting a different taxonomy of the HMA methods. A summary of some of them, in chronological order from the earlier to the most recent ones, can be found in the following section.

2.4.1 Motion analysis relative work

A review of motion-based methods is presented by Cedras & Shah (45). More specifically, they discuss about methods for human action recognition like walking, skipping and running. In addition, methods that involve the extraction of high level concepts like, hand gestures interpretation and lip reading are also discussed.

Moreover, an organization of model- and model-free based motion analysis algorithms is presented by Aggarwal et al. (46). The former methods apply an a-priori shape model while the latter do not. Aspects like the motion analysis of human body parts and human action recognition are discussed. Additionally, human tracking algorithms that use either a single or multi view camera system are presented.

Gavrila (47) presents a taxonomy of gesture motion and entire body analysis algorithms. Those algorithms are classified either as 2D or 3D. 2D approaches subdivided into those that use explicit shape models or not.

Moeslund & Granum (48) studied human motion capture (HMA) and they found four basic functionalities that characterize them: initialisation, tracking, pose estimation and recognition. Based on these functionalities they classify HMA methods.

More recently, Aggarwal & Ryoo (49), following an approach-based taxonomy, classified all activity recognition methodologies into two categories: single-layered approaches and hierarchical approaches. More specifically, they discuss methodologies for simple human actions and high level activities.

Weinland et al. (50) present a taxonomy based on how methods represent the spatial and temporal structure of actions, how they segment actions, and how they learn a view-invariant representation of actions.

Zhang et al. (51) discuss a feature-based taxonomy. More specifically, based on the features used for motion analysis, they classified popular methods into three groups: holistic feature-based methods, local description-based methods and biologically inspired methods. In their work, they applied for first time Slow Feature Analysis (SFA) for feature extraction in the scope of human action recognition. SFA belong to the biologically inspired methods. Below, the abovementioned categorization concentrating mostly on methods that applied SFA is discussed in greater detail.

Holistic Features Approach: Laptev et al. (52) & Zhang et al. (53) mention some of the holistic features-based methods. The properties of moving objects (blobs), silhouette, trajectories of reference joints and motion templates are utilized in such methods. However, such methods are sensitive because they require accurate actor segmentation and body tracking, processes that are affected by background motion.

Local Features Approach: To overcome the difficulties that arise from the noise background, local descriptors were proposed. Laptev et al. (54) proposed histogram of gradients (HoG) and histogram

of optic flows (HoF) in the 2D domain to characterise the motion and appearance on the neighbourhood of interest points in space and time. Klaser et al. (55) extended 2D HoG in (54) to the 3D case. Laptev & Lindeberd (56) proposed spatiotemporal interest points for action representation. Those points have rotational and translational invariance. A drawback in (55) is the sparseness of interest points which may lead to information loss. Dollar et al. (57) proposed a method where a rich corpus of interest points was computed by applying a series of spatiotemporal filters. Oikonomopoulos et al. (58) by computing the entropy of spatiotemporal regions extended the salient point detector in (59). Schuldt et al. (60) constructed video representations in terms of local space-time features and integrated these representations with SVM classification schemes for recognition.

Biologically Inspired Approach: Recently, a variety of methods for action recognition (52) (53) were proposed based on the slowness principle. The basic assumption of this principle is that primary sensory signals, which in general code for local properties, vary quickly while the perceived environment changes slowly. The slowness principle provides a natural hypothesis for the functional organization of visual cortex, and possibly also other sensory areas, casting this approach as a biologically inspired one. This principle was incarnated by Wiskot and Sejnowski (61) by proposing an unsupervised feature learning method called Slow Feature Analysis (SFA). One of the main characteristics of this method is that it produces invariant and slowly varying features from input signals.

More specifically, Zhang et al. (53) introduced for the first time SFA to the problem of action recognition. They presented an SFA framework to learn in unsupervised manner invariant features for action recognition. In particular, they proposed four types of SFA learning strategies. Namely, the unsupervised SFA (U-SFA), the supervised SFA (S-SFA), the discriminant SFA (D-SFA) and the spatially-discriminative SFA (SD-SFA). These proposed extensions, used to extract slow feature functions, from a corpus of cuboids that were selected randomly from precomputed motion boundaries. The final computed features were fused in one Accumulated Squared Derivative (ASD) feature which encodes the statistical distribution of slow features.

Sun et al. (62) combined SFA with deep learning to learn hierarchical representations from the video data itself. More specifically, they applied a two layer SFA framework with 3D convolution and max pooling. The advantage of this method, on the contrary to those where hand-made features were used (see Holistic and Local feature methods above), is that the deeply-learned SFA method is generic and fully automated.

On the contrary to the proposed methods (53) & (62) where video datasets were elaborated, Shan et al. (63) applied SFA for activity recognition on skeleton stream data collected from a depth sensor. More specifically, they proposed features of three different orders, called first, second and multi order features. Those features were later fed to a Support Vector Machine (SVM) for classification.

2.4.2 An introduction to Slow Feature Analysis (SFA)

Standard SFA is based on the Principle of Slowness (61). According to this principle, the context of a raw input signal recorded by a sensor is encapsulated into the slow variations of the signal. For example, given a video where a monkey is passing through the video frame, there are two forms of information that the human eye cortical is recording. The first types of signals are those that are recorded by the thousands of retinal neurons. The variation of these signals corresponds to the variation of the video frame pixel luminance when the monkey is passing through the video frame. These raw recorded signals do not reveal the real context of the video shot, which is the pass of an object in front of the scene. For an illustrative example see Figure 12. On the top of the figure there is the video described previously, while on the bottom left the retinal neurons are depicted, as they recorded the raw signal. At least visually, it can be observed that the primary sensory signals do not describe the context of the scene. The second type of signals, are those possibly extracted by the human brain and that support us with the visual context of the scene. These signals are a transformation of the raw sensory input signals that is described before, which reveal the meaningful behaviour of the recorded video. A visual representation of such signals is given at the bottom right

part of Figure 12. The new output signals may contain information about both the identity and the location of the moving object, which is actually part of the context information of the scene. Further on, a description of the standard SFA in mathematical terms, is provided, thus introducing the MaTHiSiS proposition of use of the 2D SFA.

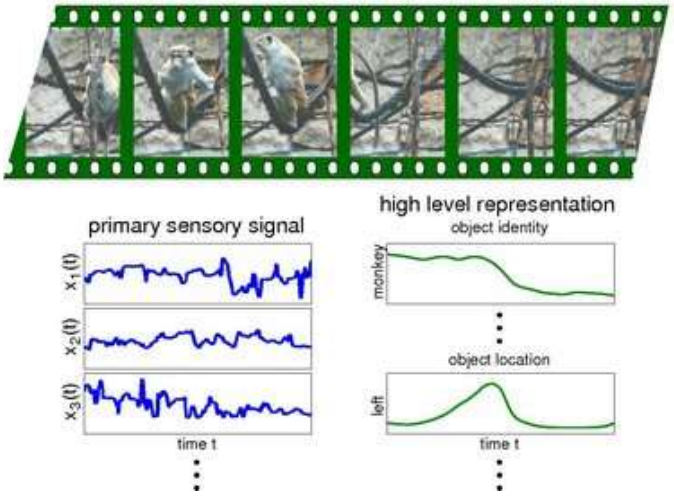


Figure 12: Illustration of the slowness principle

2.4.3 Standard SFA

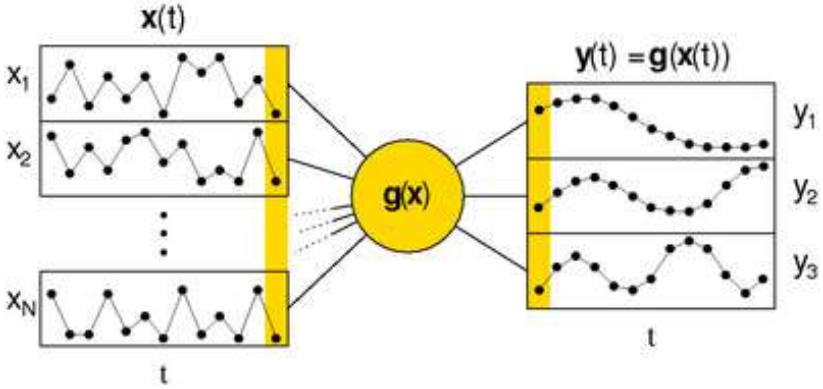


Figure 13: Schematic description of SFA flow

To begin with, a pictorial description of standard SFA is shown in Figure 13. This figure describes the three main parts of the standard SFA algorithm. On the left there is the raw sensorial input data, in the middle there is the learned mapping function and on the right there is the mapped slow signals. In more detail, the time instances of the input signals are grouped into a column of a time varying vector $x^{(t)}$. Similarly, the time instances of the output signals are grouped into the time varying column vector $y^{(t)}$. For a specific time point t the function $g(x^{(t)})$ maps the input signal $x^{(t)}$ to $y^{(t)}$. The output signals contain the slow context of the raw data (see discussion in Section 2.4.2).

Given an I dimensional input signal $x^{(t)} = [x_I{}^{(t)}, \dots x_I{}^{(t)}]^T$, with $t \in [T_0, T_1]$ indicate time, Slow Feature Analysis (SFA), computes an input-output vector-valued function $g(x^{(t)}) = [g_1(x^{(t)}), \dots, g_J(x^{(t)})]^T$ such as to obtain a final output signal $y^{(t)} = [y_1^{(t)}, \dots, y_J^{(t)}] \in R^J$ where $y_i^{(t)} = g_i(x^{(t)})$. The g_i function in the linear case is defined as $g_j(x^{(t)}) = w_j^T x^{(t)}$ and for all g_j,

$y^{(t)} = g(x^{(t)}) = W^T x^{(t)}$ where $W = [w_1, \dots, w_J]$ and $W_{I\times J} \in (R)$ (61). This definition leads to a closed form solution. Thus, SFA finds a transformation matrix W to achieve the following $x^{(t)} \mapsto y^{(t)}$. To obtain the output signals SFA solves the following optimization problem,

$$\begin{aligned} \underset{g_j,\forall j}{\text{minimize}} \quad & \mathbb{E}[\dot{g}_j^2(\mathbf{x}^{(t)})] \\ \text{subject to} \quad & \mathbb{E}[g_j(\mathbf{x}^{(t)})] = 0,\ \mathbb{E}[g_j^2(\mathbf{x}^{(t)})] = 1 \\ & \mathbb{E}[g_j(\mathbf{x}^{(t)}) g_i(\mathbf{x}^{(t)})] = 0 \\ & \forall j \neq i, i = 1, \dots, J \end{aligned}$$

where $\dot{g}_j$ denotes the first order time derivative of the output function g_j and E indicates time averaging. The objective function in abovementioned optimization problem tries to minimize the variance of the input-output function g_j and this is done by minimizing the derivative of the function over time.

2.4.4 2D SFA

In standard 2D SFA the input signals are one dimensional column vectors. This idea has been extended in MaTHiSiS so that the proposed 2D SFA takes as input two dimensional vectors, i.e., a matrix, which varies through time. A recorded skeleton consists of a number of nodes (one dimensional column vectors) which are grouped into a matrix (a two dimensional vector). The proposed 2D SFA leverages this data format to extract a mapping of the input data useful later for classifying an action.

Given a two dimensional matrix $X^{(t)} = \left[x_1^{(t)}, \dots, x_D^{(t)}\right] \in R^{I\times D}$ that represents a skeleton in a specific time point t, 2D SFA computes, $G(X^{(t)}) = \left[g_1(X^{(t)})^T, \dots, g_J(X^{(t)})^T\right]^T \in R^D$ with $y_i^{(t)} = g_i(X^{(t)}) = w_i X^{(t)} \in R^{1\times D}$ and finally outputs a 2D signal for all g_i. Note that, from the vector-valued function $g_i(\cdot)$a linear combination of the rows of $X^{(t)}$ is obtained, which is the new feature representation of each column vector, whereas from the function $G(\cdot)$the new matrix representation $Y^{(t)}$ is obtained which is a concatenation of the J linear combinations extracted from the mapping functions $g_i(\cdot), i = 1, \dots, J$. In the specific 2D-SFA, dimension is reduced in only one matrix mode. Thus, in 2D-SFA it is desired to obtain a transformation matrix W that implements the transformation $X^{(t)} \xrightarrow{W} Y^{(t)}$ and this can be achieved by solving the following optimization problem:

$$\begin{aligned} \underset{\mathbf{g}_j,\forall j}{\text{minimize}} \quad & \sum_{j=1}^{J} \mathbb{E}[\dot{\mathbf{g}}_j(\mathbf{X}^{(t)}) \dot{\mathbf{g}}_j(\mathbf{X}^{(t)})^T] \\ \text{subject to} \quad & \mathbb{E}[\mathbf{g}_j(\mathbf{X}^{(t)})] = \mathbf{0}_{1\times D} \\ & \mathbb{E}[\mathbf{g}_j(\mathbf{X}^{(t)}) \mathbf{g}_j(\mathbf{X}^{(t)})^T] = 1 \\ & \mathbb{E}[\mathbf{g}_j(\mathbf{X}^{(t)}) \mathbf{g}_i(\mathbf{X}^{(t)})^T] = 0 \\ & \forall j \neq i, i = 1, \dots, J \end{aligned}$$

Below, a sample of the conducted experiments on the MSR action 3D dataset (64) is attached. The dataset records skeletons, from a depth sensor, that perform a set of actions.

2.4.5 Action encoding and Classification

For a given skeleton sequence that portrays an action, a unified description should be extracted, to be used later on for classification purposes. Thus, in order to encode a given action, a dictionary

should be learned. Moreover, because of the temporal information an action contains, a temporal pyramid must be computed. Inspired by Shan et al. (65) and Wang et al. (66) the following scheme was followed:

Key postures: Each skeleton frame is a posture, thus an action is a set of postures. Based on this, a dictionary of key postures was extracted which was later used for action encoding. To learn such a dictionary of key postures, Gaussian Mixture Models (GMMs) have been applied, initialised with K-means centres. The mean values of the learned K Gaussian models were used as the key postures.

Temporal pyramid: Each action was encoded by applying a three level temporal pyramid. In each level, a given action was split into non-overlapping and equal-sized action parts where a K-dimensional histogram was computed for each part by using the aforementioned key postures. In first level the whole action was used to compute such a histogram. In second and third level of the temporal pyramid, the action was split into two and four non-overlapping and equal sized parts. Similar to the first level histogram computation, a histogram was computed for each part in the second and third level. In total, for a given action seven histograms were computed, one in the first level and two and four histograms in the second and third level of the temporal pyramid respectively. Finally, the seven computed histograms were concatenated into a 7K dimensional histogram and normalized by applying l_2 normalisation.

Classifier: Action recognition was performed, by feeding the computed histograms to a X^2-kernel Support Vector Machine (SVM) using one-vs-all scheme.

2.4.6 Skeleton motion analysis experiments and initial results

Early experiments were conducted on the MSR-Action3D dataset, due to lack of comprehensive datasets for the educational domain and the purposes of MaTHiSiS. Following data gathering (cf. Section 4), and subsequently additional data captured through the MaTHiSiS driver pilots, the implemented solution will be re-trained in order to accommodate the actions relevant to the MaTHiSiS learning experience, subject to the particular needs of each MaTHiSiS' use case.

MSR-Action3D dataset: A Kinect-like (depth) sensor was used to obtain the recorded skeletons. It consists of 20 different recorded actions performed by 10 different subjects/actors. In addition, each recorded action was repeated two or three times by each subject. Namely, the actions are*, high arm wave, horizontal arm wave*, *hammer*, *hand catch*, *forward punch*, *high throw*, *draw x*, *draw tick*, *draw circle*, *hand clap*, *two hand wave*, *sideboxing*, *bend*, *forward kick, side kick*, *jogging*, *tennis swing*, *tennis serve*, *golf swing*, *pick up & throw*. For each skeleton the 3D joint locations through time were available along with the connections of the nodes that define the recorded skeleton. Each recording was done in 15fps. It is worth noticing that, because of the similarity of the actions, this dataset is considered a hard case to classify actions in.

Table 4: MSR Action3D subsets

Action Set 1(AS1)	Action Set 2(AS2)	Action Set 3(AS3)
Horizontal arm wave	High arm wave	High throw
Hammer	Hand catch	Forward kick
Forward punch	Draw x	Side kick
High throw	Draw tick	Jogging
Hand clap	Draw circle	Tennis swing
Bend Two	hand wave	Tennis serve
Tennis serve	Forward kick	Golf swing

Pickup & throw	Side boxing	Pickup & throw

Experiments have been conducted using four different skeleton representations introduced by Shan et al. in (65). Namely, those features were the 1st Order, 2nd Order, 3rd Order and Multi-Order streams.

A usual experimental setup is to split each action into three action subsets AS1, AS2 and AS3. The experimental results on MSR Action3D using the aforementioned streams are included in Table 5. The results of 2D-SFA were compared with the state-of-the art action recognition algorithms. The most relevant algorithm to 2D-SFA is that proposed by Shan et. al. (65), in which the SFA algorithm was also applied. The mean accuracy of these two algorithms is very close. However, 2D-SFA lacks in accuracy compared to the rest of the algorithms. That was the base line during the experiments conducted. The next step of MaTHiSiS work is to improve the algorithm framework so as to compensate with the other methods. See Section 2.4.7 for more details.

Even though 2D-SFA has lower accuracy results than the state-of-the-art method, it has lower computational cost in the training as well as the test phase. These characteristics make 2D-SFA appropriate for online tasks, particularly necessary in the scope of MaTHiSiS. The results given in Table 5 are preliminary and the final results on the MaTHiSiS recorded data will be evaluated in the future.

2.4.7 Future work

Future plans include increasing the accuracy of the 2D-SFA algorithm by improving feature extraction and also by applying more robust techniques for dictionary learning. A possible extension of the proposed linear 2D-SFA to the kernel form is under investigation. Kernel methods provide a promising framework for better feature extraction, thus potentially achieving better results in the 2D kernel SFA. As previously mentioned, the construction of a dictionary is necessary for action encoding. Here, GMMs were applied for such a purpose. In future work, more robust dictionary learning techniques, like Sparse Coding, will be applied.

Table 5: Experimental results on the MSR Action3D Dataset, comparing % accuracy with state-of-the-art methods

Method	AS1	AS2	AS3	Ave.
Li et. al. 2010	72.9	71.9	79.2	74.7
Chen et. al. 2013	96.2	83.2	92.0	90.4
Gowayyed et. al. 2013	92.3	90.1	91.4	91.2
Shan et. al. 2014	92.4	82.1	**97.1**	90.5
Vemulapalli et. al. 2014	**95.2**	83.8	98.2	92.4
Du et. al. 2015	93.33	**94.6**	95.2	**94.4**
MaTHiSiS: 1st order	84.7	83.9	95.3	87.9
MaTHiSiS: 2nd order	90.2	82.1	97.1	89.8
MaTHiSiS: 3rd order	85.8	72.3	95.2	84.4

MaTHiSiS: Multi-Order	90.2	83.9	97.1	90.4

2.5 Speech recognition and speech-based affect recognition

It is common sense that one of the basic means of human communication is the vocalised speech. Apart from meaning, speech also carries emotions. Although the latter are more easily recognised through visual channels, e.g. facial features, gestures, etc., in many practical applications, e.g. in human-computer interaction through voice-user interfaces (VUIs), speech may be the only available modality for emotion recognition. The latter comprises probably the most challenging speech-related task, e.g., when compared to automatic speech recognition (ASR), speaker identification etc. In general, one may argue that speech carries two distinct types of information (67): explicit or linguistic information, which concerns articulated patterns by the speaker; and implicit or paralinguistic information, which concerns the variation in pronunciation of the linguistic patterns. The former may be qualitatively described, while the latter may be quantitatively measured, using certain spectral features and also features such as the pitch, the intensity etc. Using either or both types of information, one may attempt to classify an audio segment that consists of speech, based on the emotion(s) it carries.

However, emotion recognition from speech appears to be a significantly difficult task even for a human, no matter if he/she is an expert in this field (e.g. a psychologist). Many approaches are assisted by ASR aiming to fuse linguistic and paralinguistic information. The main disadvantage of these is that they are not able to provide language-independent models. Of course, another disadvantage is that there exists a plethora of different sentences, speakers, speaking styles and rates (68). Thus, most approaches that aim to be language-independent tend to rely on paralinguistic speech information. Nevertheless, even in this case, such information may be significantly diverse, depending on cultural particularities. Additionally, a speaker's potential chronic emotional state may suppress the expressiveness of several emotions. Still, relying solely on paralinguistic information is probably the most appealing approach, when dealing with speakers' emotion recognition.

Within the context of MaTHiSiS, it is planned to collect data from users to train both a paralinguistic and an ASR-assisted approach. MaTHiSiS also aims to exploit data from publicly available data sets, for training purposes. As described in Section 4, a data gathering process will take place in order to create a comprehensive dataset for all modalities. Then based on this dataset all the algorithms of the MaTHiSiS Sensorial Component modalities can be trained in order to detect cues over the learners' affective behavior.

2.5.1 Data Pre-processing

Initially, each audio signal is transformed to a sequence of feature vectors which are then used for the training. Features are extracted on a short-term basis and from 20 ms windows and afterwards the final feature vectors are formed by concatenating the mean and variance values of the features over a mid-term window of 1 s. The short-term process can be conducted either using overlapping (frame step is shorter than the frame length) or non-overlapping (frame step is equal to the frame length) framing. This process is illustrated in Figure 14.

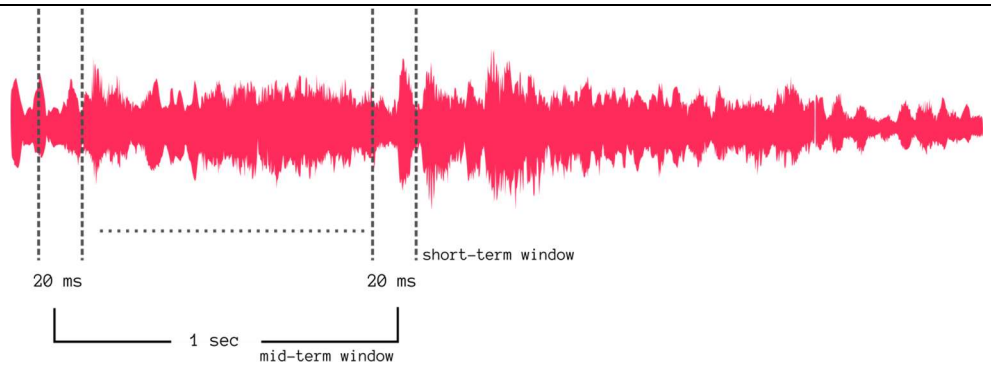


Figure 14: Mid-term window extraction process of low-level audio features.

2.5.2 Low-level Audio Feature Extraction

Table 6 presents a summary of the implemented low-level audio features that are extracted from audio signals, providing a brief description for each. A more detailed description is out of the scope of this deliverable but may be found in the following references (69), (70), (71).

Table 6: Implemented Audio Features

Index	Name	Description
1	Zero Crossing Rate	Rate of sign-changes of the frame
2	Energy	Sum of squares of the signal values, normalized by frame length
3	Entropy of Energy	Entropy of sub-frames' normalized energies. A measure of abrupt changes
4	Spectral Centroid	Spectrum's centre of gravity
5	Spectral Spread	Spectrum's second central moment of the spectrum
6	Spectral Entropy	Entropy of the normalized spectral energies for a set of sub-frames
7	Spectral Flux	Squared difference between the normalized magnitudes of the spectra of the two successive frames
8	Spectral Rolloff	The frequency below which 90% of the magnitude distribution of the spectrum is concentrated
9-21	MFCCs	Mel Frequency Cepstral Coefficients: a cepstral representation with mel-scaled frequency bands
22-33	Chroma Vector	A 12-element representation of the spectral energy in 12 equal-tempered pitch classes of western-type music
34	Chroma Deviation	Standard deviation of the 12 chroma coefficients.

2.5.3 Emotion Model Classification

For classification of feature vectors to emotions, the well-known and widely used Support Vector Machines (SVMs) (72) were used. SVMs are well-known supervised learning models, and during

recent years they have been extensively used in both classification and regression problems. Their goal is to find the optimal hyperplane separating data classes in a feature space. More specifically, an SVM model is built in a way that the margin between the mappings of the examples of the categories is maximized. Then, a hyperplane is constructed, which is used to separate (i.e., classify) unknown examples, based on the side they fall on. Although they are linear models, using appropriate kernels (i.e., a technique called "kernel trick") they are able to handle non-linearly separable data in feature spaces of higher dimensionality than the one of the original problem.

Using the method described in subsection 2.5.1, the audio signal is transformed to a sequence of feature vectors which are used for the SVM models' training. Features are extracted from 20 ms windows and afterwards the final feature vectors are formed by concatenating the mean and variance values of the features over a mid-term window of 1 sec. As it can be seen in Table 6, there are 34 features in total which result in a 68-dimensional feature vector for each mid-term window. Upon experiments with several kernels, an SVM with RBF kernel has been adopted within the presented approach.

2.5.4 Datasets Used and Initial Results

Since the data gathering process had not started at the moment of writing this deliverable, for training and evaluation of the aforementioned emotion recognition approach using solely non-linguistic audio features 3 widely known emotional speech datasets were selected, all of which are freely available from their authors. More specifically, these datasets are:

- EMOVO (73) - an emotional speech corpus, containing speech in Italian language from 6 actors who performed 14 sentences. The emotions represented here are disgust, fear, anger, joy, surprise and sadness.
- SAVEE (74) - a larger dataset, since besides speech, it contains video of the participating actors while expressing the same 6 emotions as in the EMOVO case. The data consists of 15 TIMIT sentences per emotion played by 4 English male speakers.
- EMO-DB (75) - a German acted database, consisting of 493 utterances performed by 10 (5 male and 5 female) actors expressing the emotions of anger, boredom, disgust, fear, happiness, sadness and neutral.

Since in MaTHiSiS it is planned that emotions/states recognisaed are going to be based on the Theory of Flow (76), i.e., boredom, engagement and anxiety, 5 of the common emotion classes, namely Happiness, Sadness, Anger, Fear and Neutral were selected from the aforementioned datasets. These were the emotions deemed to be closest to the ones that we would need to extract within MaTHiSiS.

A major difficulty resulting from the choice of datasets is the differences between languages, since besides the linguistic differences, there is also big variability in the way each emotion is expressed. For each classification method 6 different experiments were carried out where a single dataset was used for testing and the remaining 2 for training or 2 datasets were used for training and the remaining one for testing. Experimental results are depicted in Table 7 and are indicative of the capabilities of the adopted approach, emphasising the fact that training and testing data derive from *different* datasets.

Table 7: Indicative Experimental Results of Speech-based Affect Recognition

Emotion	Min Precision	Max Precision
Anger	0.231	0.795
Fear	0.222	0.469
Happiness	0.205	0.355

Neutral	0.348	0.574
Sadness	0.541	0.789

2.5.1 Future Work

Among immediate plans for MaTHiSiS is to train the speech-based affect recognition module using the data that would be gathered upon the completion of the corresponding process, which is described in Section 4.

To conclude, reaching the end of the first year of the project, first version of the speech recognition and speech-based affect recognition module has been completed, which is part of the sensorial component. Currently, this version is based only on the non-linguistic features of speech, i.e., it does not rely on speech recognition. The latter shall be implemented within the second year of the project. The flow of information will ultimately be the one depicted in Figure 15, i.e. the keywords spotted will be fused with the audio features upon an early fusion process, so as to improve the accuracy of classification.

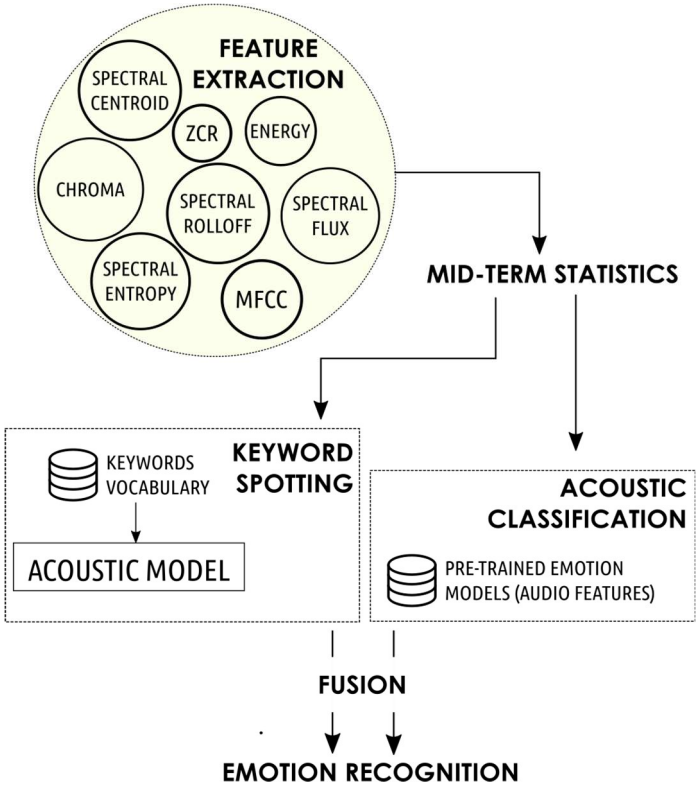


Figure 15: The flow of information within the speech recognition and speech-based affect recognition module.

3. Modality interconnections

This section describes the connections among modalities in the MaTHiSiS context. More specifically, the five Sensorial Component modalities (facial expression analysis, gaze estimation, skeleton motion analysis, speech analysis, mobile-based inertia sensor data analysis) will have to coordinate and provide output to the multimodal fusion mechanism (T4.3) in a uniform representation format and based on specific synchronisation tactics.

3.1 Output features per modality

This section provides a complete and aggregated list, along with explanatory details, of the affect-related features that are extracted through the SC per modality.

Table 8 Affect related features per modality

Modality	Affect related features
Facial expressions Analysis – Graph-based method	1x98 vector/frame. Vector contains values from first and second eigenvectors corresponding to greatest eigenvalues (see 2.1.2.1 for more details)
Facial expressions Analysis – Appearance-based method	1X2112 vector/video. The number of frames in video sequence can vary from 1, up to 500 frames. Fisher vectors dimensionality is 2Kd which depends on the number of the GMM components (K), and the dimensionality of the used set of features. Then, the Fisher vector ϕ is computed by stacking the differences (the assignment of the local features to the first and second differences of GMM centers): $\phi = [\Phi(1)^{(1)}, \Phi(1)^{(2)}, ..., \Phi(K)^{(1)}, \Phi(K)^{(2)}]$. The value used for K = 16, and d is the length of SIFT histogram reduced by PCA from 128 to 64, augmented by the spatial information = 64+2=66. As a result, this length is: 2*66*16 = 2112
Gaze Estimation	A Gaze vector describing the look direction. Target-based engaged non engaged label.
Mobile device-based emotion recognition	249 descriptors of 3D motion and 2D surface gestures. 3D motion: • Acceleration values and acceleration's derivative (Jerk) (x, y and z projections) • Spectrum analysis 2D surface gestures: • Touch parameters Stroke levels
Skeleton Motion Analysis	A 1x250 feature vector (histogram) that represents the video

Modality	Affect related features
	frame. For more details on the histogram computation see Section 2.4.5.
Speech recognition and speech-based affect recognition	1x34 vector/audio segment. Vector contains features described in 2.5.2 (Table 6)

3.2 Data Homogenisation

Data Synchronisation: Each sensorial component has a different time interval for data actuation. For example, visual modalities based on face detection can be done each 25ms while skeleton tracking and detection can be achieved each 22ms. This gap and time difference can be even wider in mobile based sensorial components such as accelerometers and gyroscope. Therefore, the features provided by various sensorial components in T4.1 have to be synchronized such that they belong to corresponding time interval. This is a vital issue since the task will produce affective states and affect related features to the multimodal learning component in T4.3. As a result, T4.1 must provide synchronized and homogenised features in order to fuse them and produce reliable and accurate affective states.

Feature Representation and Homogenisation: Data perception can be achieved through different and complementary sensorial components such as audio, video, and skeleton joints. The joint analysis of the sensory inputs leads to an improved recognition of the environment, since it enhances the understanding of an event through different channels. In this project, there are various sensorial components and corresponding modalities. Therefore, various feature extraction and representation techniques are applied for different modalities where their data comes from diverse input channels. However, each modality has its own distribution and statistical properties, and different sensory data have highly non-linear relationships. There have been many approaches that try to benefit from data stemming from various modalities, and subsequently try to optimise such multimodal frameworks, in order to obtain shared representations of data across different modalities.

In the case of MaTHiSiS, Fisher vector encoding is going to be used in order to map the extracted features into a common space, for achieving a higher layer feature description, which shares similar statistical and discriminative properties. Thus, different modalities are projected into one domain by means of fisher vectors. The newly obtained fisher vector based representation is independent of the input modality, as opposed to the low-level features which are modality-dependent.

Fisher Vector Representation: The pipeline for Fisher vector encoding typically starts with extracting a set of features (e.g. D-SIFT for face, and audio features), and then aggregates the large set of feature vectors across time or in each time interval into a high dimensional Fisher vector which is better suited for linear classification. This is achieved by fitting a parametric generative model such as Gaussian Mixture Models (GMM) to the features. GMM can be referred to as a probabilistic visual vocabulary. The next step consists of encoding the gradient of the local descriptors log-likelihood with respect to GMM parameters. GMM parameters are estimated on a large set of local descriptors using the Expectation Maximization(EM) algorithm to optimize the log-likelihood. In Fisher vector computation the covariance of the GMM is assumed to be diagonal and only the derivatives with respect to Gaussian mean and covariance are considered. This leads to a vectorial representation that captures the average first and the second order difference between dense features and each of the GMM centres:

$$\Phi(k)^{(1)} = \frac{1}{N\sqrt{w_k}} \sum_{p=1}^{N} \alpha_p(k)\left(\frac{X_p - \mu_k}{\sigma_k}\right)$$

$$\Phi(k)^{(2)} = \frac{1}{N\sqrt{2w_k}} \sum_{p=1}^{N} \alpha_p(k)\left(\frac{(X_p - \mu_k)^2}{\sigma_k} - 1\right)$$

Where w_k ,μ_k ,σ_k are the GMMs weights, means and diagonal covariance. $\alpha_p(k)$ is the soft assignment of the p-th feature x_p to the k-th Gaussian component. Fisher vectors dimensionality is 2Kd which depends on the number of the GMM components (K), and the dimensionality of the used set of features. Then, the Fisher vector ϕ is computed by stacking the differences (the assignment of the local features to the first and second differences of GMM centers): ϕ = [Φ(1) $^{(1)}$,Φ(1) $^{(2)}$,....,Φ(K) $^{(1)}$,Φ(K) $^{(2)}$]. A Fisher Vector representation has many advantages: (i) it is a generic representation which combines the benefits of generative and discriminative approaches, (ii) it can be computed using a small number of parameters (GMM parameters), (iii) more importantly, it is efficient and it shows a significant benefit when used in combination with linear classifiers such as linear-SVM.

4. Data gathering for MaTHiSiS dataset creation

The data acquisition tool aims to gather the raw signals of user (learner) behaviour, which reflect their affective state during the actuation of the learning process. The purpose of gathering this data is to create a comprehensive dataset, based on which the algorithms of the MaTHiSiS Sensorial Component modalities, as well as the rest of the WP4 algorithms (Interaction with PAs (T4.2) and Multimodal Fusion for uniform affect recognition (T4.3)), can be trained in order to detect cues over the learners' affective behaviour.

Data gathering is going to take place in the premises of the MaTHiSiS educational institutions, where a subset of the MaTHiSiS learners, that will take part in the pilots, will be asked to undertake a series of activities, which will be adapted to the requirements of each MaTHiSiS use case.

4.1 Technical Requirements

The sensorial cues of the learners' affective state in MaTHiSiS are captured through facial expressions analysis, gaze estimation, skeleton motion analysis, speech-based emotion recognition and inertia sensors input analysis for mobile devices. These modalities have several technical requirements that need to be met during data capture in order to efficiently capture all possible, relevant, expressions of users' affective behaviour. Beyond the sensorial modalities, this activity aims to gather data for the rest of the WP4 components, i.e. tracking learners' interactions with the Learning Materials on the PA (T4.2), and holistic characterisations of the overall affect state of a user during a Learning Experience.

In general, during data gathering a wide range and variety of all data, which can be extracted from the sensors that will take part in the MaTHiSiS end product, should be recorded along with timestamps that will allow precise synchronization of all the different inputs in post-processing. The sensory data that will be recorded in particular are HD video (from Kinect 2, including depth and information that accrue from it, such as head pose, body skeleton, etc.), non-HD video (web camera – also emulating NAO cameras), audio (from Kinect 2, including sound direction) and mobile inertia sensory data (touch, proper acceleration, orientation/rotation, etc.). Furthermore, interaction tracking in terms of accuracy and time of response to a question will be tracked (for T4.2).

4.2 Workflow

Certain modalities require a (minimal) set of actions to be performed by the session manager before recoding the users, for **calibration** purposes. Therefore, while setting up the data acquisition tool, either the tutor/trainer, the MaTHiSiS partner that manages the task on-site, or a technician will be called to interact with the tool on a calibration panel, given appropriate instructions. A scheme is devised to define when and how many times calibration data should be re-tuned for the system, since movement of either the monitor or the cameras used for recording will compromise the fidelity of the data acquired, for the camera-dependent components. For this reason, (re-)calibration will be conducted, each time the system is first set up, before any other activity (for the very first time, for the day or after a long interval between user sessions) or whenever there is a visible displacement of the monitor or one of the cameras, but also every 2-3 user sessions, to ensure that there is no significant deviance in the spatial correlation among the visual sensors and between them and the visual target due to gradual, non-striking, displacement.

The first time a learner interacts with the system, the tutor/trainer (or the user him/herself where possible) will be called to fill in a **form** with some personal information and minimal metadata that will facilitate indexing of the acquired data. Personal information will include a user id, the MaTHiSiS user case that the user falls under (ASC, PMLDC, MSC, CGDLC, ITC), age and gender, along with the academic competence of the learner in subjects of the interaction tracking activity (cf. Section 4.4.2) and some context information, such as the learner's preferences or wishes that can affect her/his

emotional state (e.g. the learner is very motivated with the use of computers and this issue can provoke a better engagement during the tests performed using this device, or the learner has limited to none vocal expression capabilities, or the learner has limited mobility). Metadata will include the particular institution where the data gathering takes place and the ID of the MaTHiSiS partner(s) on site. If the learner doesn't complete her/his predefined tasks in one session (there are three milestone points in which the user may interrupt the session, without losing pre-conducted information) or if the same learner will take up a second session, this step will merely involve logging in the system and selecting the corresponding learner profile based on the unique user ID.

After these introductory steps, the learner will conduct the core data gathering task, where s/he will be **recorded** as s/he interacts with the data acquisition tool **in scripted activities/games**. S/He will be positioned (sitting or standing per specific activity) in front of a computer screen, with two cameras (Kinect v2 and web camera) placed opposite her/him in predefined positions, except in the case of inertial data gathering, where the learner will be interacting with a mobile device, where no particular positioning is required – cameras will keep recording however, in order to maintain a reference for the second phase's annotator. The system will present several activities to the learner (in random order, so as to avoid bias in gathered data, e.g. all learners are bored in the same, last, activity), which s/he will be called to complete to the best of her/his abilities. During the learner's execution of the activities, her/his interactions will be recorded. The process and requirements of it will be detailed in the following sections. Finally, the other core data gathering task will comprise of **annotating** gathered data. An annotator will be called to annotate recordings of the user sessions with a set of predefined labels, representing 3 affect states, corresponding to the most reliably trackable subset, through sensory means, of the 8 affect states of Theory of Flow, for which more details will be presented in Section 4.3.3.

The recordings of this event will comply to strict security procedures and follow the data protection strategy deployed by Task 2.4's LEPOSA framework (cf. Deliverable 2.6).

4.3 Pedagogical Requirements and Scenarios

The purpose of this section is to outline the pedagogical specifications concerning the data collection task. It will provide an overview of the scenarios/activities that the users will perform in order to fulfil the task or goal, and of the desired variety and range of data, with respect to the pedagogical requirements.

4.3.1 Pedagogical Requirements

Special consideration was given to pedagogical requirements, especially in what has to do with younger learners and more so in the cases of ASC and PMLDC students. To this end, Task 2.1 (and T4.3) pedagogical experts have guided the design of the DAT activities in order to accommodate the particular needs of these students.

The guidelines included valuable indications towards interface design (e.g. high text-to-background contrast for the ASC learners to be able to take to the system), relatability to the activities, pointers to simplification of activities designed initially for the MEC, ITC & CGDLC cases towards activities of the same underlying mechanics but at the same time performable by learners with limited communication capabilities (whether those might be vocal, motion-related or otherwise), as well as with respect to covering the vast diversity of communication means and needs of the diverse MaTHiSiS learners.

It should be stressed that the tool that will be designed in order to actuate this data acquisition task is ***not*** directly corresponding to the actual MaTHiSiS platform, but rather aims to cover the technical requirements concerning needed training data for the WP4 algorithms. To this end, the criterion of

the design is the uptake and responsiveness to the tool's activities, and not the exact mapping of the activities performed in the DAT to the MaTHiSiS use cases and specific scenarios (although many activities are applicable to the actual scenarios and will be used in the pilots and the pedagogical guidelines elicited in this process will apply to the actual MaTHiSiS framework). Meaning, that the ITC and CGDLC cases will mostly conduct the same or similar activities as the MEC students, since the goal is the affective reaction to the system, which can be achieved through the more simplistic (for adults) MEC activities.

4.3.2 Learning Activities Scenarios design and requirements

The Sensorial Component modalities require vast diversity in expressions and settings in order to capture adequate data to train the SC algorithms to run in the wild. To this end, in this data collection task, variant positioning of the recorded user will be pursued if at all possible (while the scenario will be simplified for those learners who have limited expression capabilities) in order to capture the full range of physical behaviour of users, thus exhibiting a real-life expression of their affective states – as opposed to the rigidity of the fixed positioning and/or behaviour of users in publicly available datasets.

More specifically, in the case of **facial expressions analysis**, one or more activities should trigger the expression of the basic (Eckmanian) emotions by the user, i.e. guide the learner to express anger, disgust, fear, happiness, sadness and surprise through facial expressions. To this end, pictures, a video or sounds maybe displayed on the data acquisition tool's player that either evokes these emotions to the user, or that exemplifies them so that the user can copy them.

Another important technical requirement for facial analysis, with respect to the MaTHiSiS use cases that require natural interaction of the users with the (not always static) MaTHiSiS PAs, is to capture expressions on faces that are oriented in a variety of angles (head poses) with respect to the camera, i.e. not only the fully frontal orientation. To this end, the scenario design makes provision for a range of head poses of 90^o angle spanning left to right (45^o left and 45^o right of the fully frontal pose) and of 90^o up-down span (45^o upwards and 45^o downwards). In addition, a variety of different positions, with respect to the distance of the user from the camera, will be pursued, in a range of 50-200 cm.

In the case of **gaze estimation**, a wide range of gaze vectors need to be captured. To this end, activities are designed in order to compel the user to follow with his/her gaze a small-sized object on a computer screen, though a series of known, alternating positions of the object within the user's field of view.

As in the case of the facial analysis, detection in a variety of head poses, spanning 90^o degrees on both the horizontal and vertical axes, needs to be carried out. In terms of distance to the camera, positions of a range of 50-80 cm will be pursued.

In the case of **skeleton motion analysis**, again activities are be designed that mimic certain predefined intrinsic motions that express either certain predefined spontaneous emotions, i.e. happiness, anger, fear, sadness or certain pre-defined body motions that are indicative of the user's affect state, e.g. facing a stimulus, turning away from the stimulus (body trunk), head positions relative to body (facing/turning away), moving towards and away from target (e.g. screen), sway/tilting left-right (especially relevant to the ASC and PMLDC).

The motions of the body should be captured once by all users on the fully frontal orientation of each user's body towards the camera, while a second iteration is needed, with different orientations and positions of the user relative to the camera (90^o degrees horizontal and vertical orientation span, 50-200 cm. distance from camera).

The variations in angles and distances is chosen empirically, since the actions/expressions of any user placed beyond the aforementioned distances or turning more than the defined angles will not be able to be captured by the sensors of each respective case.

In the case of **speech-based recognition**, activities should be designed in a way that the learners articulate vocalised speech, if vocal expression is possible (not always the case for learners in the PMLDC). A prerequisite would be that the learner is asked to read or respond to a predefined text, or predefined text segments, wherever that is possible, which are appropriately selected so as to i) be neutral in terms of contextual sentiment and ii) contain a set of predefined words which would need to be spotted by the automatic keyword spotting method.

In the designed activity, the learner is prompted to vocalize the predefined words/sentences - if possible, or respond to a stimulus with simple words, or even utter sounds (if vocal expression capacity is limited), while free speech is encouraged throughout the entire capturing process.

Regarding **mobile sensor information capturing**, the designed games/activities will take advantage of all the mobile sensors and stimulate a variety of movements, in order to capture data from the accelerometer and gyroscope and interactions with the touch panel in order to capture touch behaviour. Obviously, devices that offer all types of sensors (accelerometer, gyroscope, touch panel) are required to be used during data gathering. In addition, the user must hold the mobile in his/her hands and avoid putting the mobile in any flat surface and perform the task in a controlled area

Finally, **interaction tracking** will be manifested in a controlled exercise, where the user will be guided to interact with the activity by clicking on specific items on the screen, in response to quiz questions, while her/his response will be recorded and correlated with the correct response, along with the time it took her/him to respond.

All these requirements will not be captured by the same activity at once, in order to avoid inducing the learners to a mentally/psychologically strenuous exercise. Rather, several activities/games are developed, designed in order to fulfil all the aforementioned requirements. Moreover, all requirements need not to be met by all users. For instance, users can be separated into groups, each group being assigned with a different position and/or orientation towards the camera, while interacting with the data acquisition tool.

The last essential requirement of the data gathering process that was defined in collaboration between technical and pedagogical partners, is the method to conduct the holistic affect state **annotation** of gathered data. This involves a second level of annotation (in contrast with the previous activities, where affect-related cues will be extracted automatically from recorded content), by an observer, with the help of the supervising tutor. The observer will manually annotate intervals of intrinsic user activity with the most appropriate affect state labels (boredom, engagement, frustration), on each recorded learner session, while adding a scalar value to that label that will represent how much this affect state was prominent for that learner at that given time interval.

To this end, the WP4 and pedagogical partners have considered simple methods that might compel the learner to **emulate or evoke the affective states** during parts of an activity. E.g. providing rewards during some timeframe might increase engagement, asking to perform repetitive actions for another timeframe could bring about boredom etc.

4.3.3 Annotation tool background and process design

The Theory of Flow (76) will be the pillar upon which affective state recognition within WP4 will build. It revolves around the notion of maintaining an actor fully focused and involved throughout the execution of an activity. Based on the flow theory, the emotional condition of people can be classified in 8 different states (Figure 16). Furthermore, there is a simpler approach based on three states, i.e. anxiety – or frustration, flow – or engagement and boredom (Figure 17). Transition between these three states, and guidance towards the desired one, namely flow, can be controlled through the modification of the challenge associated to the current activity performed by the user, while taking into account their skill level at the activity at hand.

In fact, engagement is the conjunction of the Flow theory's arousal and flow states. However, distinguishing between the two cannot be performed through sensory information alone, and relies

on user performance over tasks at hand. To this end, sensory-based modalities will focus on tracking the more straightforward superset of engagement, which will then be further broken down to the respective arousal and flow states based on the learners' performance over conducted activities.

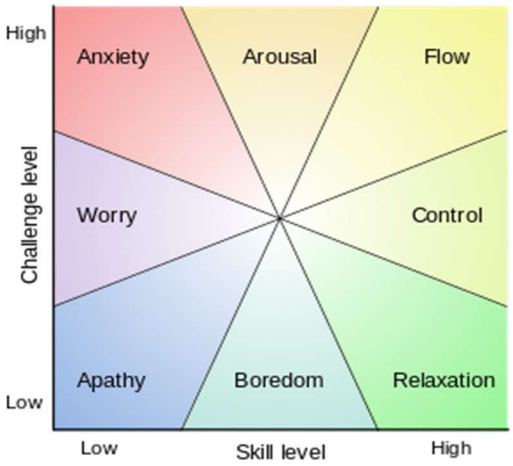


Figure 16: Theory of Flow affective states

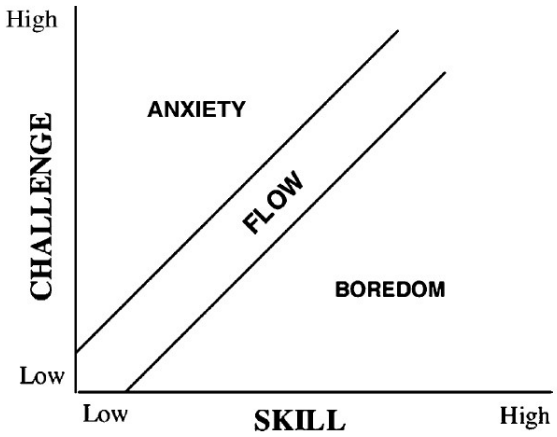


Figure 17: Flow model

The supporting annotation interface within the DAT will provide the annotator with the recording of both the user (through one of the tracking cameras) and the interactions s/he has conducted at the same time (through screen capture). The annotator will be called to label outstanding timeframes of the user session recording with one or more of the 3 labels of the aforementioned prominent affect states. This annotation will be correlated with the first level features that will be produced in T4.1 (and T4.2), in order to effectively provide a more holistic and comprehensive mapping of the different features to overall affect states (i.e. affect states as a result of the combinations of information among all modalities – T4.3 multimodal fusion)

4.4 Data Acquisition Tool (DAT)

As mentioned previously, the Data Acquisition Tool (DAT) is the core application responsible for collection of the data needed for training the algorithms of the SC modalities described in this document. It is also responsible for the Interaction Tracking component (T4.2) and subsequently the multimodal fusion for unified affect recognition (T4.3) which will be analysed technically in future MaTHiSiS deliverables. The Data Acquisition Tool supports all the languages from all countries of the

MATHISIS consortium, and from the introduction screen, the user (tutor, trainer or individual learner) can choose the desired language, before heading into the main application interface.

4.4.1 Main Window

The main window (Figure 18) contains the three main buttons of the DAT application: “New Session”, “Menu” and “Exit”.

Menu involves 4 submenus which are: “Languages” (DAT functionality to add or edit new languages), “Resources” (an interface to add or edit the all the application text in each language), “Calibration” (detailed in Section 4.4.4) and “Annotation Tool” (the interface which is used to annotate the data acquired with affect labels. More details in Section 4.4.3)

New Session involves all the functionalities corresponding to the activities that the learner is going to conduct in order to capture her/his affective cues and, subsequently, upon the recording of which affect state annotation is going to take place. More details in 4.4.2

New Session involves all the functionality corresponding to a learner. More details in 4.4.2

Exit closes the application.

Figure 18: DAT Main Window

4.4.2 User Session

The user session is the core of the DAT, where the set of activities developed for the data acquisition purposes will be presented to each learner, who will be called to interact with the material presented. While doing so, s/he will be recorded through a depth and an RGB camera, a microphone (embedded in the depth – Kinect v2.0 camera), as well as in terms of her/his interactions with the mobile device and with a specific interaction tracking (i.e. response accuracy and time to reaction tracking) quiz.

4.4.2.1 Learner details form

The first step of every session is to choose the learner who will be the tracked subject of this session and provide some metadata for archiving purposes, such as the name of Institution/Organization in which the data gathering process takes place for this session, and Supervisor, i.e. the person from the MaTHiSiS consortium (or from within the Institution) who supervises the whole process.

Following the creation of a new (or selection and modification of the details for a known) learner, the learner details form is brought up (Figure 19). For each learner, the DAT stores some information, i.e. the (anonymised) learner ID, her/his gender and age, the MaTHiSiS use case that the learner falls into (ASC, CGDLC, ITC, MSC, PMLD) and her/his academic competence over some predefined subjects, relevant to the interaction tracking quiz, presented further on. Also, some parameters of the DAT activities, personalized per user preference, are set. These parameters are: the Background colour of the games, the colour which highlights the words in text enunciation (where applicable) the colour of presented text and the way it is displayed.

Especially designed for PMLDC and ASC learners, but applicable to all cases where it is deemed useful, there exists an option for the tutor to select a personally relatable image (or even sound) for

each specific learner, from their own local collection of content, which is correlated to simple emotions that the user is called to express or recognize in the process of some activities. This is particularly important in the PMLDC and ASC, since relatability is the key-most factor that can ensure actual interaction with the DAT, or any other system.

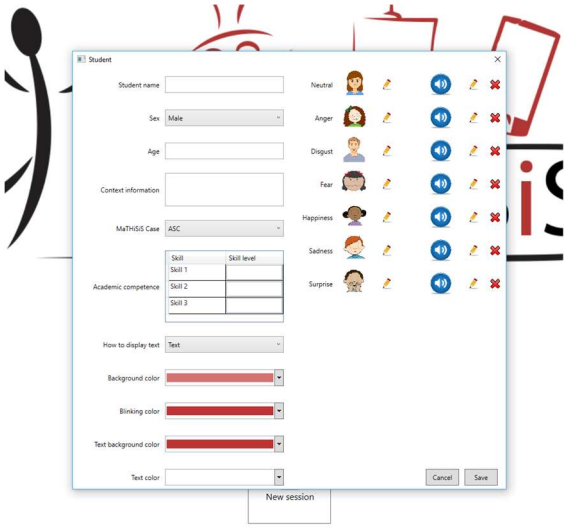


Figure 19: Add - Edit Learner window

4.4.2.1 Activities

Two major activity storylines are available as options for the learners, according to his/her communication capabilities and learning styles: the “Cookie monster”, especially for PMLDC and ASC students, as well as for very young MEC students, which provides a somewhat uniform storyline of significantly simple tasks/activities. The factor of relatability again drives this storyline, where following a specific, previously introduced, character (in this case the Cookie Monster) is imperative for the participation of some Special educational needs (SEN) students. The “Smiley faces” game is again rather simple but meaningful enough in terms of required interactions/behaviours, so that it can be followed through by all the other cases, from MEC students to ITC and CGDLC adult trainees, even though the theme does not match the ITC and CGDLC cases.

To meet the requirement as described in 4.1 images, video, sound, and gaze position are collected. Images for facial analysis are gathered by asking the learner to express the emotion presented by a cartoon (Figure 20).

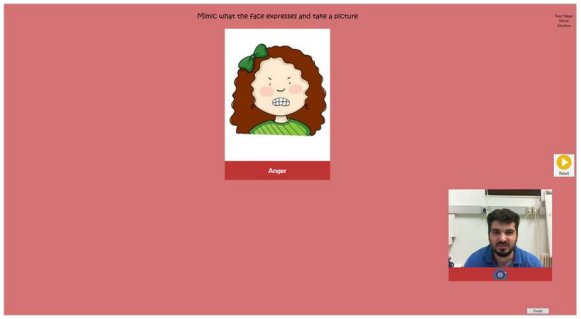


Figure 20: Smiley faces game: mimic emotion

The video for skeleton analysis is captured by asking learner to perform the motion that learner sees on screen, where a video that exemplifies motions is displayed Figure 21)

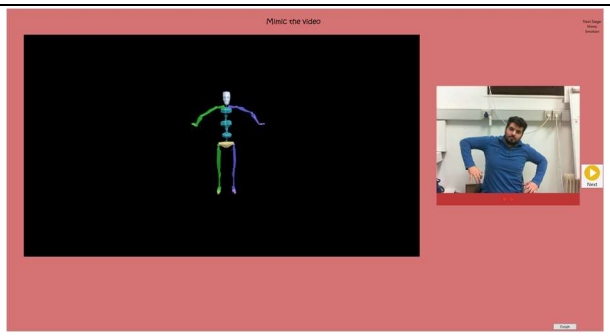


Figure 21: Smiley faces game: mimic motion

The MATHISIS game asks the learner to click a specific cartoon (for MEC, ITC and CGDLC learners) or help the "cookie monster" to eat the cookie (for ASC and PMLDC learners) (Figure 22). In that way the gaze target of the learner can be collected.

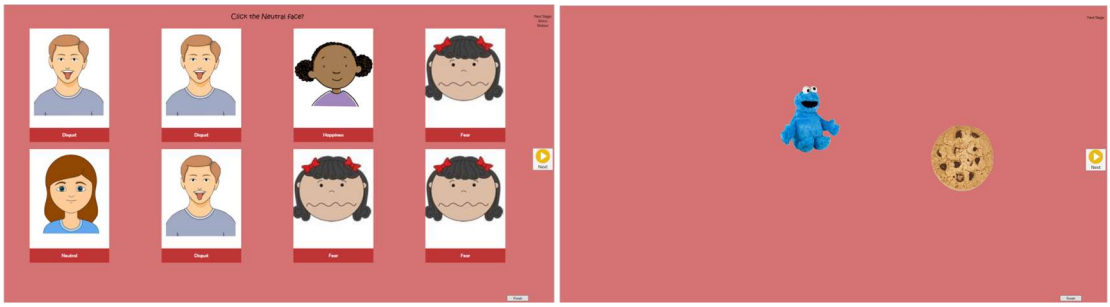


Figure 22: Smiley faces game screenshot; (right) cookie monster (left) point-n-click game.

At some point of the trials, the user will be asked to pick up the available mobile device and play the DAT mobile game (which should be installed on the device by the tutor or technical assistant before-hand). Accelerometer, gyroscope and interactions with the touch panel data are gathered during the simple mobile game (Figure 23). Two different games will be used, namely a version of the popular '2048'[2] game for older MEC learners and for ITC and CGDLC learners and an even simpler cookie monster-themed game for young MEC, ASC and PMLDC learners. While the game is running and the user interacts with the device, a background service will be capturing and transmitting the inertia sensor-extracted data to the main desktop application, to be registered in the central database.

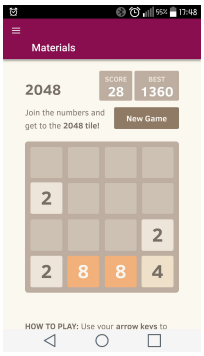


Figure 23: Mobile screen – the 2048 game.

An interaction tracking quiz has been designed for both acquisition of data related to sensorial components (video recording) and the tracking of learner responses while interacting with Platform

[2] https://en.wikipedia.org/wiki/2048_(video_game)

Agents and Learning Materials. This application consists of a "question and answer" serious game which can track different interactions such as performance during the learning experience (questions score or time needed to provide an answer) or mouse/keyboard events. This information is very useful for learner's affect recognition and learning analytics.

The application interface has been developed following the recommendations of pedagogical experts, especially for ASD and PLMD cases. In those cases, the interface shows a better contrast between useful information and background, bigger size of the font and specific questions can be included, where answers are based on the use of symbols.

In connection with the database of questions, different courses for each use case have been defined. Each available course contains 4 different levels of difficulty and each of these levels includes, at least, 7 questions. For mainstream, a mathematics course has been included. In the case of industrial and career guidance cases, specific courses has been developed which include questions related to technical information in the companies where the DAT sessions will take place or questions about portfolio creation respectively.

This application has already been tested and used to gather data. The data acquired has been used to train algorithms for personalization and adaptation of the learning content in the context of task 6.2.

Figure 24: Interaction tracking quiz: Assessment questionnaire

4.4.3 Annotation

Following the completion of a user session (or a set of sessions), the annotator, whether that would be the tutor/trainer, an observer/assistant, or a MaTHiSiS consortium partner who attended the user session and had noted down observations based on feedback from the tutor/trainer[3] opens the "Annotation tool" window (Figure 26), selects a learner and a specific session to annotate.

The annotator plays back a video which simultaneously shows the camera/audio recording of the learner and, superimposed, the screen capture of the activities that the learner performed at that moment of the recording. S/he selects time intervals (marks the start and end on the video timeline) in which one (or a combination/transition) of the three affective states mentioned in Section 4.3.3 took place. S/he assigns one or more label(s) on that timeframe, while s/he also appoints a scalar

[3] Feedback from the tutor/trainer is desired due to the familiarity that s/he has with the learner and how that person expresses her/himself.

degree (1-5) of the perceived intensity of the particular affect state for that time interval. There are also functionalities to remove and edit annotations.

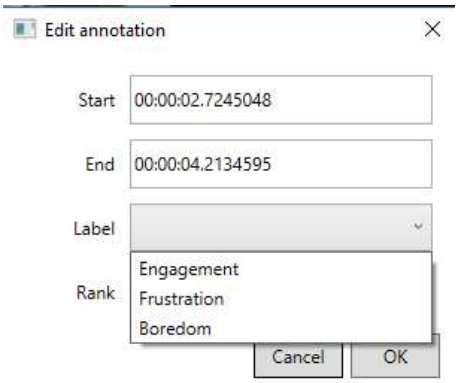


Figure 25: Annotation tool: annotation window

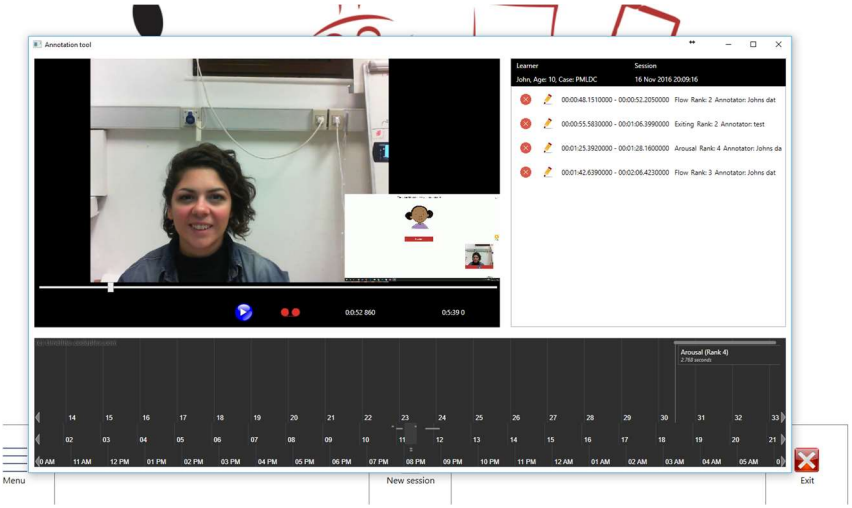


Figure 26: Annotation tool: main window

4.4.4 Calibration

As detailed in Section 4.2, minimal calibration actions are required by a technical assistant/supervisor at regular intervals. These activities aim at maintaining the two cameras used in the data acquisition (Kinect and standard web camera), as well as the visual target (screen) in the same coordinate space.

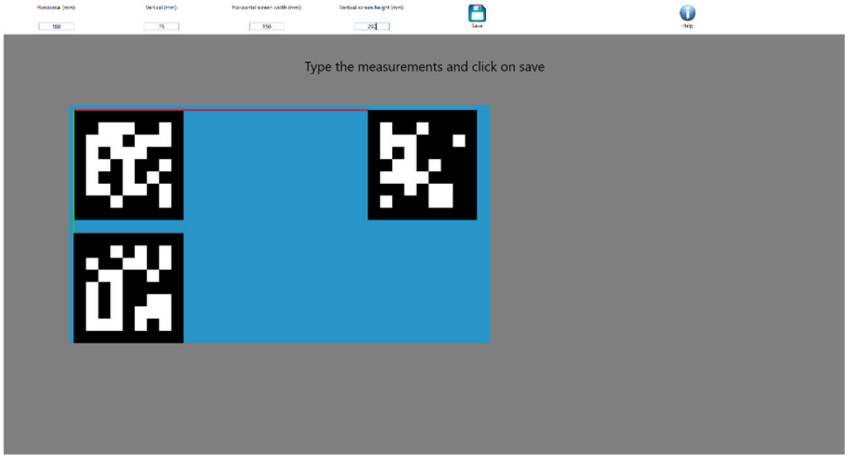


Figure 27: Mirror calibration screen measurements

Mirror-based calibration aims to calibrate the cameras with the screen reference plane. In the first phase of the mirror calibration procedure, the technical assistant performs screen measurements, mainly measuring length of a red line (line between points A & B in Figure 28) and a green line (line between points A & C in Figure 28), both lighting up on the screen, along with the horizontal screen width and vertical screen height.

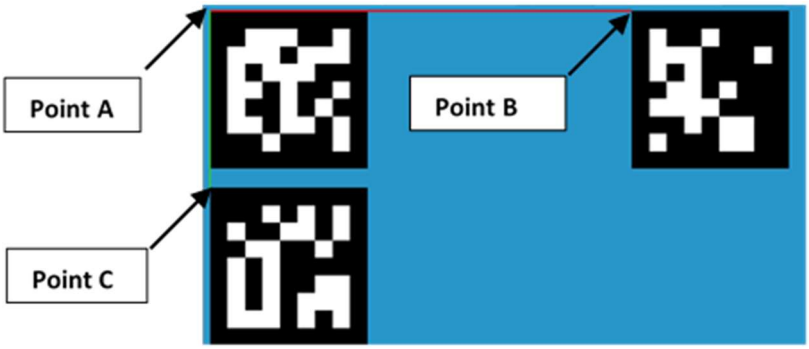


Figure 28: The three points detected in the image

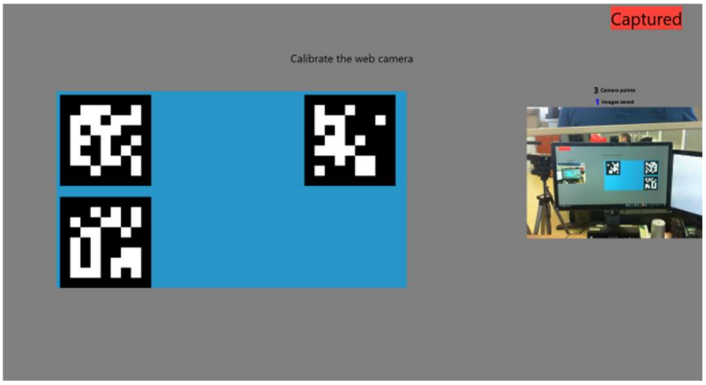


Figure 29: The Web camera calibration screen

After this, the calibration operations proceed to the "web camera to screen calibration" screen (Figure 29). The assistant places the mirror facing the screen, so that the mirror reflects the entire image of the "arucho board" (the blue plane in Figure 29). Then images of the screen through the mirror (detecting automatically the three black and white markers on the screen) are captured by the application. On the right side of the screen the stream of the Web camera is presented.

In order to calibrate the Kinect camera, the technical assistant performs the same procedure with the Kinect camera streaming this time (The system moves to this screen automatically).

4.4.4.1 Data collection for mirror calibration of the screen

The motivation of a mirror-based extrinsic camera calibration (77) is to calibrate a camera with respect to a reference object which is not observable from the camera. The key point to solve the problem is the use of mirrored images, by observing three points of a reference object on the screen via mirrors (mirrored images) under different unknown positions and orientations. Certain positions of the mirror are very important in order to extract useful information. Some important positions are: a) the mirror located in parallel with the screen and in a variety of distances (Figure 30) b) the mirror rotated around its y axis to the left by the parallel previous position (w.r.t. the mirror's y axis) c) the mirror rotated around its y axis to the right by the parallel position of a). The same procedure can be applied in other distances either closer or further from to the mirror. Also in each different distance the mirror can be rotated around its x axis forward or backward.

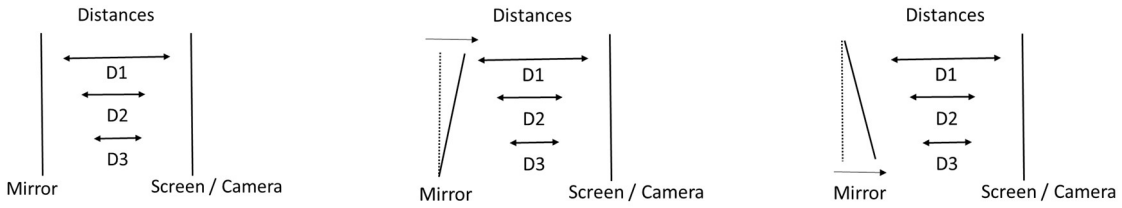


Figure 30: The position of the mirror w.r.t the screen and the camera a) in parallel b) rotated to the left c) rotated to the right

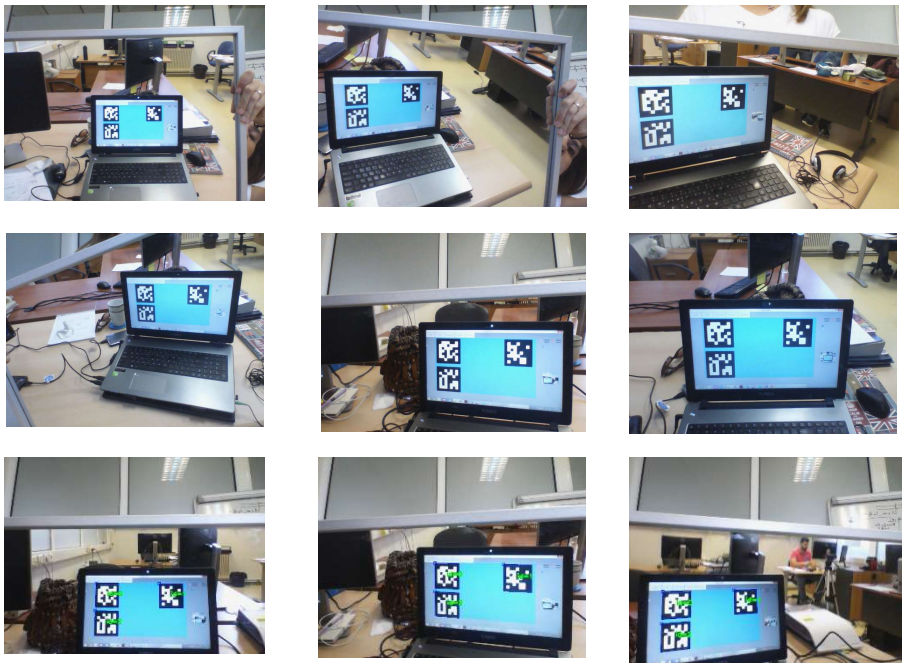


Figure 31: The screen captured via mirror in various distances and orientations and the three points detected in the screen through the application.

Lastly, the technical assistant picks up the chess board and captures 25 pictures. The whole chessboard must be inside the two monitor when the pictures are captured (Figure 32). The details of the purpose for this calibration, having to do with calibrating different cameras of different PAs (e.g. between a robot and a fixed monitor), is detailed in Deliverable D5.1.

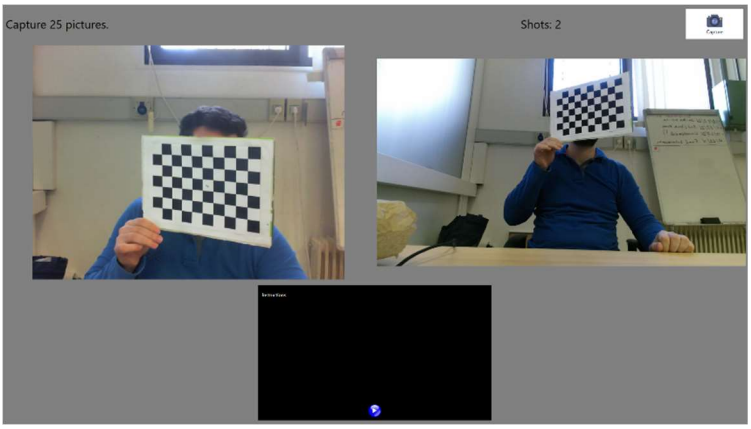


Figure 32: **Chessboard calibration**

4.4.5 Technical Background

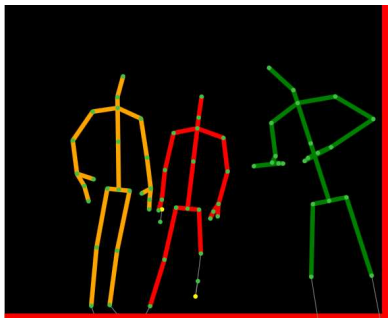
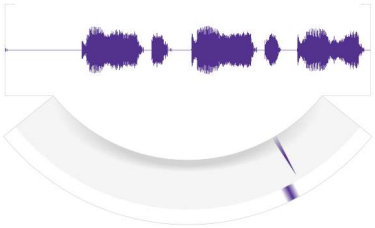
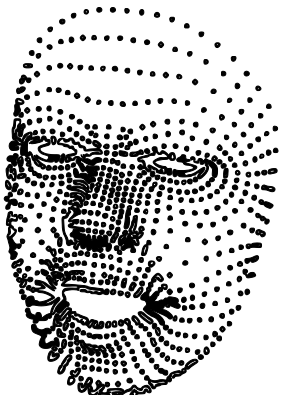
Kinect v2 was used for the data gathering. A Windows-compatible version of the new Kinect, Kinect for Windows v2, was released on July 15, 2014 alongside the Kinect for Windows SDK 2.0. The improvements of Kinect v2 in comparison with Kinect v1 are summarized in Table 9. Five different streams of information are provided form Kinect v2. Namely, those streams are the colour, depth, infrared, body skeleton and audio stream. A visual representation of the later streams is depicted in Table 10. In the following subsections, further details about the stored information for all the four streams is provided.

Table 9: Kinect v1 and v2 technical differences

	Kinect v1	Kinect v2
Depth range	0.4m → 4.0m	0.5m → 4.5m
Colour stream	640×480@30fps	1920×1080@30fps
Depth stream	320×240	512×424
Infrared stream	None	512×424
Type of Light	Structured Light (Light Coding)	Time of Flight (ToF)
Audio stream	4-mic array 16 kHz	4-mic array 48 kHz
USB	2.0	3.0
# Bodies Tracked	2 (+4)	6
# Joints	20	25
Hand Tracking	External tools	Yes
Face Tracking	Yes	Yes + Expressions

FOV	57° H 43° V	70° H 60° V

Table 10: Kinect sources

Colour	Infrared	Depth
		
Body	**Audio**	**HD face**
		

4.4.5.1 Colour, Depth, Infrared

In the colour stream a video frame of dimensions 1080x1920 is returned. Along with the visual content a Kinect timestamp is also provided and stored. For reasons of later synchronization of colour visual content with an external non-Kinect v2 stream (webcam stream) a Coordinated Universal Time (UTC) is also stored for each frame. For depth and infrared streams, a video frame of 512x424 is stored. Also, for the latter two streams, a Kinect v2 timestamp is stored. As far as the data format of the storage data, colour, depth and infrared information are stored all together in a binary file. The UTC information of the colour stream is recorded into a text file.

4.4.5.2 Kinect body skeleton

Kinect v2 provides the capability of body skeleton estimation and tracking. For a visual representation of the extracted Kinect v2 body skeletons look at the bottom left image in Table 10. Kinect v2 offers the capability to detect and track up to six different actors simultaneously. For each tracked body skeleton a sequence of information was stored into a text file adopting a JSON data structure storage style, see Figure 34. For reasons of better organisation, three levels of skeleton semantics are defined.

First level semantics: In the first level of semantics, general description of the captured Kinect skeleton frame is included. The JSON properties that refers to this level of description are depicted in the right part of Figure 34. Also a detailed description of the JSON properties can be found in Table 11. Finally, in this level of semantics, a parent property node of information named

"DataSkeletonSubSkeleton" is included. This property contains information of the recorded skeletons described in further detail in the second level of semantics.

Second level of semantics: In the second level of semantics, a general description of the tracked skeletons is given. See middle part of Figure 34, for a JSON description of this level of semantics. Here, a detailed description of each property is given in Table 12. Finally, a parent property node, named "DataSkeletonSubSkeletonJoints" that refers to the skeleton nodes and is described in the next level is also included.

Third level of semantics: In this level of semantics, information about the 25 skeleton joints is provided. Figure 33 depicts the skeleton node position returned by Kinect v2. For more details about the JSON properties and their detailed description see the right part of Figure 34 and Table 13 respectively.

Table 11: First level body skeleton semantics details.

First level of semantics	Description
LineIndexInFile	An index stored for database ordering.
FrameIndex	A Kinect-assigned index of the frame in which a skeleton was extracted.
FrameTimeStamp	A Coordinated Universal Time (UTC).
FrameTimeStampKinect	A Kinect-assigned timestamp.
UserCount	Number of recorded skeletons.
FloorVectorX	The X-coordinate of floor plane.
FloorVectorY	The Y-coordinate of floor plane.
FloorVectorZ	The Z-coordinate of floor plane.
FloorVectorW	The w-coordinate of floor plane.
DataSkeletonSubSkeleton	Related information of each tracked skeleton.

Table 12: Second level body skeleton semantics details.

Second level of semantics	Description
SkeletonLeanX	The X-coordinate of lean vector.
SkeletonLeanY	The Y-coordinate of lean vector.
SkeletonID	The Z-coordinate of lean vector.
SkeletonState	Three different values: *Tracked*, *PositionOnly*, or *NonTracked*
DataSkeletonSubSkeletonJoints	Information related to the skeleton joints.

Table 13: Third level of body skeleton semantics details.

Third level of semantics	Description
JointPositionX	The X-coordinate of position vector.
JointPositionY	The Y-coordinate of position vector.
JointPositionZ	The W-coordinate of position vector.
JointRotationX	The X-coordinate of orientation quaternion.
JointRotationY	The Y-coordinate of orientation quaternion.
JointRotationZ	The Z-coordinate of orientation quaternion.
JointRotationW	The W-coordinate of orientation quaternion.
SkeletonJointConfidence	??
SkeletonJointState	??

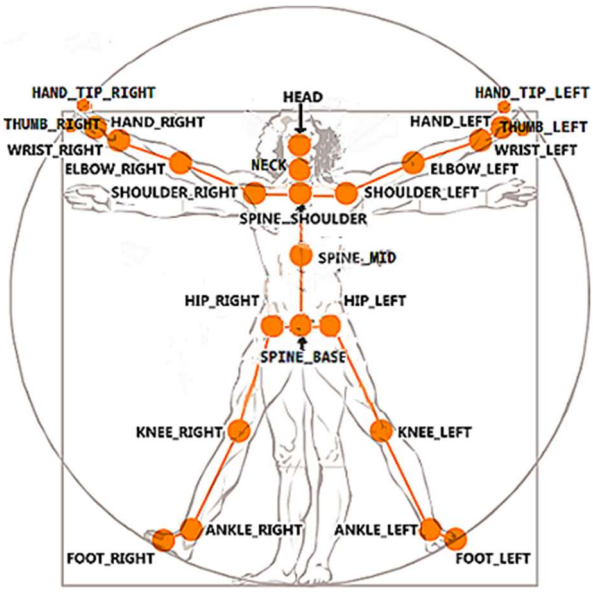


Figure 33: Kinect v2 skeleton joint position

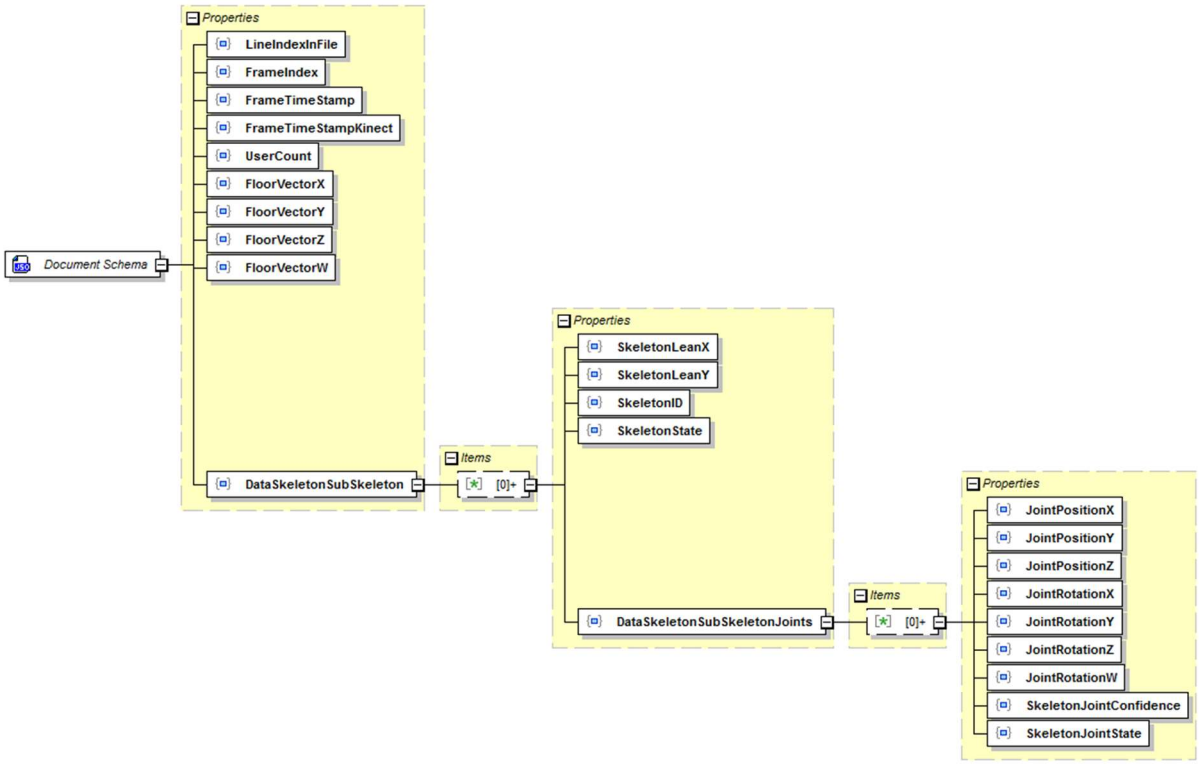


Figure 34: Body skeleton JSON scheme

4.4.5.3 Kinect face landmarks

Similar to the skeleton semantics described in Section 4.4.5.2, face landmarks semantics are also described in two levels

First level: In the first level of semantics, a general description of the face Kinect captured frame is given. Properties like, the Kinect frame index, the number of faces detected, etc., are included. The JSON scheme of these properties is given in Figure 35 .

Second level: In the second level of semantics, face related information is included. For instance, for each face bounding boxes that enclose the face, the eye of other face parts is stored using geometric coordinates. More specifically, for the face bounding box, two points are needed to be stored, the upper left $(faceColorBBoxTopLeftX, faceColorBBoxTopLeftY)$ and bottom right $(faceColorBBoxBottomLeftX, faceColorBBoxBottomLeftY)$ box points.

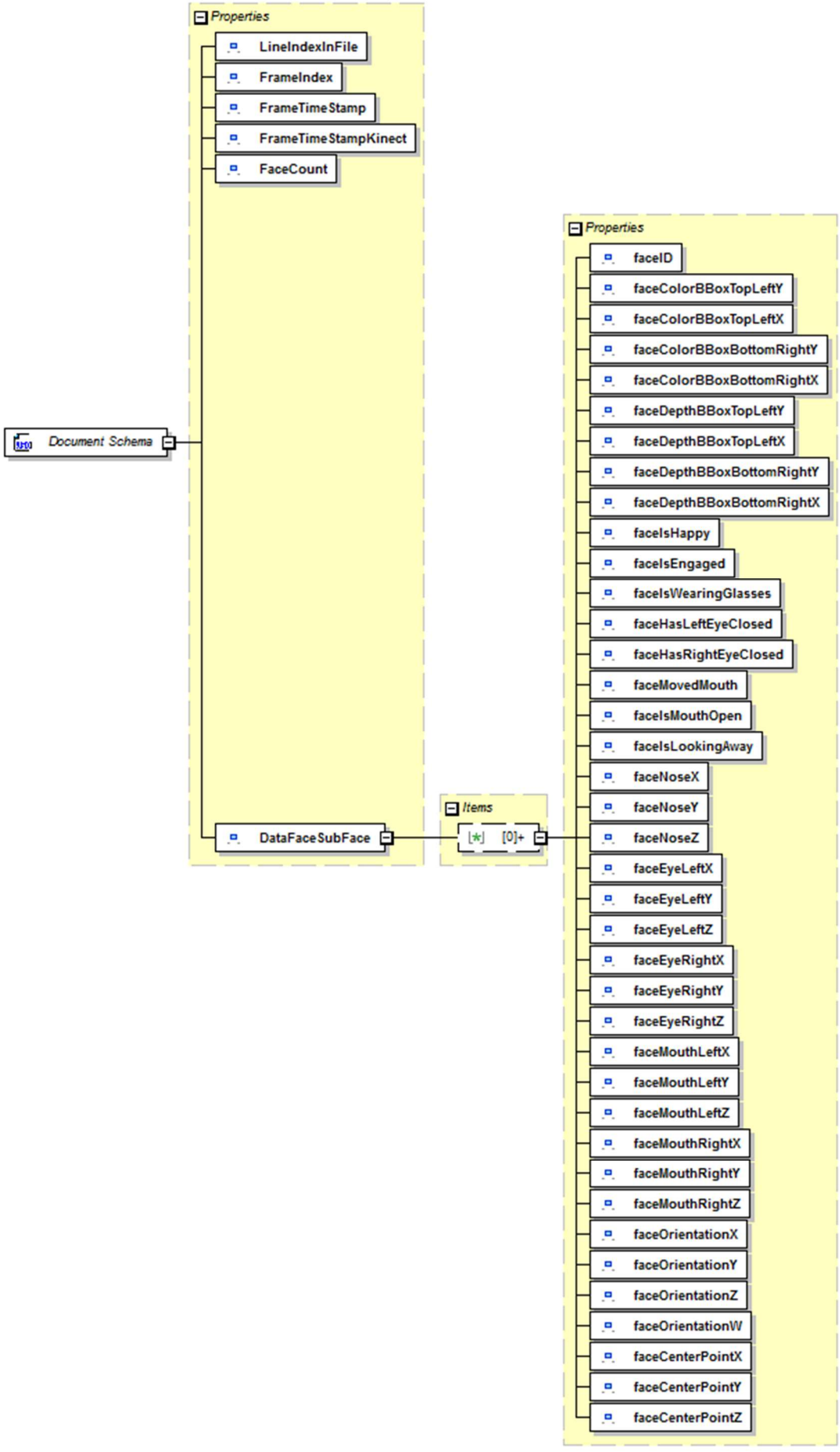


Figure 35: **Face landmarks JSON scheme**

4.4.5.4 HD face

The data structure of HD face semantics is simpler than the ones described in Sections 4.4.5.2 and 4.4.5.3. In the first level, general information of the HD face is contained, like the HD face position, and orientation while in the second level, a sequence of 1347 points is stored. For the entire JSON scheme see Figure 36.

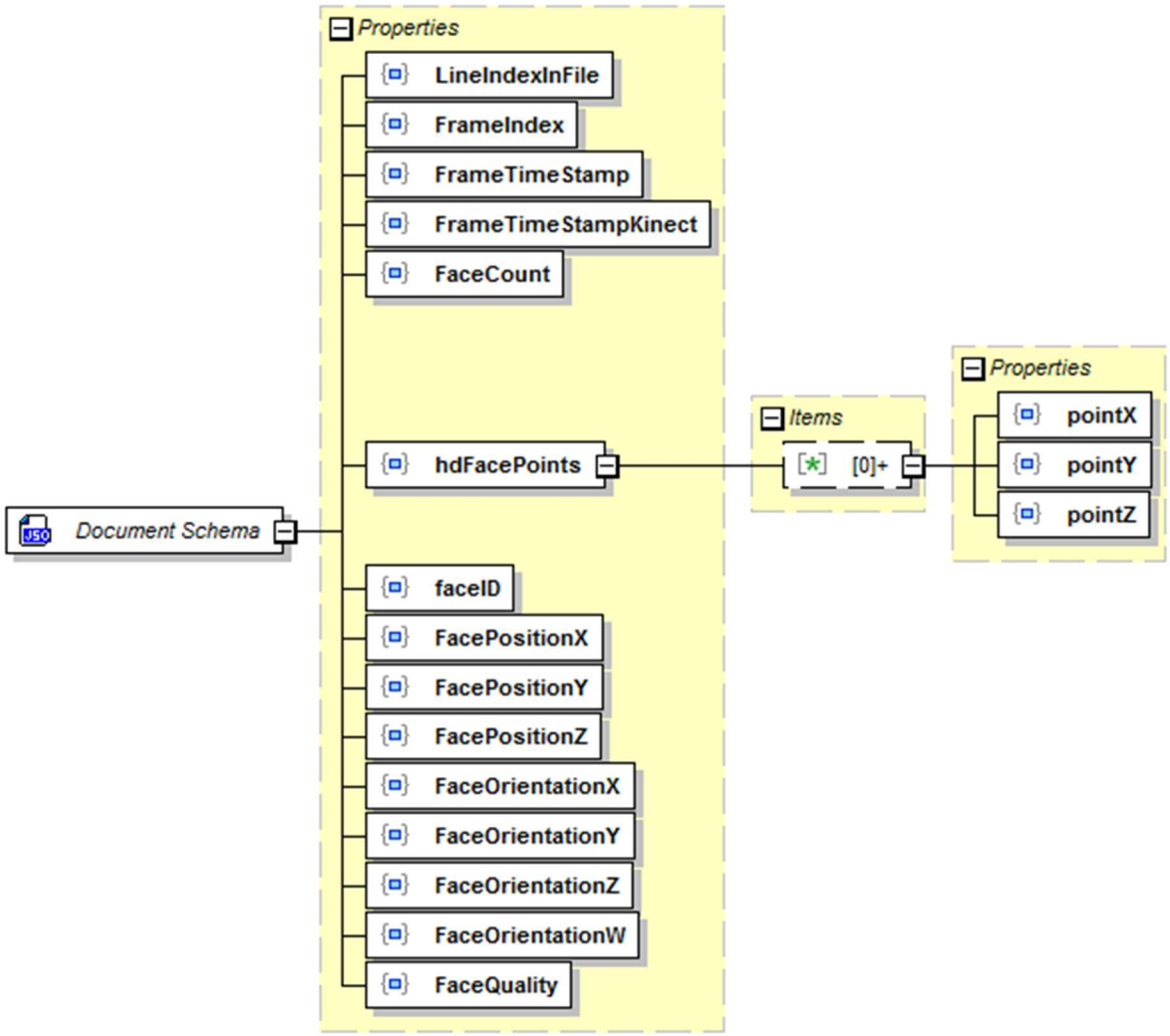


Figure 36: HD face landmarks JSON scheme

4.4.5.5 Kinect Audio

Audio: Single channel audio was recorded using Kinect v2. Audio information was stored into a binary file where raw data (without header file) is included. In addition to the audio information semantics like the beam direction, which is the direction of source sound and speaker ID estimation is also stored. For a visual description of the aforementioned see Figure 38 . For the JSON scheme see Figure 37.

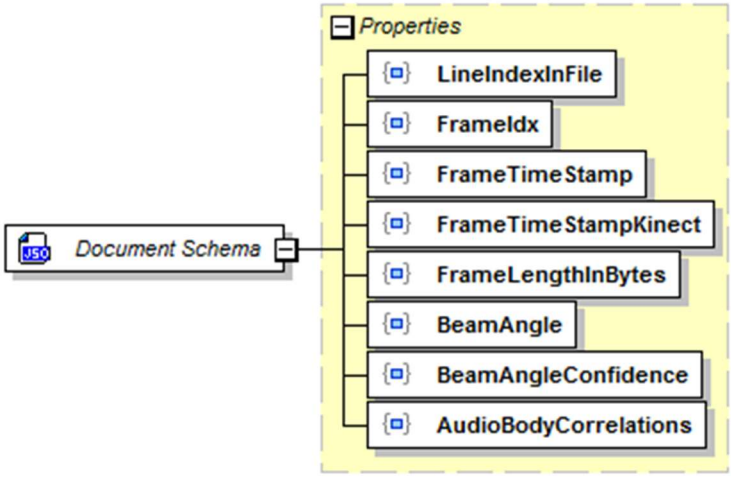


Figure 37: Audio JSON scheme

Figure 38: Kinect v2 audio

4.5 Database description

As described in Sections 4.4.5.3, 4.4.5.4, 4.4.5.5 the data obtained from sensor streams were stored into a JSON storage format. For reasons of better indexing and retrieval, a SQL light database was created to store this information. The form of the row items stored in the database are depicted in Table 14. Also a detailed description of each attribute is given in Table 15. Having this type of indexing scheme, specific parts of the recorded streams can be retrieved in a specified time interval.

Table 14: Database row item

ID	Kinect Timestamp	SessionID	SkeletonLineIDX	FaceLineIDX	HDfaceLineIDX	SoundLineIDX	UTC

Table 15: Database attributes

Database attributes	Description
ID	The unique DB identifier of the stored item
Kinect Timestamp	The Kinect timestamp for a specific recorded

	Kinect stream, e.g., body skeleton.
SessionID	The ID of each recorded session.
SkeletonLineIDX	An index that defines the order of the recorded row item for skeleton stream.
FaceLineIDX	An index that defines the order of the recorded row item for face landmarks stream.
HDfaceLineIDX	An index that defines the order of the recorded row item for HD face stream.
SoundLineIDX	An index that defines the order of the recorded row item for audio stream.
UTC	A Universal Time Count, assigned along with each row item and defines the time of the row item record.

5. Implementation Strategy of the Sensorial Component on the MaTHiSiS Platform Agents

This section describes the local implementation strategy of the SC pre-processing unit of raw data to affect related features on the various MaTHiSiS PAs.

Table 16: Supporting Sensorial Component modalities per platform agent

Platform Agent Layer	Platform Agent	Supported SC modality
Robotic	NAO	Facial expressions analysis, Gaze estimation Skeleton Motion Analysis (coupled with a Kinect), Speech recognition and speech-based affect recognition
	TurtleBot	Face emotion recognition
Mobile devices	Android (version 5 and newer) and iOS (version 9 and newer) mobile devices (smartphones and tablets)	Mobile device-based emotion recognition
IWB	Surface Hub	Facial expressions analysis, Gaze estimation, Skeleton Motion Analysis, Speech recognition and speech-based affect recognition

5.1 Robotic Layer

The robotic layer in MaTHiSiS consists of the Turtlebot and NAO robots where the modalities described on this document are going to be deployed. At this section, the capabilities and deficiencies of each PA regarding the implementation of the modalities will be introduced. In the case of any shortcomings a pre-designed strategy plan will be employed in order to fall back to a method that will overcome the shortcomings. It is possible that some issues may remain unresolved, while others will be solved partially.

5.1.1 NAO

In this section, NAO's mounted and accompanying sensorial resources will be detailed, along with a plan of how to exploit them so as to implement the algorithms described in section 2 on the NAO robot. Likewise, a strategy plan is introduced regarding issues that have occurred with one of the modalities.

For Facial Expression Analysis and Gaze Estimation, the PA can use its RGB camera, where the captured images could be processed accordingly and infer affect-related features for the facial expression and gaze estimation. For Speech Recognition and Speech-Based affect recognition, its microphones could be used in order to process the tracked sounds from the user. However, the Skeleton Motion Analysis module is a more demanding task for this PA because the robot itself has no sensors or other means to extract the skeleton of a user. To overcome this issue a Kinect V2 sensor is correlated, and calibrated with NAO, providing it with the skeleton of a user and enabling the analysis of its motion on the PA. At this moment, this setup requires the stillness of the PA, but future implementations will focus on achieving the same correlation while the robot is moving.

Notice that NAO supports the (wireless or wired) connection with a PC where processes could be executed remotely. Therefore, the complexity or the high processing requirements of an algorithm, will not constrain the implementation of the modules on the PA.

In future versions, depending on the scenarios, the required SC components implemented on NAO could vary. Initially, face (using webcam data) and speech could already be implemented. However, in order to include skeleton analysis some extra implementations and modifications will be needed, such as the addition of a Kinect v2 as a new sensor of the robot. The methodology to achieve this is reported in Deliverable 5.1 – *Description of the robotic layer*.

5.1.2 TurtleBot

The available scenarios in TurtleBot don't allow for affect recognition through face, speech, gaze or skeleton data since these scenarios are based on the follower ability and the use of the camera to perform colour recognition. For this reason, TurtleBot won't use any SC data during the first release, even though face emotion analysis has already been implemented. However, this PA will be able to adapt the learning content using affect understanding through interactions with the learning material (78).

.

5.2 Mobile devices layer

Mobile device-based emotion recognition is based on inertial sensors' data and touch screen information. In order to detect the emotional state of the learners using inertial data, MaTHiSiS will make use of gyroscope and accelerometer. Also, information acquired through the touch screen will be analysed to obtain useful information. This information will be based on touches on the screen (tapping, long pressing or swiping).

5.3 Interactive White Board layer

The specifications of the Surface Hub (MaTHiSiS core IWB) relative to sensors are:

- Kinect compatible
- 2 wide angle HD cameras 1080p @ 30fps
- Infrared presence sensors
- Ambient light sensors
- Touch screen
- Stereo speakers

- Microphone array

According with section 2, there are 5 types of Sensorial component modalities. From them **Facial expressions analysis** and **Gaze estimation** can be covered by the 2 built in cameras of the IWB to capture images that later would be processed. **Skeleton Motion Analysis** couldn't be done the same way with the built in cameras, but the IWB is fully compatible with Kinect and it could be used to enable the use of this Sensorial component modality.

For **Speech recognition and speech-based affect recognition** the IWB counts with a good quality microphone array so tracking sounds from the user wouldn't be a problem.

With **Mobile device-based emotion recognition** there is a limitation, as the IWB doesn't contain an accelerometer or gyroscope. That means that any tasks that require inertia sensory data won't be possible. Any tactile requirements will be available from the device's touch screen.

6. Conclusion

This document reported on the first implementation and first results of the modalities that comprise the MaTHiSiS Sensorial Component. Namely, the current literature was reviewed and specific methods for facial expression analysis, gaze estimation, skeleton motion detection, speech recognition and speech-based emotion, as well as emotion recognition from mobile-based inertia sensor data were developed.

The outputs of each modality, comprising of distinct affective cues, as well as the homogenisation and synchronisation strategy adopted in order to feed these outputs to the multimodal fusion mechanism of WP4 (T4.3), were also discussed in this document. In addition, the implementation of these modalities on the different MaTHiSiS Platform Agents was discussed, due to the different sensorial and computational capacities of the PAs, with an interest to clarify and provide a strategy on how these can be leveraged in order to accommodate all SC modalities.

During the first release, the methodologies adopted and machine learning algorithms developed were both trained and tested with data from public datasets, where available, due to the lack of real data from MaTHiSiS users. On one hand, this allowed consolidation of the developed algorithms with widely-accepted benchmarks and have a rich basis of data to train the algorithms. On the other hand, the data in currently available public datasets do not represent neither the significant diversity of users/use cases that MaTHiSiS applies to, nor – in most cases – the extent of 'in the wild' settings that MaTHiSiS will be delivered in. In the case of newer technologies, like learning from mobile inertia sensors, no data was even available, therefore MaTHiSiS is called to pioneer in this field.

To this end, an extensive data acquisition task is organised within WP4, in order to collect data for all WP4 modules, to help train and implement the algorithms developed in real MaTHiSiS settings and cases. This task will take place around M13 of the project and will result to a first comprehensive dataset for all SC modalities, but also for T4.2 and T4.3 methods. This dataset will be used to train WP4 algorithms, which will be subsequently used (and tested) during the MaTHiSiS driver pilots. This first set of data will then be further enriched from data collected in the pilots and along with the test results will guide the implementation of the second release of the approach presented in this document.

7. References

1. *Static and dynamic 3D facial expression recognition: A comprehensive survey.* **Sandbach, G, et al., et al.** 2012, Image and Vision Computing, Vol. 30 (10), pp. 683-697.

2. *A survey of affect recognitionmethods:Audio, visual, and spontaneous expressions.* **Zeng, Z., et al., et al.** 2009. Pattern Analysis andMachine Intelligence.

3. *Communication and automatic interpretation of affect.* **Salah, A. A., Sebe, N. and Gevers, T.** 2010. Affective Computing and Interaction: Psychological, Cognitive.

4. *Automatic analysis of facial affect: A survey of registration, representation and recognition.* **Sariyanidi, Evangelos, Gunes, Hatice and Cavallaro, Andrea.** 2015. IEEE Transactions on Pattern Analysis andMachine Intelligence.

5. *Survey on rgb, 3d, thermal, and multimodal approaches for facial expression recognition: History, trends, and affect-affectrelated.* **Corneanu, C.A., et al., et al.** 2016. Pattern Analysis and Machine Intelligence, IEEE Transactions on.

6. **Schmidt, K. L. and Cohn, J. F.** Human facial expressions as adaptations: Evolutionary perspectives in facial expression research. *Yearbook of Physical Anthropology.* 2001, pp. 116:88-24.

7. **Goss, H. Gray and C. M.** *Anatomy of the human body.* s.l. : 28th ed. Lea Febiger, 1966.

8. **Burrows, AnneM. and Cohn, JeffreyF.** Comparative anatomy of the face. *Encyclopedia of.* 2014, pp. 1-10.

9. *A selective meta-analysis on the relative incidence of discrete affective states during learning with technology.* **D'Mello, S.** 2013, Journal of Educational Psychology, Vol. 105(4), pp. 1082-1099.

10. *Automatic detection of learning-centered affective states in the wild.* **Bosch, Nigel, et al., et al.** 2015, pp. 379-388.

11. *"Supervised Descent Method and Its Applications to Face Alignment,".* **Torre, X. Xiong and F. De la.** Portland : s.n., 2013. 2013 IEEE Conference on Computer Vision and Pattern Recognition. pp. 532-539.

12. *Support-vector networks.* **Cortes, Corinna and Vapnik, Vladimir.** 1995, Machine Learning, pp. 273-297.

13. *Fully automatic 3D facial expression recognition using differential mean curvature maps and histograms of oriented gradients.* **P. Lemaire, M. Ardabilian, L. Chen and M. Daoudi.** Shanghai : s.n., 2013. 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG).

14. *3D facial expression recognition using SIFT descriptors of automatically detected keypoints.* **Berretti, Stefano, et al., et al.** 2011, The Visual Computer, Vol. 27, p. 1021.

15. **Cohn, J. M. Girard and J. F., et al., et al.** Social risk and depression: Evidence from manual and automatic facial expression analysis. *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG).* 2013, pp. 1-8.

16. *Facial Action Coding System: A Technique for the Measurement of Facial Movement.* **Ekman, P. and Friesen, W.** Palo Alto : s.n., 1978. Consulting Psychologists Press.

17. *Better to Be Frustrated Than Bored: The Incidence, Persistence, and Impact of Learners' Cognitive-affective States During Interactions with Three Different Computer-based Learning Environments.* **Baker, Ryan S. J. d., et al., et al.** 2010, Int. J. Hum.-Comput. Stud., Vol. 68, pp. 223-241.

18. **Bishop, Christopher M.** *Pattern recognition.* s.l. : Machine Learning 128 , 2006.

19. *Deep Deformation Network for Object Landmark Localization.* **Yu, Xiang, Zhou, Feng and Chandraker, Manmohan.** Amsterdam : s.n., 2016. Computer Vision – ECCV 2016: 14th European Conference.

20. *Long Short-Term Memory.* **Hochreiter, Sepp and Schmidhuber, Jurgen.** 1997, Neural Comput, Vol. 9, pp. 1735-1780.

21. *The opencv library.* **Bradski, G.** s.l., 2000, Dr. Dobb's Journal of Software Tools.

22. *Histograms of oriented gradients for human detection.* **Dalal, N. and Triggs, B.** 2005, In 2005 IEEE Computer.

23. **King, D. E.** *Max-margin object detection.* s.l. : CoRR, abs/1502.00046., 2015.

24. *Accurate scale estimation for robust visual tracking.* **Danelljan, M., et al., et al.** 2014. In Proceedings of the British Machine Vision Conference. BMVA Press.

25. *One millisecond face alignment with an ensemble of regression trees.* **Kazemi, Vahid, and Josephine Sullivan.** 2014. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

26. *Fisher vector faces in the wild.* **Simonyan, K., et al., et al.** 2013. in BMVC, vol. 2, no. 3, p. 4. 4.

27. *A compact and discriminative face track descriptor.* **Parkh, O. M., et al., et al.** 2014. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1693–1700. 4.

28. *In the eye of the beholder: A survey of models for eyes.* **Hansen, Witzner, Dan and Ji, and Qiang.** Pattern Analysis andMachine Intelligence, IEEE Transactions, p. 2010.

29. *Sparse and semi-supervised visual.* **Oliver Williams, Andrew Blake, and Roberto Cipolla.** 2006. In Computer Vision and Pattern Recognition IEEE Computer Society Conference.

30. *Appearance-based eye gaze estimation.* **Kar-Han Tan, David J Kriegman, and Narendra Ahuja.** 2002. In Applications of Computer Vision Sixth IEEE .

31. *Inferring human gaze from appearance via.* **Feng Lu, Y. Sugano, T. Okabe, and Y. Sato.** 2011. Computer Vision (ICCV), IEEE International Conference.

32. *Eye gaze tracking techniques for interactive applications .* **RMMimica, Carlos HMorimoto andMarcio.** 2005. Computer Vision and Image Understanding.

33. *Appearance-based gaze estimation in the wild.* **Xucong Zhang, Yusuke Sugano,Mario Fritz, and Andreas Bulling.** 2015.

34. *Eye tracking for Everyone.* **Krafka, K., et al., et al.** 2016. Proceedings of the IEEE, Conference on Computer Vision and Pattern Recognition.

35. *Gaze estimation from multimodal Kinect data.* **Odobez and Jean-Marc, Kenneth Alberto FunesMora and.** 2012. In Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE Computer Society Conference on, pages 25–30. IEEE, 2012.

36. *"A tutorial survey of architectures, algorithms, and applications for deep learning.".* **Deng, Li.** 2014. APSIPA Transactions on Signal and Information Processing.

37. *Deep learning.* **LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton.** 2015, Nature, pp. 436-444.

38. *Face detection using surf cascade.* **Li, Jianguo, Tao Wang, and Yimin Zhang.** 2011. Computer Vision Workshops (ICCV Workshops), IEEE International Conference on IEEE.

39. *Model-based object pose in 25 lines of code.* **Dementhon, Daniel F., and Larry S. Davis.** 1995, International journal of computer vision , Vols. 15.1-2, pp. 123-141.

40. *Effective face frontalization in unconstrained images.* **Hassner, Tal, et al., et al.** 2015, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

41. *Identifying emotions expressed by mobile users through 2D surface and 3D motion gestures.* **Coutrix, Céline and al, et.** 2012. Proceedings of the 2012 ACM Conference on Ubiquitous Computing. ACM.

42. *Exploring emotional preference for smartphone applications.* **Kim, H. J. and Choi, Y. S.** 2012 . In 2012 IEEE Consumer Communications and Networking Conference (CCNC) (pp. 245-249). IEEE.

43. *A touch based affective user interface for smartphone.* **Kim, Mira and al., et.** 2013 . IEEE International Conference on Consumer Electronics (ICCE). IEEE, 2013.

44. *Toward e-motion-based music retrieval a study of affective gesture recognition.* **Amelynck, Denis and al., et.** 2012. IEEE transactions on affective computing 3.2 .

45. *Motion-based recognition: a survey.* **Cedras, M. and Shah, M.** 1995, IEEE Transactions on Image and Vision Computing, Vol. 13, pp. 129-155.

46. *Human motion analysis: a review.* **Aggarwal, J.K. and Cai , Q.** 1997. Nonrigid and Articulated Motion Workshop.

47. *The Visual Analysis of Human Movement: A Survey.* **Gavrila, D. M. .** 1999, IEEE Transactions on Computer Vision and Image Understanding, Vol. 73, pp. 82-98.

48. *A Survey of Computer Vision-Based Human Motion Capture.* **Thomas, B. M. and Granum, E.** 2001, IEEE Transactions on Computer Vision and Image Understanding, Vol. 81, pp. 231-268.

49. *Human Activity Analysis: A Review.* **Aggarwal, J.K and Ryoo , M.S.** 2011, IEEE Transactions on ACM Computing Surveys, Vol. 43, pp. 16:1-16:43.

50. *A survey of vision-based methods for action representation, segmentation and recognition.* **Weinland, D.** 2011, IEEE Transactions on Computer Vision and Image Understanding, p. 115.

51. *Slow Feature Analysis for Human Action Recognition.* **Zhang , Z. and Tao, D.** 2012, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 34, pp. 436-450.

52. *Learning realistic human actions from movies.* **Laptev, I, et al., et al.** Computer Vision and Pattern Recognition, p. 2008.

53. *Slow Feature Analysis for Human Action Recognition.* **Zhang, Z and Tao, D.** 2012, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 34, pp. 436-450.

54. *Learning realistic human actions from movies.* **Laptev, I, et al., et al.** 2008, Computer Vision and Pattern Recognition.

55. *A spatio-temporal descriptor based on 3d-gradients.* **AKläser, A, Marszałek, M. and Schmid, C.** 2008, British Machine Vision Conference.

56. *Space-time Interest Points.* **Laptev, I and Lindeberg, T.** 2003. International Conference on Computer Vision.

57. *Behavior Recognition via Sparse Spatio-temporal Features.* **Dollar, P, et al., et al.** 2005. International Conference on Computer Communications and Networks.

58. *Human action recognition with spatiotemporal salient points.* **Oikonomopoulos, A, Patras, I and Pantic, M.** 2003, IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics, pp. 25-28.

59. *Scale Saliency: a novel approach to salient feature and scale selection.* **Kadir, T and Brady, M.** 2003. International Conference on Visual Information Engineering.

60. *Recognizing Human Actions: A Local SVM Approach.* **Schuldt, C, Laptev, I and Caputo, B.** 2004. International Conference on Pattern Recognition.

61. *Slow Feature Analysis: Unsupervised Learning of Invariances.* **Wiskott, L and Sejnowski, T.J.** 2002, IEEE Transactions on Neural Computation, Vol. 14, pp. 715-770.

62. *DL-SFA: Deeply-Learned Slow Feature Analysis for Action Recognition.* **Sun, L, et al., et al.** 2014. Computer Vision and Pattern Recognition.

63. *Learning Skeleton Stream Patterns with Slow Feature Analysis for Action Recognition.* **Shan, Y, Zhang, Z and Huang, K.** 2015. European Conference on Computer Vision Workshops .

64. *Li, Wanqing;Zhang, Zhengyou; Liu, Zicheng.* **points, Action recognition based on a bag of 3d.** 2010. Computer Vision and Pattern Recognition.

65. *Learning skeleton stream patterns with slow feature analysis for action recognition.* **Shan, Yanhu, Zhang, Zhang and Huang, Kaiqi.** 2014. European Conference on Computer Vision.

66. *Mining actionlet ensemble for action recognition with depth cameras.* **Wang, Jiang, Zicheng , Wu and Ying, Yuan.** 2012. Computer Vision and Pattern Recognition (CVPR).

67. *Features and classifiers for emotion recognition from speech: a survey from 2000 to 2011.* **Anagnostopoulos, C.N. and Iliou, T., Giannoukos, I.** 2015, Artificial Intelligence Review 43(2), pp. 155–177.

68. *Survey on speech emotion recognition: Features, classification schemes, and databases.* **M., El Ayadi, Kamel, M.S. and Karray, F.** 2011, Pattern Recognition, Vol. 44(3), pp. 572– 587.

69. *Introduction to Audio Analysis: A MATLAB® Approach.* **T., Giannakopoulos and A., Pikrakis.** 2014. Academic Press.

70. **S., Theodoridis and K, Koutroumbas.** *Pattern Recognition, Fourth Edition.* s.l. : Academic Press, Inc., 2008.

71. **K., Hyoung-Gook, M., Nicolas and T., Sikora.** *MPEG-7 Audio and Beyond: Audio Content Indexing and Retrieval.* s.l. : John Wiley & Sons, 2005.

72. *Statistical learning theory.* **Vapnik, V.** vol. 1. Wiley New York (1998) : s.n., 1998.

73. *Emovo corpus: an italian emotional speech database. In: Chair), N.C.C., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (eds.) .* **Costantini, G., et al., et al.** Reykjavik, Iceland : s.n., 2014. Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14).European Language Resources Association (ELRA).

74. *Speaker-dependent audio-visual emotion recognition. In: Proc. Int. Conf. on Auditory-Visual Speech Processing (AVSP'08).* **Haq, S. and Jackson, P.** Norwich, UK : s.n., 2009.

75. *A database of german emotional speech. .* **Burkhardt, F., et al., et al.** Lissabon : s.n., 2005. in Proceedings of Interspeech.

76. *Flow and the psychology of discovery and invention.* **Csikszentmihalyi, Mihaly.** 1996. New Yprk: Harper Collins.

77. *A New Mirror-based Extrinsic Camera Calibration Using an Orthogonality Constraint.* **Takahashi, K. and Matsuyama, S. Nobuhara and T.** 2012. Computer Vision and Pattern Recognition (CVPR), 2012,IEEE Conference on. IEEE, 2012.

78. *D5.1 – Description of the robotic layer.* s.l. : Maastricht University, 2017.

79. *Spontaneous facial expression analysis based on temperature changes and head motions.* **Yin, Peng Liu and Lijun.** Ljubljana : s.n., 2015. 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG).

80. *Fully automatic 3D facial expression recognition using local depth features.* **Mingliang Xue, A. Mian, Wanquan Liu and Ling Li.** Steamboat Springs : s.n., 2014. IEEE Winter Conference on Applications of Computer Vision.

81. *LSTM-Modeling of continuous emotions in an audiovisual affect recognition framework.* **Martin Wöllmer, Moritz Kaiser, Florian Eyben, Björn Schuller, Gerhard Rigoll.** 2013, Image and Vision Computing, Vol. 31, pp. 153-163.

82. **He, Lang and Jiang, Dongmei and Yang, Le and Pei, Ercheng and Wu, Peng and Sahli, Hichem.** Multimodal Affective Dimension Prediction Using Deep Bidirectional Long Short-Term Memory Recurrent Neural Networks. *Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge.* Brisbane, Australia : ACM, 2015, pp. 73-80.

83. *Deep learning for real-time robust facial expression recognition on a smartphone.* **I. Song, H. J. Kim and P. B. Jeon.** Las Vegas : s.n., 2014. IEEE International Conference on Consumer Electronics (ICCE).

84. **Kahou, Samira Ebrahimi and Pal, Christopher and Bouthillier, Xavier and Froumenty, Pierre and G\"{u}l\c{c}ehre, \c{C}aglar and Memisevic, Roland and Vincent, Pascal and Courville, Aaron and Bengio, Yoshua and Ferrari, Raul Chandias and Mirza, Mehdi and Je.** Combining Modality Specific Deep Neural Networks for Emotion Recognition in Video. *Proceedings of the 15th ACM on International Conference on Multimodal Interaction.* Sydney, Australia : ACM, 2013, pp. 543-550.

85. *Emotion recognition in the wild.* **A. Dhall, R. Goecke, J. Joshi, M.Wagner, and T. Gedeon.** Sydney, Australia : s.n., 2013. ACM International Conference on Multimodal Interaction.

86. *Learning Expressionlets on Spatio-temporal Manifold for Dynamic Facial Expression Recognition.* **M. Liu, S. Shan, R. Wang and X. Chen.** Columbus : s.n., 2014. IEEE Conference on Computer Vision and Pattern Recognition.

87. *Facial Expression Recognition Using Kinect Depth Sensor and Convolutional Neural Networks.* **Mohan, E. P. Ijjina and C. K.** Detroit : s.n., 2014. 13th International Conference on Machine Learning and Applications.

88. *Slow Feature Analysis for Human Action Recognition.* **Zhang, Z.** 2012, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 34, pp. 436-450.

89. *In the eye of the beholder: A survey of models for eyes.* **Ji, DanWitzner Hansen and Qiang.** Pattern Analysis andMachine Intelligence, IEEE Transactions, p. 2010.

90. *Eye tracking for Everyone.* **Krafka, Kyle, et al.** 2016. Proceedings of the IEEE, Conference on Computer Vision and Pattern Recognition.

91. *Effective face frontalization in unconstrained images.* **Hassner, Tal, et al.** 2015, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

92. *A complete expression dataset for action unit and emotion-specified expression.* **Lucey, P., et al., et al.** San Francisco : s.n., 2010. Proceedings of the Third International Workshop on CVPR for Human Communicative Behavior Analysis.

93. *Collecting Large, Richly Annotated Facial-Expression Databases from Movies.* **Dhall, Abhinav, et al., et al.** IEEE Multimedia 2012.