Development of Robot-enhanced Therapy for Children with Autism Spectrum Disorders

Project No. 611391

DREAM

Development of Robot-enhanced Therapy for Children with Autism Spectrum Disorders

Agreement Type: Collaborative Project
Agreement Number: 611391

D4.1 Sensorized Therapy Room Design and Algorithms for Data Sensing and Interpretation

Due Date: **01/04/2015**
Submission date: **01/04/2015**

Start date of project: **01/04/2014** Duration: **54 months**

Organisation name of lead contractor for this deliverable: **University of Portsmouth**

Responsible Person: **Honghai Liu** Revision: **4.0**

Project co-funded by the European Commission within the Seventh Framework Programme		
Dissemination Level		
PU	Public	**PU**
PP	Restricted to other programme participants (including the Commission Service)	
RE	Restricted to a group specified by the consortium (including the Commission Service)	
CO	Confidential, only for members of the consortium (including the Commission Service)	

I. Contents

1. Executive Summary

Deliverable D4.1 provides information for sensorized therapy room design and algorithms for data sensing and interpretation. This deliverable documents the development of the DREAM infrastructure for sensory data collection. It provides an overview the state of the art in sensory data acquisition & analysis, and sets out the specification, design, implementation, and testing of the DREAM sensory system based on the requirements set out in deliverable D1.1. This deliverable contain results from primarily from task T4.1 with additional input from task T4.2.

2. Principal Contributors

The main authors of this deliverable are as follows (in alphabetical order)

Haibin Cai, University of Portsmouth
Yinfeng Fang, University of Portsmouth
Dongxu Gao, University of Portsmouth
Xiaodong Jiang, University of Portsmouth
Zhaojie Ju, University of Portsmouth
Honghai Liu, University of Portsmouth
Ting Wang, University of Portsmouth
Yiming Wang, University of Portsmouth
Hui Yu, University of Portsmouth
Wei Zeng, University of Portsmouth
Shu Zhang, University of Portsmouth

3. Revision History

Version 0.1 (Jiang, X., Cai, H., Fang, Y. 25-01-2015)
Version 0.8 (Ju, Z. 28-01-2015)
Version 1.0 (Liu, H. 29-01-2015)
First draft
Version 1.1 (Ju, Z., Jiang, X. 30-01-2015)
Updated the fixing kits for Kinects and Cameras with more screw sets.
Version 1.4 (Ju, Z., Cai, H., Jiang, X. Fang, Y. 16-02-2015)
Updated the camera configuration, camera fixing and software installation.
Version 1.8 (Cai, H., Ju, Z., Gao, D. 24-02-2015)
Updated the hardware configuration and data sensing
Version 1.95 (Cai, H., Ju, Z., Gao, D. 2-03-2015)
Updated the figures showing the links between the sensors and ports.
Version 2.7 (Ju, Z., Cai, H., Gao, D., Fang, Y., Wang, Y., Zhang, S., Wang, T., 24-03-2015)
Added introduction and data interpretation.
Version 3.0 (Ju, Z., Cai, H., Gao, D., Y., Wang, Y., Zhang, S., 28-03-2015)
Added more pictures and explanation to explain the experimental results.
Version 4.0 (Ju, Z., Cai, H., Gao, D., Y., Wang, Y., Zhang, S., Yu, H., 26-09-2015)
Restructure the report; rewrite the report to reflect the tasks of the deliverable.

4. Introduction

The tasks in Deliverable 4.1 include the design of a multi-camera system for data acquisition and preliminary vision signal processing. The system will be able to capture the child movements and interactions among the child, the robot and the therapist. This deliverable thus consists of two parts: multi-camera system design; preliminary algorithms for data sensing and interpretation.

Since the designed system will be used for therapy purpose of ASD children, non-intrusive sensors are considered due to their non-contract, natural and unobstructed way of data capturing, which will significantly reduce the possible excruciating and fearful feelings of the children by caused by wearing the wearable sensors. To this end, the sensors used in this multi-camera system include video cameras and consumable depth sensors only. The system is required to be able to capture varying movements of the children and interaction including body gesture, facial expression, gaze and even speech. Thus, a small camera network is designed to meet these requirements along with a software package described in following sections.

To enable the advances of the multi-camera system to capture multiple cues of the children and the interactions with the therapist, a software package consisting a set of efficient algorithms is developed. The architecture of the software package is formed by 7 modules integrating varying task and data communication functions, which essentially includes camera pose estimation, gaze estimation, human action analysis, object tracking, facial expression analysis, speech recognition and sensor fusion. Among those modules, the pose estimation modules can set a global coordinate framework for the system through estimating the pose of each individual sensor, which enables the alignment of the captured data for further communications. The sensor fusion module synchronises and fuses multiple sensor data. Each of the rest 5 modules achieves single function as an indispensable component of multi-camera system.

The rest of the report is arranged as follows. Section 5 reviews the state-of-the-art methods for each component of the system. It intends to provide a background and related methods of the technologies used or developed in the system. Section 6 presents the detail of the design of the multi-camera system with hardware specification, installation and the installation of the software package and instructions. Section 7 describes the detail of the methods used in the system with preliminary testing results.

5. The State-of-the-Art Methods

5.1. Gaze Estimation

Gaze estimation is to estimate the gaze direction or point of regard of a person. Gaze is also an important visual cue of a child and can provide useful information for therapist to detect and interactive with child with ASD. In DREAM project, we aim to provide the real time gaze estimation of the child. Further gaze analysis based on the detected gaze direction and its spatio-temporal gaze information will also be explored. The frequency and dwell time of fixation on different parts of the robot will be calculated based on the spatio-temporal gaze information. By combining the objects position, which is in front of the robot, the detected gaze direction will be able to provide useful information for the joint attention behaviours.

Providing the head position of the robot, the mutual gaze can also be addressed using the gaze direction of the child.

Although there are many accurate eye-gaze tracking devices available at the market, most of them have the wearable limitation such as the need of wearing a glass with camera. In DREAM, it is necessary to track the gaze direction of the child using non-contact visual sensors since the child won't feel comfortable wearing these devices. Compare to wearable devices, estimating the gaze direction by just using the non-wearable sensors is brings more challenges. The second significant challenge is the position of the vision sensors. Most existing non-wearable gaze estimation method chose to set the vision sensor under the screen or on the top of the screen to capture good quality of eye images. However, in DREAM, the robot will be in the middle of the table to interact with the child, which makes the gaze estimation much more challenge. The third significant challenge is the large head movement of the child, together with the position of the camera; this makes it impossible to capture all the faces using one sensor. Thus a multi sensor solution is necessary. This will further cause the challenge of multi sensor fusion and also real time performance.

With the development of image capturing devices, vision based gaze estimation has attracted more attention because of its non-intrusion and convenience. Various vision based gaze estimation methods have been proposed in the literature. Those methods can be classified into two categories, namely 2D based methods and 3D based methods.

2D based gaze estimation methods estimate the point of regard directly using the eye images. One of the most classic gaze estimation methods is Pupil Centre Corneal Reflection (PCCR) that makes use of the glint reflection on the eye and pupil centre. Then a regression method is used to map the vector of these two centre points to the point on the screen. This method has been adopted into many commercial eye estimation devices such as tobii. However this method requires an IR light source and the head of the user should be fixed during the experiment, which is not suitable for interacting child with ASD. Zhu et al. [1] improved the PCCR method by mapping the head movement to a reference position and then performing gaze estimation, thus allowed for a small range of head movement. However an IR source was still needed to estimate the gaze in his method. Valenti et al. [2] proposed to use a hybrid scheme to estimate gaze direction by combining the head pose and eye centre position. Lu et al. [3] proposed an Adaptive Linear Regression method to map the gaze point in the screen via sparsely collected training samples. Thus less training images were required during the calibration stage. Sugano et al. [4] estimated the point of regards when people were watching a video by combining the visual saliency of the video and the captured eye images. Williams et al. [5] proposed a sparse and semi-supervised Gaussian process regression model to map the eye images to screen coordinates. However most of the 2D-based methods need numerous of calibrations for training. If the training samples are not big enough, they can't handle large head movements. Moreover, they only estimate the gaze point in the screen rather than the direction in a 3D space.

Unlike 2D based gaze estimation methods, 3D based gaze estimation methods make use of the depth information of the eyes for gaze estimation. Lu et al. [6] acquired the depth information of the eye centre by using binocular vision that two cameras were placed on the top of the screen. Then the support vector regressor was used to map the space coordinates and local pattern model (LPM) to screen coordinates. Xiong et al. [7] proposed to estimate the

gaze direction through an RGB-D camera where the facial landmarks were tracked by using a supervised decent method. Funese et al. [8] proposed to create a face mesh by fitting a 3D Morphable Model to Kinect depth data and then cropped eye images to frontal looking to compensate head movement for gaze estimation.

The main challenges in gaze estimation include low resolution of the eye region images, real time performance requirement, large head pose, occlusions as well as variable illuminations. Thus the related algorithms should be able to deal with the low-resolution challenge, such as eye centre localization method and facial feature extraction method. Furthermore, all those algorithms should have low computation complexity to acquire real-time performance. Although the 3D based gaze estimation methods can allow free head movement compared to the 2D based gaze estimation methods, it remains a challenge to handle large head movement in child-robot interaction scenario. Thus a multi-sensor solution is necessary to deal with the large head pose in the scenario of interacting child with ASD.

5.2. Multiple Sensor Fusion

Multi-sensor fusion refers to combination of sensory data derived from disparate sensors in order to achieve better resulting information than using individual sources. Advantages of using multi-sensor system include robustness to environment and noise, rich information and better performance [9]. Human activities and interactions are inherently multimodal, and they involve gait, posture, body movements, speak, eye contact, and gaze and facial expressions. Multisensor fusion is an ideal tool to perceive, monitor and analysis these associated behaviours in achieving robustness to environmental, facilitating natural human-computer interaction and exploiting complementary information across modalities [9] [10]. Multi-sensor fusion strategies can be divided into three main categories, which are the data fusion, feature fusion, decision fusion [11].

The data fusion in low level will combine several sources of raw data into a new raw data, and synchronization and adaptation are usually needed before the fusion process. Statistical approaches are popular in the low level fusion, including non-recursive methods (such as weighted average method and the least square methods) and recursive methods (such as Kalman filter (KF) and extended KFs (EKFs)). For example, a complementary Kalman filter (CKF) [12] was proposed to overcome the drift problem in micro-sensor human motion capture. Matzka et al. [13] proved that covariance intersection and use of cross-covariance turned out to yield significantly lower errors than a Kalman filter at a comparable computational load. Covariance intersection also outperforms KF when the sensory data measured form multiple sensors were independent and it can generate a consistent estimate without considering the correlation between multisensory data [14]. Moreover, covariance union algorithm aims to solve the problem of information corruption which the covariance interaction cannot deal with [15].

The feature fusion methods in the intermediate level extract and concatenate features from several data sources to form a new feature which is more discriminating and higher dimensional than the source features [16]. The high dimensionality of the fused feature vector puts challenges to the classifiers, and dimension reduction techniques have been employed to solve this problem, such as principal component analysis (PCA), Independent component analysis (ICA), projection pursuit (PP), and linear discriminate analysis (LDA) [17, 18], [19,

20]. To model individual streams, classification methods such as Hidden Markov Models (HMMs) and their hierarchical counterparts, Support Vector Machine (SVM) and Dynamic Bayesian networks (DBNs), can be used [21-23]. The feature fusion methods are more popular than the data fusion and decision fusion because they can combine modalities with different weights and have access to the low level data and features [24]. Multistream Fused Hidden Markov Model (MFHMM) was employed to detect four cognitive states and seven prototypical emotions with a better performance than face-only HMM, pitch-only HMM, energy-only HMM, and independent HMM fusion [25]. Liu et al. [26] presented a sensor fusion method for assessing physical activity of human subjects, based on SVMs and demonstrated that the proposed multisensory fusion technique was more effective in identifying activity type and energy expenditure than the traditional accelerometer-alone-based methods.

Decision fusion methods consider and combine probability scores or likelihood values achieved from separate unimodal classifiers to develop a final decision. The quality of final decision is up to the way to estimate the best weighting factors based on the training datasets. For example, a self-optimizing approach was proposed to automatically select features, such as linear-prediction coefficients, and classifiers, such as Gaussian Mixture Models and SVM, and then combine the classifiers' results based on Dempster-Shafer theory to generate the final decision [27]. Stergiou et al. [28] presented an audio-visual person identification system with a audio-only recognition system, a video-only recognition system and an audio-visual fusion subsystem. The final fusion was done by combining the unimodal identities into the multimodal one, using a suitable confidence metric for the results of the unimodal classifiers. Zouba et al. [29] proposed a new multisensor based activity recognition approach which used video cameras and environmental sensors in order to recognize interesting elderly activities at home. The fusion was done at the decision level by combining video events with environmental events.

In addition, the above mentioned fusion strategies can also be combined in order to improve the performance [30, 31]. In [32], a multilevel multisensory fusion method was proposed to measure an exact recharging current, by combining fusion methods and rule-based methods to decide an exact output for maximum and minimum condition of the recharging current and generate high-reliability results. Heracleous [33] combined the feature fusion, multistream HMM fusion and late fusion methods to analyse noisy audio speech with Electro-Magnetic Articulography speech and obtained a satisfactory recognition result in a noisy environment.

Our objectives are to use date fusion methods in the low level to combine the raw data streams from different sensors in order to obtain a global 3D map of real-world environments containing information for the objects, body gestures, head pose, face expressions, etc., in the Deliverable 4.1, and to employ feature and decision fusion methods, such as hidden Markov models as well as conditional random fields and their variants to analyse the spatio-temporal gaze pattern and behaviour patterns of the children with ASD, in the Deliverable 4.2.

5.3. Camera Pose Estimation

Given n (n≥3) 2D-3D correspondence of 2D points in camera image and 3D points in space, calculating the pose of the camera which captures the image is known as the perspective-n-

point (PnP) problem [34]. It is widely used in practical applications such as SLAM, SFM, Augmented Reality, etc. Recently, a lot of researches have been explored in this research area. Among these implantations, the minimal number of 2D-3D correspondences is 3, which makes the PnP problems being P3P [35, 36]. To improve the accuracy and robustness, researchers tend to solve the camera pose estimation problems using redundant data, which employs more than 3 pairs of 2D-3D correspondence for the calculation. The PnP problems can be divided into two groups, which are the Multi-State Methods (MSM) and the Direct Minimization Methods (DMM). They all have their own advantages and disadvantages.

Normally, MSM only estimates the coordinates of partial points or camera projection matrix with partial points at earlier stages, and refines the results in later stages. Although the computing speed is fast when n is small, the accuracy is low. On the contrary, the accuracy is high when n is large with low computing speed due to the computational complexity [37]. DMM considers all the pairs of 2D-3D correspondence at one time. With proper energy function, the minimization process is carried on to find the best projection matrix of the camera [38]. The calculation process could be quick because of the employment of the linear or direct minimization algorithms [39]. However, due to the minimization process, the methods need a good initialization to avoid local minima[40].

5.4. Human Action Analysis

Human actions consist of the interactions between human and human, human and objects, adverse variations to motion pattern consistency, subtle articulated movements and the spatio-temporal activities. Depth sensors provide an innovative way of dealing such problems, and have been extensively applied in related academic research. Despite the intra-class variations, 3D depth data captured by the depth sensors for constructing a delicate model of human motions are more efficiently. Action recognition routinely includes two main tasks, which are feature extraction and dynamic pattern modelling. More discernible spatio-temporal information can be captured through exploiting the depth sensors.

A combination of spatio-temporal interest points localization like STIP [41] and low-level features like HOF [42] or HOG [43] is commonly adopted in the field of video-based motion recognition. When trying to utilize depth data, aforementioned local features are not optimally attributed to the situations that are with no texture in the depth map. Utilization of 3D articulated joint positions contributes to more precise motion recognition, which can be achieved by Multi-camera motion capture (MoCap) systems [44]. However, such system is of expensive equipment and requires markers, which impedes the wide usage. A marker-free motion capturing system is desirable and remains an interest among the research based on regular image sensors. As a result, low-cost depth cameras have been adopted for motion capturing with a trade-off in motion data quality.

Despite the susceptibility to occlusion, which may bring noise to the data, the results captured by the depth sensors are reasonable. To remedy the differences of motion data quality between the MoCap systems and the depth sensors, a specific recognition method should be developed.

There are various temporal models for human action recognition. Typically, two main models are employed: one is generative model such as Hidden Markov Model (HMM) and Conditional Random Field (CRF); another is dynamic temporal warping (DTW). For example,

Lv [45] employed HMM to model pre-defined relative positions that were obtained from the 3D joints, while Wu [46] utilized CRF over 3D joint positions. In [47], Xu also used HMM/CRF to model human actions in videos. Generally, the 3D joint positions obtained from depth maps are noisier than those from the MoCap data. Without careful feature selections, it is difficult to obtain the accurate states when the difference between actions is slight. This difficulty usually undermines the performance of such generative models. DTW [48] defines the distance of two time series as the edit distance, which can be incorporated with the nearest-neighbour classification method to achieve action recognition. However, the performance of DTW greatly relies on a good metric measurement of the frame similarity. Moreover, it may suffer from large temporal misalignment for periodic actions such as "waving" and consequently degrade the classification performance [49].

Recently, many works have been done for action recognition in depth data and skeletons. In [50], along the human silhouette, HMM was used for dynamics analysis and each depth frame was represented as a bag of 3D points. In [51], an efficient random sampling approach was presented to learn semi-local features from data. In [52], a dimension-reduced skeleton feature was described. In [53], spatio-temporal occupancy patterns were used, but all the cells in the grid should be in same size, and the number of cells was set empirically. In [54], the features based on the distances between each pairs of joints was represented, and multiple instance learning method was used for feature selection. In [55], the histogram of gradient was used as feature representation over depth motion maps. In [56], linear dynamic systems were employed to model the dynamic medial axis structures of human parts, and discriminative metrics were developed to compare the sets of linear dynamics systems for action recognition. In [57], a Kinect was used for dance action recognition. However, this system organized skeleton joints into human parts manually rather than automatically learned from data.

5.5. Face Expression Analysis

Automatic face analysis, which includes, e.g., face detection, face recognition, and facial expression recognition, has become a hot topic in computer vision because of its various applications in psychology, medicine, security, and computer technology. A face recognition system plays an important role in biometrics [58], and automatic facial expression recognition forms the essence of human behaviour understanding [59].

Face recognition aims to automatically identify and verify a person from a digital image or video sequence. Facial expression recognition considers two main streams: facial affect (emotion) detection and facial muscle action (action unit) detection [60]. The emotions conveyed by facial expressions are modelled with six categories: happiness, sadness, surprise, fear, angry and disgust. Ekman, etc., who propose these, argued that these emotions were universally displayed and recognized. Facial action descriptors rely on Facial Action Coding System (FACS). FACS defines 32 atomic facial muscle actions, called Action Units (AUs), which encode nearly any possible facial expressions.

Depending on temporal relations, facial representations can be categorized into spatial and spatio-temporal. Spatial representations encode image sequence frame-by-frame, whereas spatio-temporal representations consider temporal dependency in image sequence. According to facial feature descriptors in space, another classification is achieved: appearance-based approaches and geometric-based approaches. Appearance representations use textural

information, while geometric-based approaches ignore texture and describe shape explicitly. It is generally believed that appearance-based approaches outperform geometric-based approaches in both face recognition and facial expression recognition, and for expression analysis, spatio-temporal models outperform their spatial counterparts [59].

The main challenges in face analysis include occlusions, illumination changes and large variations in appearances and head-pose. Spontaneous behaviours often occur with occlusions and head-pose variations. Appearance-based approaches have the problems of face aging and identification bias for face recognition [58] and facial expression analysis, respectively [60, 62]. The problem of face analysis includes three main steps: (1) face registration, (2) facial feature extraction and representation, (3) feature analysis and recognition.

Face registration aims to find faces in an image. A successful face registration should localize or detect faces regardless of clutter, occlusions and head-pose variations. A rigid registration aligns the input face to a prototypical frontal face. Sometimes one has to select constant illumination or perform illumination correction, and normalize the faces to a fixed size [61].

An effective facial representation is a crucial for a successful recognition. Local Binary Patterns (LBP) [62] and Local Phase Quantisation (LPQ) [63] are two popular appearance-based methods in face analysis [64, 65]. Other commonly used textural representations include Histogram of Gradient (HOG), Gabor, sparse coding etc. It is popular to extract features from Three Orthogonal Planes (TOP) to extend spatial appearance representations to their spatio-temporal domain [66]. LBP-TOP and LPQ-TOP have been successfully used for facial expression recognition [67, 68]. Geometric-based approaches are seldom used for face recognition, but they form an important part in facial expression analysis [69]. The most frequently used shape representation is facial points representation, which describes a face as a number of fiducial points [69]. These methods have many advantages in head-pose wise facial expression recognition [70] and sptio-temporal models [71].

With derived facial representation, feature selection is sometimes needed when the number of patterns within a feature descriptor is too large. Principle Component Analysis is the most commonly used method for this task. For recognition, some popular classification methods, such as Support Vector Machine (SVM), Linear Discriminant Analysis (LDA) and AdaBoost are usually employed [72].

5.6. Object Tracking

Object tracking is to spatially locate objects in each image frame and temporally associate the located objects between consecutive frames in a video or image sequence. When the initial states of interested objects are known, the aim of object tracking is to accurately and efficiently locate and identify objects with times whatever their positions and sizes change. Object tracking is one of the most active topics in the field of computer vision and pattern recognition [73]. Nowadays, extensive tracking technology has been explored especially for advanced cameras, which promotes the development of automatic video analysis. Typically, there are three steps for realizing this process which are detecting objects of interest, tracking objects and analysing the tracked objects to recognize their behaviours [74]. In addition,

object tracking can also be used in many areas such as motion analysis, automatic surveillance as well as human-computer interaction etc. [75].

Recently, numerous representation schemes [76-78] have been proposed for robust object tracking. However, some major challenges, which might cause huge effect during tracking such as appearance variation caused by rotation, scale illumination, occlusion, to name a few. To address these problems, different methods can handle some of these issues but cannot solve all. For example, when processing occlusion aspect, PCA subspace representation [79] with online update was more effective compared with some other representations. Furthermore, an improved method using online updating and learning technology[80] was proposed to handle huge changes with time. However, a drift might occur [81] when using direct template updating. To remedy this, a Multiple Instance Learning (MIL) can be employed [74]. Moreover, sparse representation of trivial templates [81], which used to represent object, was proposed to process image noise. However, the real-time performance of this kind of method was unsatisfied.

To solve the above mentioned problems, an effective way is to adapt online updating to learn a low-dimensional subspace [82]. Moreover, extracting key features and online boosting method [83] can be combined for improving the computational speed. Alternatively, low-dimensional space can represent high-dimensional features by using a constructed classifier. If these features could be projected to arbitrary low-dimensional space, a necessary condition would be that the feature space should be sufficiently high according to the compressive sensing (CS) theory [84, 85]. This is because that to reconstruct the original high-dimensional features, enough information should be provided. During this progress, some important information and condition such as data-independent and information preserving should be guaranteed. For the background information could provide more specific information, it is also possible to use temporally information and local context information [86].

5.7. Speech Recognition

Speech recognition has wide applications, which makes it a hot topic of research. Typically, speech recognition consists of two basic operations, namely feature extraction and classification.

The feature extraction technique plays an important role for the speech recognition in a high accuracy. There are many feature extraction methods that are used in speech recognition, such as Fast Fourier Transforms (FFT), Linear Predictive Coding (LPC), Mel Frequency Cepstral Coefficients (MFCC) and Discrete Wavelet Transforms (DWT). The FFT has widely practical applications in computer vision such as signal processing, image analytics, speech recognition, etc. It is considered to be one of the most useful mathematical tools in computer science. For example, the FFT was used as feature extractor for speech recognition in [87]. Polur *et al.* [88] also carried out experiments for dysarthria speech recognition using FFT. The LPC is widely used to analyse the voice files with better result for low bit rate coding. For instance, in [89], LPC can obtain important features from the input signals. In [90] and [91], LPC was also successfully applied on speech processing. MFCC is used to extract unique features from human voice. For example, the MFCC was used in [92] for spoken letter recognition. The DWT is used to extract features from non-stationary signals such as audio. For instance, in [93], the feature extracted by DWT was classified with Support Vector

Machine (SVM) for speech recognitions. Sunny *et al.* [94] proposed a method for isolated spoken words recognition using DWT combining with Artificial Neural Networks.

There are many classification methods used in speech recognition. The Hidden Markov Model (HMM) is the popular one for speech recognition. In [95], the recognition using HMM to gain encouraging results with a high accuracy. Support Vector Machine (SVM) is another widely used and effective algorithm as a classifier for speech recognition. SVM is a binary classifier that divides the inputs into two different groups. It was successfully employed in [93] for speech recognition. Shady et al. [96] also presented a speaker independent arabic speech recognition using SVM. Dynamic Time Warping (DTW) is a method to find optimal registration between two sets of time-dependent data. It is considered to be the most suitable algorithm for speech recognition due to its capability of coping with different speaking speeds. In [97], the DTW was successfully used for speech recognition with a high accuracy.

6. Multi-camera System Design and Installation

This section includes system design, configuration and installation.

6.1. Multi-camera System Design

The design of the sensorized Therapy room mainly includes the intervention table, multiple sensors, fixing accessories and a workstation.

6.1.1. Intervention Table

The design of DREAM intervention table is provided in Figure 1. The detailed CAD file is provided as an appendix attachment. The table is specifically designed for DREAM project, accommodating ASD children interacting with humanoid robots. It provides a platform for a humanoid robot to perform, to mount multiple sensors, and to support a curtain to minimize the distraction to the children by covering data capturing sensors.

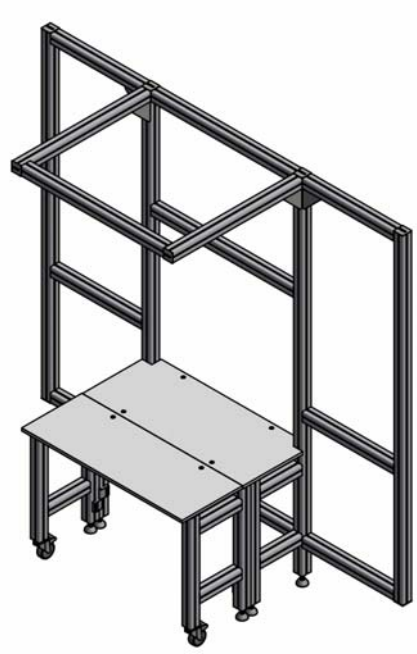


Figure 6.1: DREAM Intervention Table

6.1.2. Sensors

Five sensors are used and placed on the table including **two** RGB-D sensors and **three** ordinary cameras.

1. RGB-D Sensor:

   Kinect sensors are used to track human body gesture, robot position, object position, sound, voice, etc. More details about the "Kinect for windows" purchase and specifications:

   http://www.ebuyer.com/343455-microsoft-Kinect-for-windows-l6m-00002

2. Camera:

Cameras are used to capture human face, gaze, etc. More details about camera purchase and specifications:

http://scorpionvision.co.uk/CatalogueRetrieve.aspx?ProductID=5134875&A=SearchResult&SearchID=1557753&ObjectID=5134875&ObjectType=27

3. Lens:

http://scorpionvision.co.uk/CatalogueRetrieve.aspx?ProductID=6723233&A=SearchResult&SearchID=1557755&ObjectID=6723233&ObjectType=27

4. USB2 cable:

http://scorpionvision.co.uk/catalogue-index/cables/usb-cables/2-metre-usb2-a-male-to-mini-b-female

6.1.3. Accessories

5. Mounting kit for Kinect

1) A mounting kit is needed to fix a Kinect onto the bench, as shown on the right in the figure 2, which can be purchased at: http://www.ebay.co.uk/itm/221231532116?_trksid=p2059210.m2749.l2649&ssPageName=STRK%3AMEBIDX%3AIT (Only the top part will be used.)
2) In addition, one screw set is need for each Kinect. Screw (M6×16)×2 and nut (M6 Square nut Z) × 2 for Kinect : http://www.minitec.de/en//Web/produkte/Components/profile_system/system_components/Fastening-Elements/stat_schrauben_muttern.php

Figure 6.2: Mounting kit for Kinect. The left figure shows the mounting kit and the right shows the screw set to fix Kinect onto the table.

6. Mounting kit for cameras

1) Each camera is planned to be mounted on the table by TWO “Guard unit fixing angle 45 AL supplied loose”, shown in figure 3. http://www.minitec.de/en//Web/produkte/Components/profile_system/system_components/Fastening-Elements/Guard-unit-fixing-angle-45-AL.php?SESSID=fa15h3lnllspqaha30lf7fmga4

In addition, one screw set is need for each camera. Screw (M6×12)×3 http://www.minitec.de/en//Web/produkte/Components/profile_system/system_components/Fastening-Elements/stat_schrauben_muttern.php

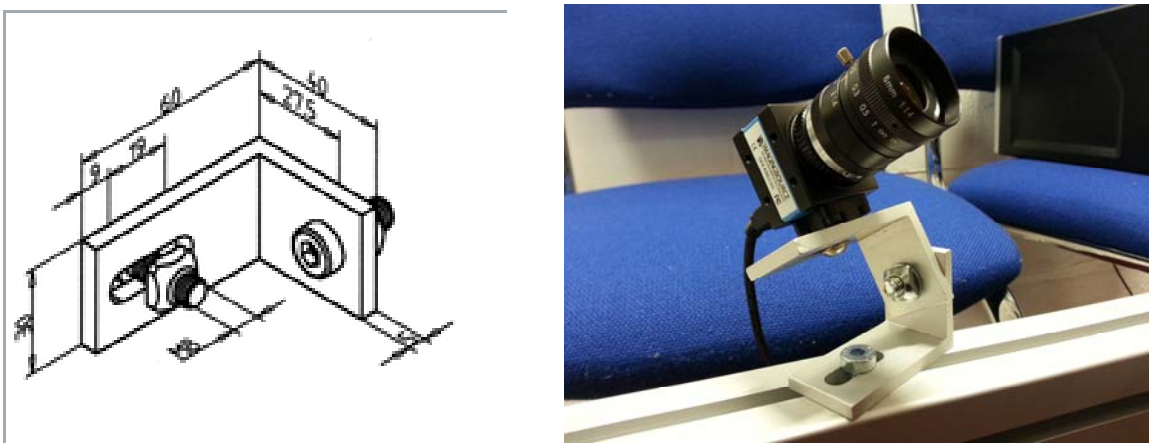


Figure 6.3: Fixing angle for camera

6.1.4. The Workstation

The minimum requirement for the workstation:

1. Intel Xeon CPU E5-1650 v2 3.5GHz or a faster processor
2. 5 USB Host Controllers (Three usb3.0 and two usb2.0)
3. 8 GB RAM
4. Graphics card supporting DirectX 9.0c

The workstation we recommend is:

http://store.hp.com/UKStore/Merch/Product.aspx?id=ECC_BUNDLE_4586044&opt=&sel=WKS#merch-marketing

Need to upgrade it from 8G memory to 16G memory, and 256G SSD to 256G SSD + 1T HDD. This workstation requires two extra usb3.0 express card adapters.

Here is one recommendation for the card adapter:

http://www.amazon.co.uk/Transcend-Express-Interface-Dual-Expansion/dp/B003MVJG8Q

In order to run the software smoothly, each sensor must be connected to a separate usb controller: each camera (DFK 42BUC03) must be connected to a separate USB3.0 controller; each Kinect sensor needs to connect to a separated USB2.0 controller (it will be better if there are more USB3.0 controllers for Kinects).

The reason why one USB controller can only be connected to a single sensor is that our image has a resolution of 1280*960 at 25fps. This means it needs to transfer a huge amount of data per second. After testing the sensors with the workstation, it found that connecting two cameras with a controller resulted in low quality images. At the same time, we need connect a DFK camera to an independent USB3.0 controller, which is much faster than USB2.0. There will be an obvious noise showing on the capture images if the camera is connect to a separate usb2.0 controller, which may cause loss of data. So we recommend connecting each camera to a separate/independent usb3.0 controller. Based on the testing, Kinect has a good/satisfying signal when connecting to a separate usb2.0 controller.

6.1.5. Summary

A list to summarize the above devices is provided as below:

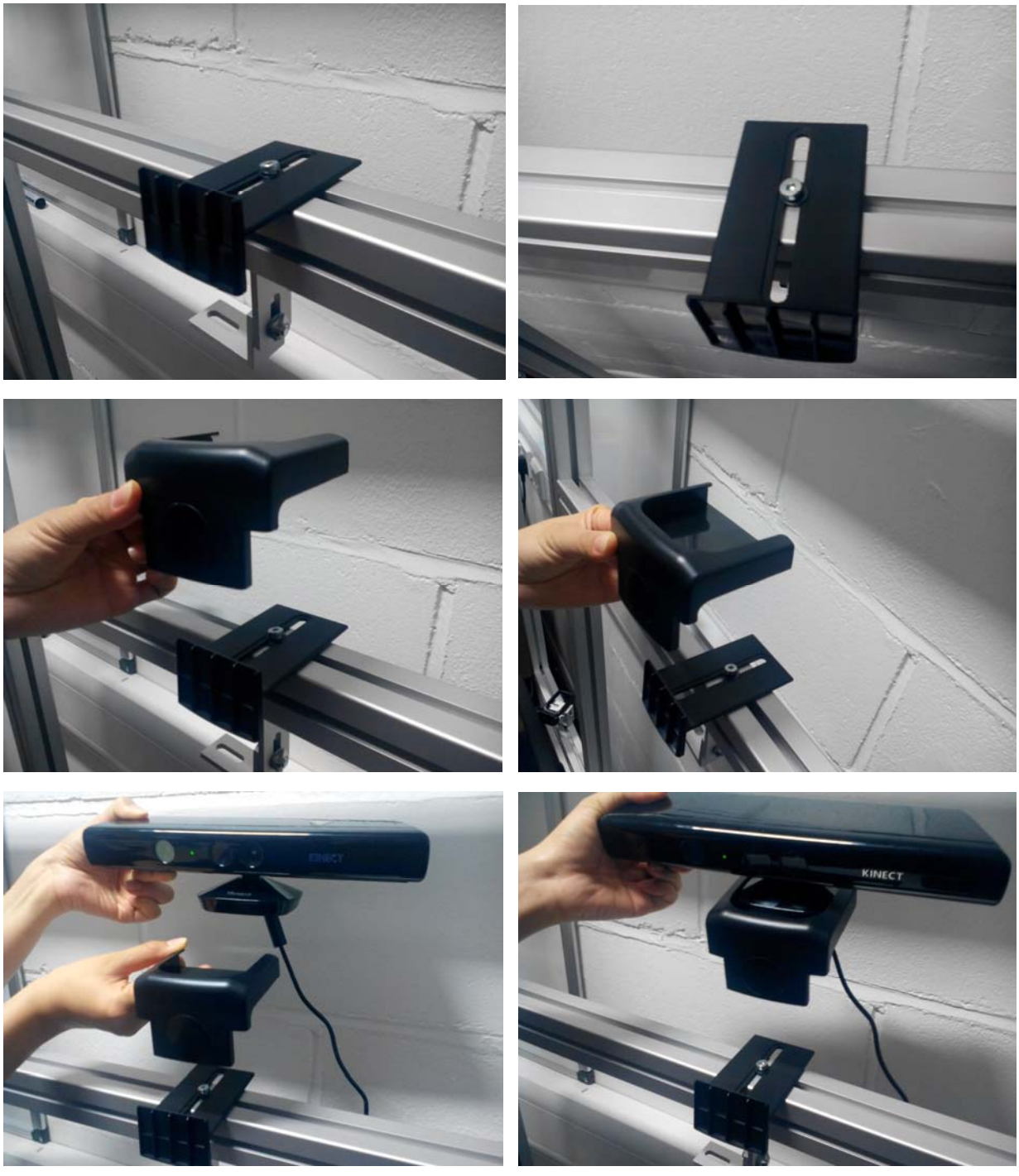
1. Intervention **Table** ×1: Specification, quotation and CAD drawing are attached.
2. **Kinect** ×2 (Kinect for **windows**): http://www.ebuyer.com/343455-microsoft-Kinect-for-windows-l6m-00002
3. **Camera**×3 (DFK 42BUC03 1.2mp Colour Camera with trigger and IO from Scorpion vision ltd): http://scorpionvision.co.uk/CatalogueRetrieve.aspx?ProductID=5134875&A=SearchResult&SearchID=1557753&ObjectID=5134875&ObjectType=27
4. **Lens** x 3 (HF6M-2 Spacecom 6mm C Mount Lens from Scorpion vision ltd): http://scorpionvision.co.uk/CatalogueRetrieve.aspx?ProductID=6723233&A=SearchResult&SearchID=1557755&ObjectID=6723233&ObjectType=27
5. Mounting kit for Kinect × 2: http://www.ebay.co.uk/itm/221231532116?_trksid=p2059210.m2749.l2649&ssPageName=STRK%3AMEBIDX%3AIT
6. Screw (M6×16)×2 and nut (M6 Square nut Z) × 2 for Kinect : http://www.minitec.de/en//Web/produkte/Components/profile_system/system_components/Fastening-Elements/stat_schrauben_muttern.php
7. Mounting kit for Camera, "Guard unit fixing angle 45 AL supplied loose" × 6, each camera needs two: http://www.minitec.de/en//Web/produkte/Components/profile_system/system_components/Fastening-Elements/Guard-unit-fixing-angle-45-AL.php?SESSID=fa15h3lnllspqaha30lf7fmga4 (*they have been **added** in the quotation of the table, see attached)*
8. Screw (M6×12)×3 for camera: http://www.minitec.de/en//Web/produkte/Components/profile_system/system_components/Fastening-Elements/stat_schrauben_muttern.php
9. Workstation ×1: http://store.hp.com/UKStore/Merch/Product.aspx?id=ECC_BUNDLE_4586044&opt=&sel=WKS#merch-marketing
   (Need to upgrade it from 8G memory to 16G memory, and 256G SSD to 256G SSD + 1T HDD + two usb3.0 express card adapters).
   http://www.amazon.co.uk/Transcend-Express-Interface-Dual-Expansion/dp/B003MVJG8Q

6.2. System Installation

6.2.1. Hardware Installation

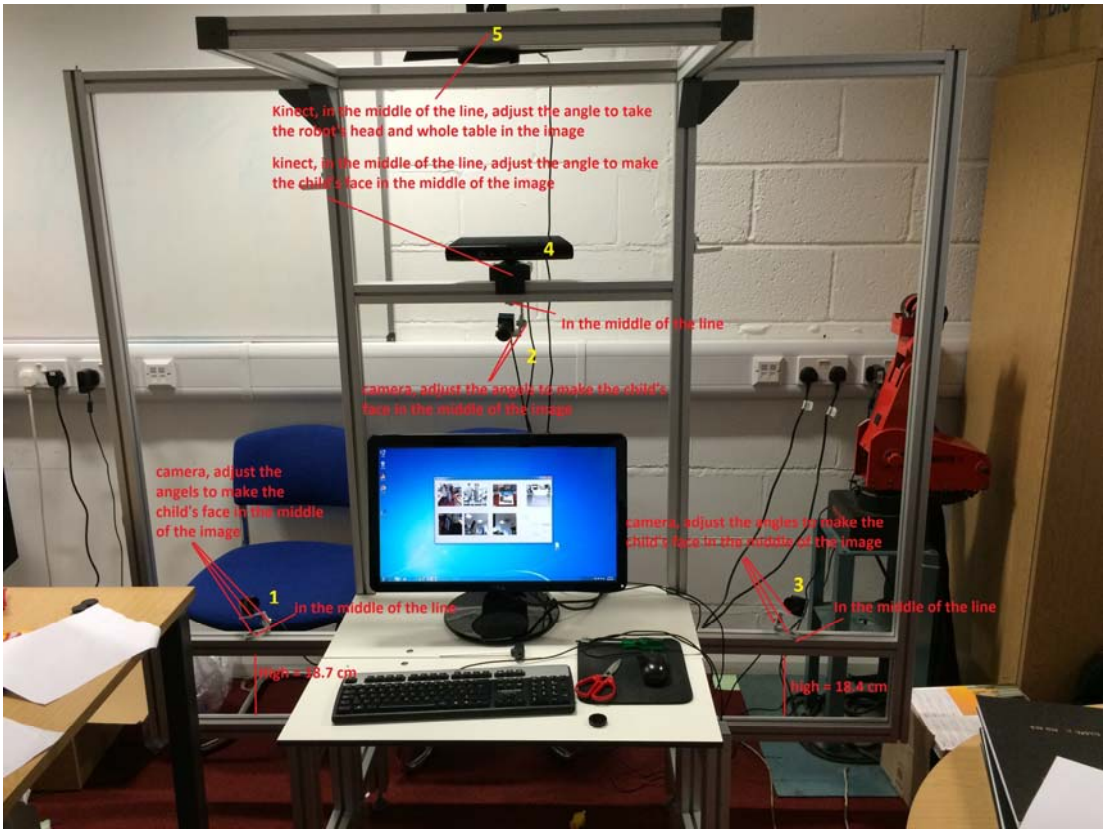
6.2.1.1. Kinect Installation

Both Kinects are installed at the middle of the bars. A screw nut (M6 square nut Z) should be pre-inserted in the bar in advance. The procedure is shown in the following figures.



6.2.1.2. Camera installation

Three cameras are fixed at the places shown in the following figure.



The following figures show the steps of installing the camera on the angles:

6.2.2. Software installation and instruction

6.2.2.1. Software installation

The software package requires to run on a Windows 7 (64bit) operating system.

1. Make sure the computer has 5 separate usb controllers, which include at least 3 USB3.0 controllers for the three cameras (one for each) as mentioned in the workstation. After inserting two usb3.0 express card adapters, click the \Dreamsetup\USB3.0 adapter\RENESAS-USB3-Host-Driver\ setup.exe to install the driver for the adapter.

2. Plug the DFK 42BUC03 camera in an independent USB3.0 controller. The recommended workstation has three usb controllers. Each usb controller has a number of usb ports. As shown in the following figure, there are three usb ports in the front of the computer. The black usb ports are of usb2.0 and they belong to usb2.0 **control1**. The two blue usb ports belong to usb3.0 **control2**.

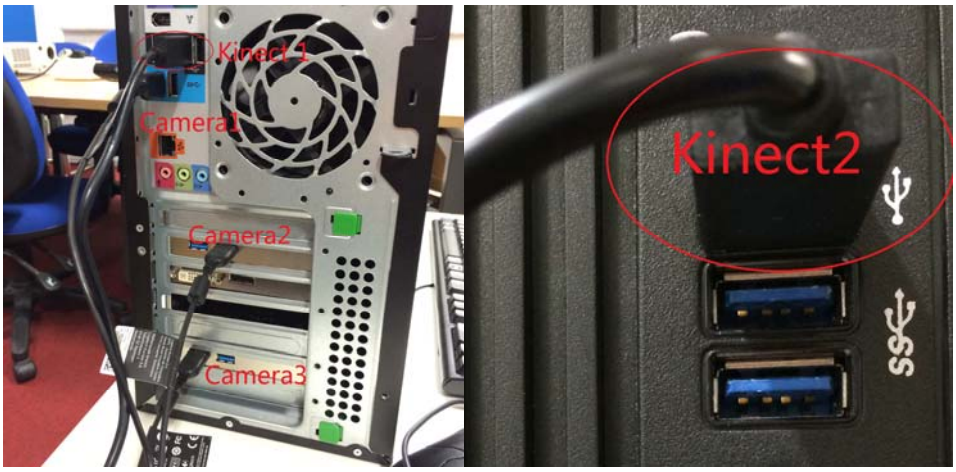
There are 6 usb ports at the back of the case. Four black ports belong to usb2.0 **control3**; two blue ones share a same usb3.0 controller, which is usb3.0 **control2**.

Plug one camera in one of the blue usb ports; insert the other two cameras into two PCIE to usb3.0 adapters respectively.

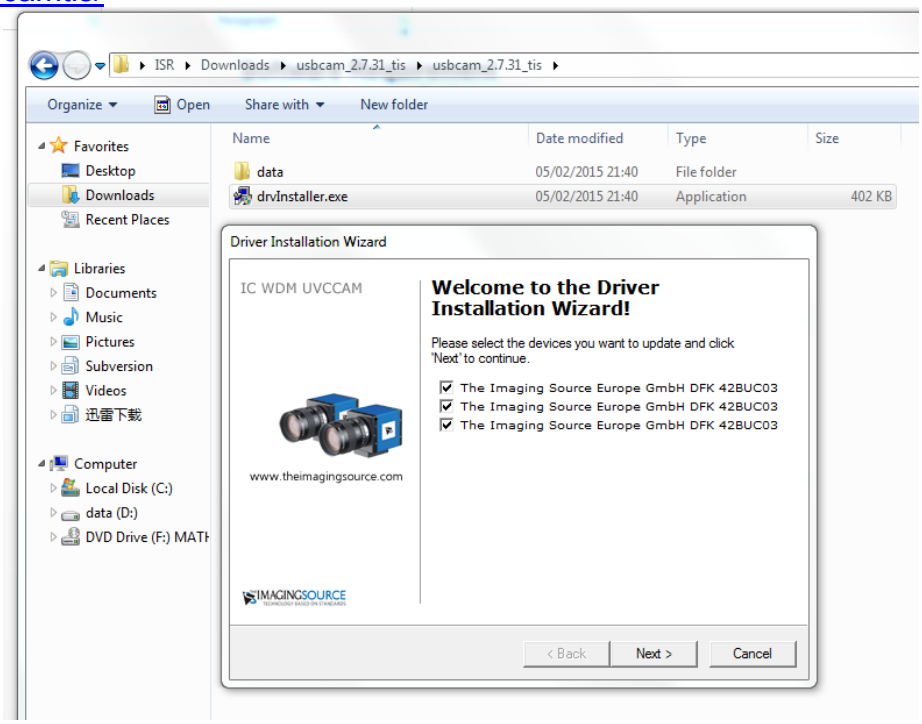
The mapping are summarised as:

Kinect1	→	Control1
Kinect2	→	Control3
Camera1	→	Control2
Camera2	→	PCIE card adapter 1
Camera3	→	PCIE card adapter 1

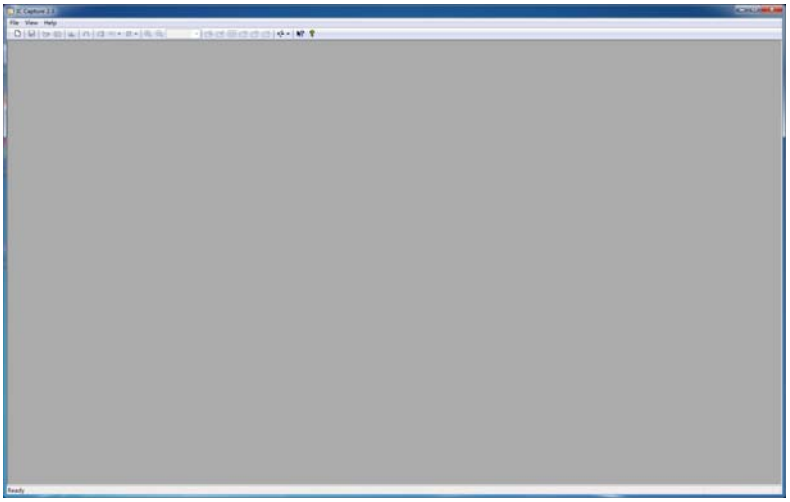
There links are shown in the following figures.



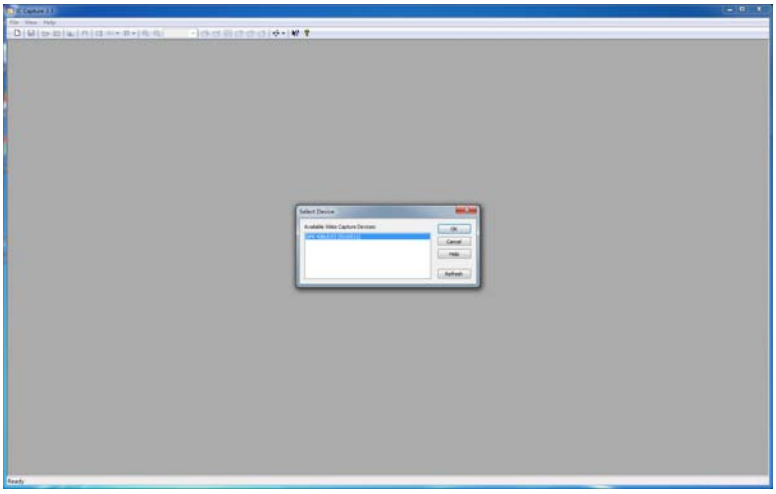
3. To install the DFK 42BUC03 camera driver, double click the “drvInstaller.exe” in the folder of /Dreamsetup/camera_driver. The driver can also be downloaded from the link:
   http://www.theimagingsource.com/en_US/support/downloads/details/icwdmuvccamtis/



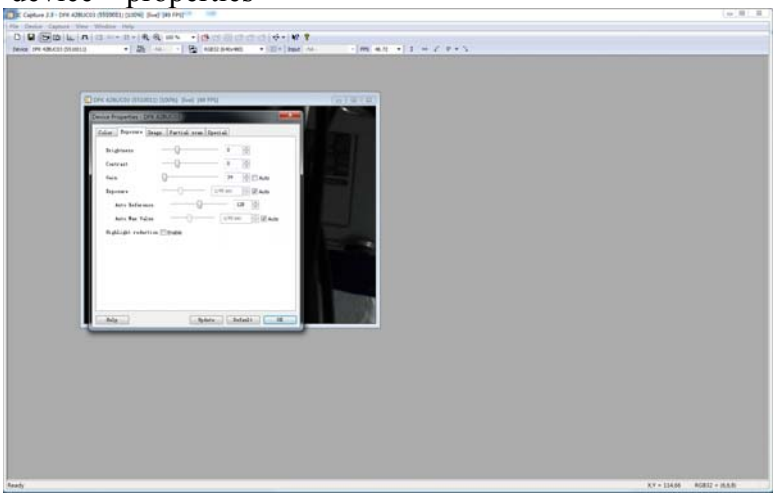
4. Set the environment for the cameras. Cancel the gain option of the camera so that the image will get rid of the noise.
   Double click the ‘IC Capture.exe’ in the folder of /Dreamsetup/IC Capture 2.3.
   Click file --- new



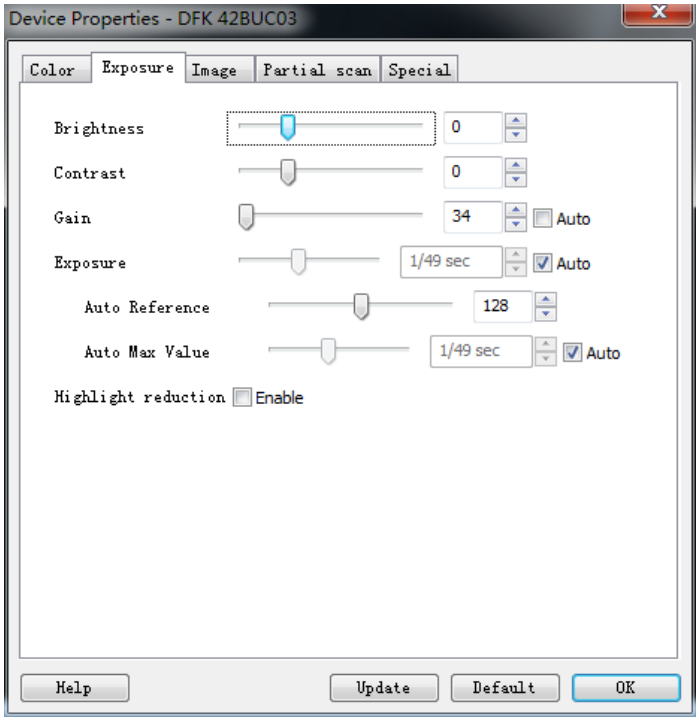
Select the camera and click OK



Click---device---properties

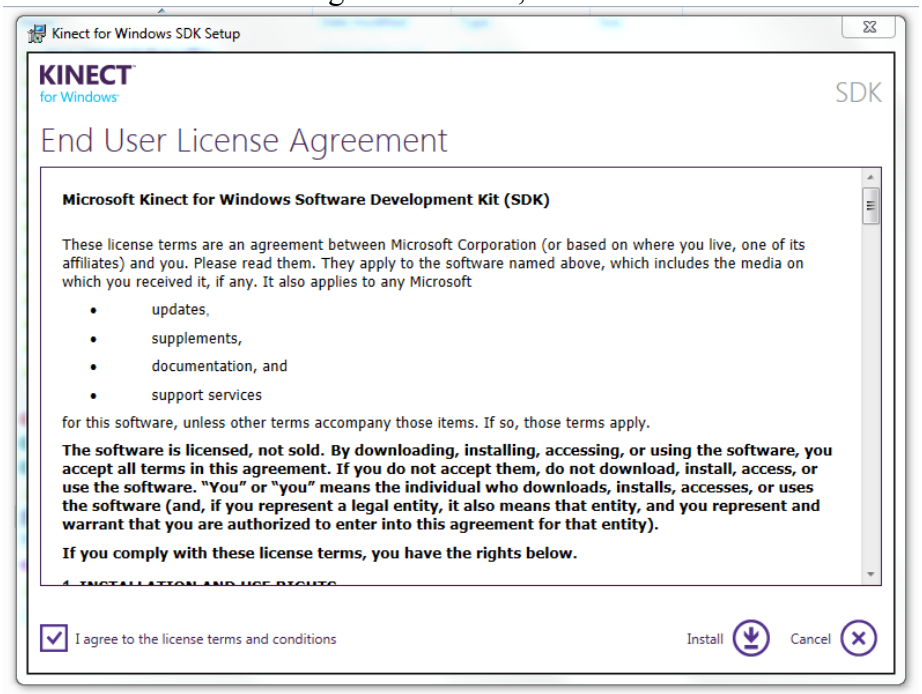


Click exposure and cancel the tick of Gain and set it to the smallest as the following image. Then click OK.



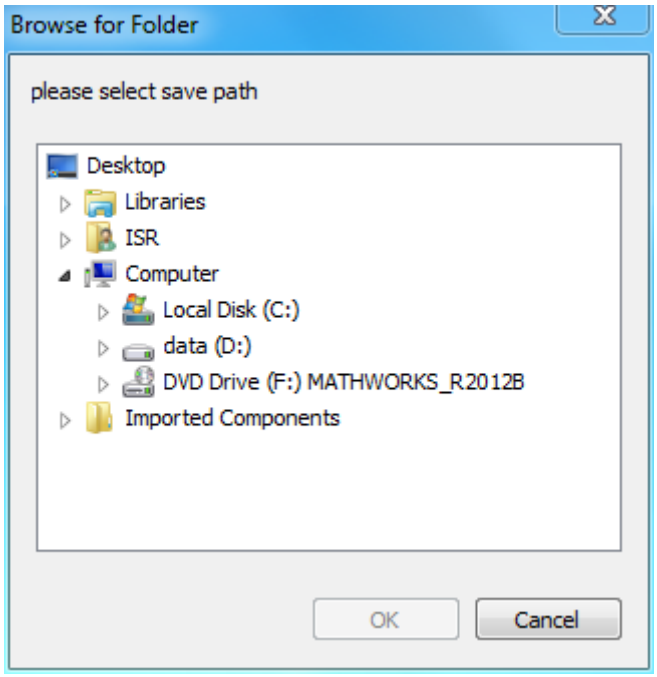
Repeat this step three times for all the three cameras and make sure that all the camera's parameters are set.

5. Install Windows SDK 1.8 for Kinect, double click the KinectSDK-v1.8-Setup.exe and then tick on "I agree to the …", then click the install.

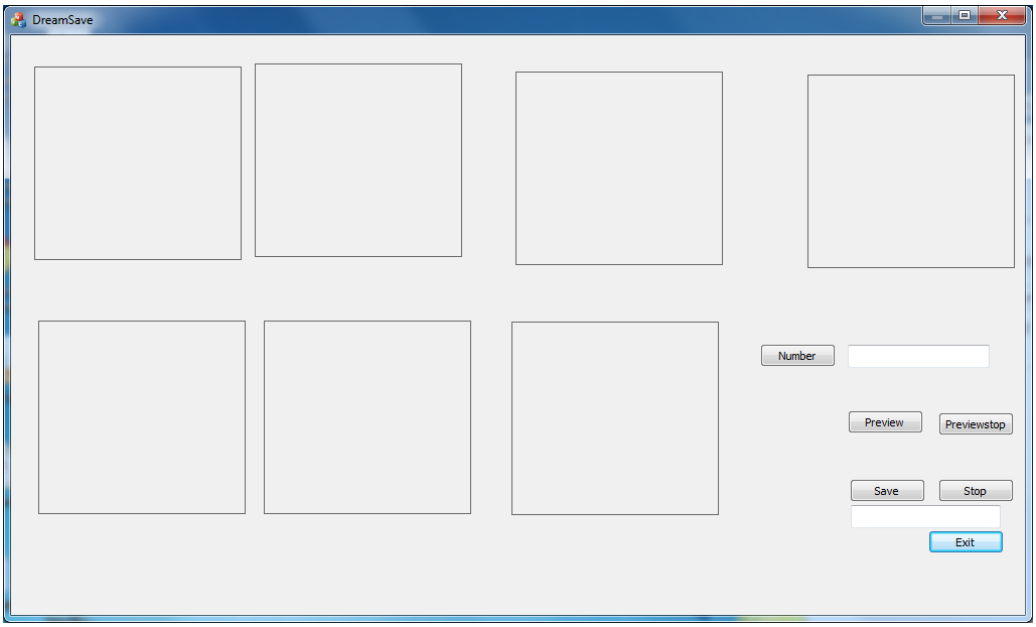


6.2.2.2. User instruction

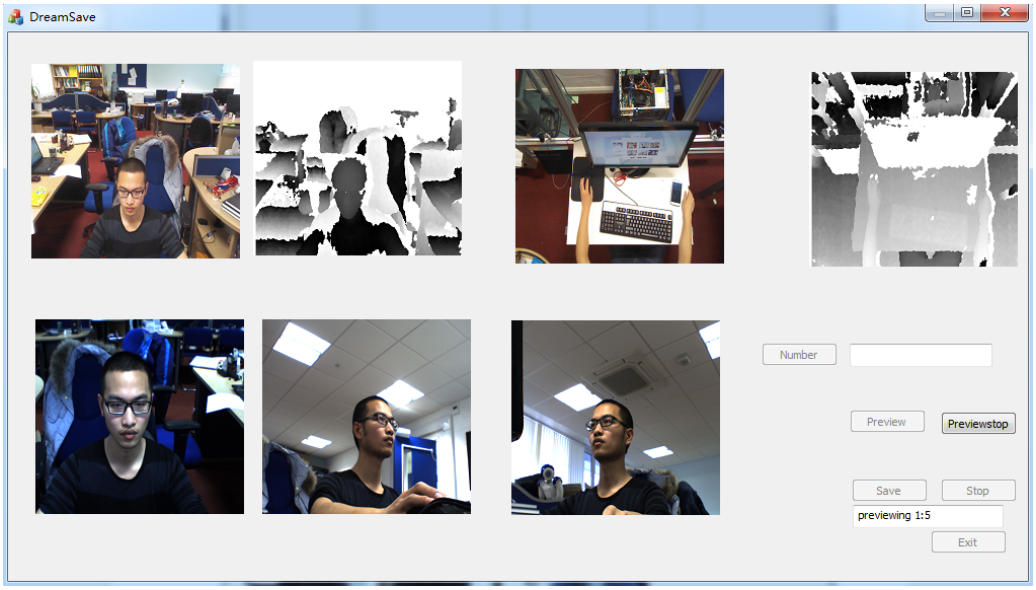
1) Right click the \Dreamsetup\Release\Dreamsave.exe and select run as administrator.
2) Then a browse for Folder to place the captured data:



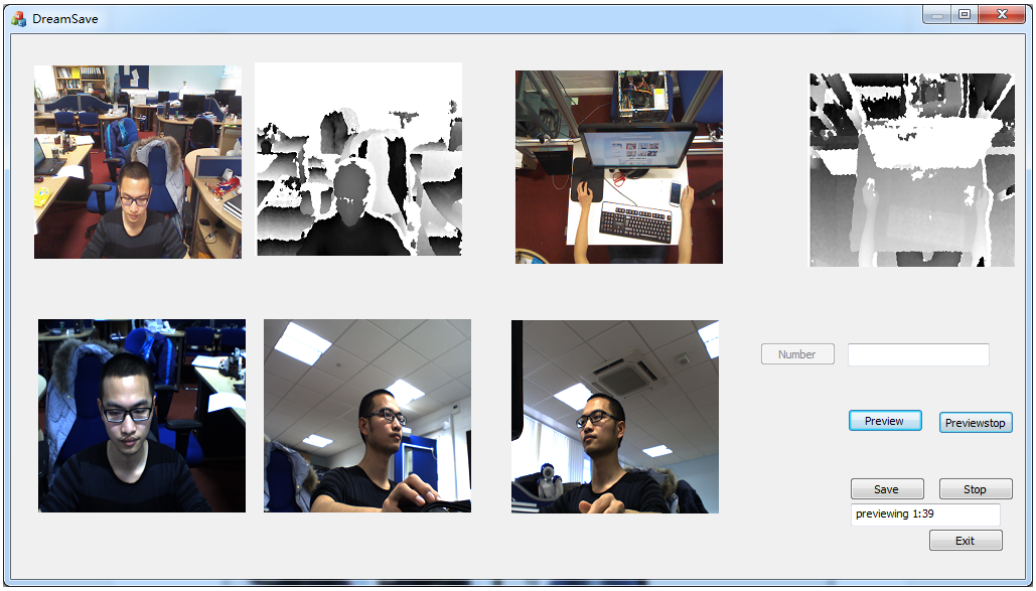
3) Select a folder that has suffient space to save data. After the path is selected, one following window will prompt out:



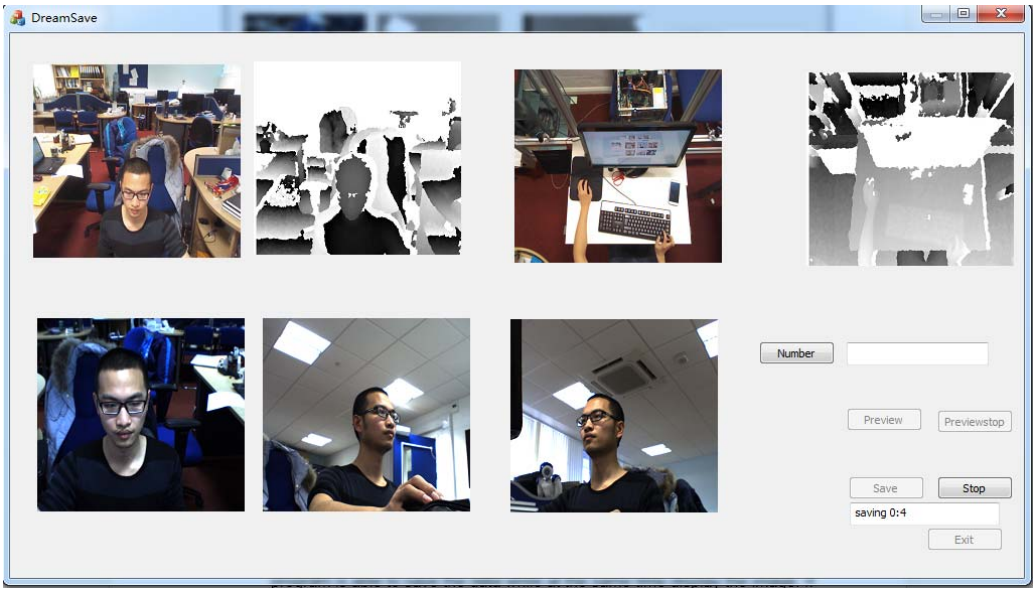
4) We can click the preview button to check whether the sensors are working well. Here is one preview example:



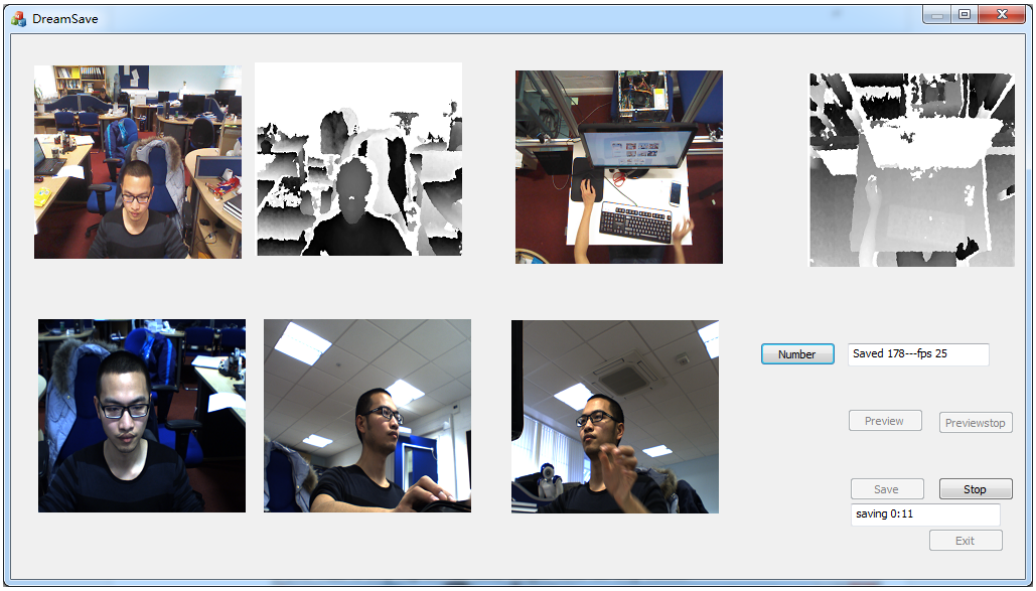
The previewing time is displayed at the right-bottom of the window. As we can see that 51 seconds have passed. We can simply click the previewstop button to stop preview.



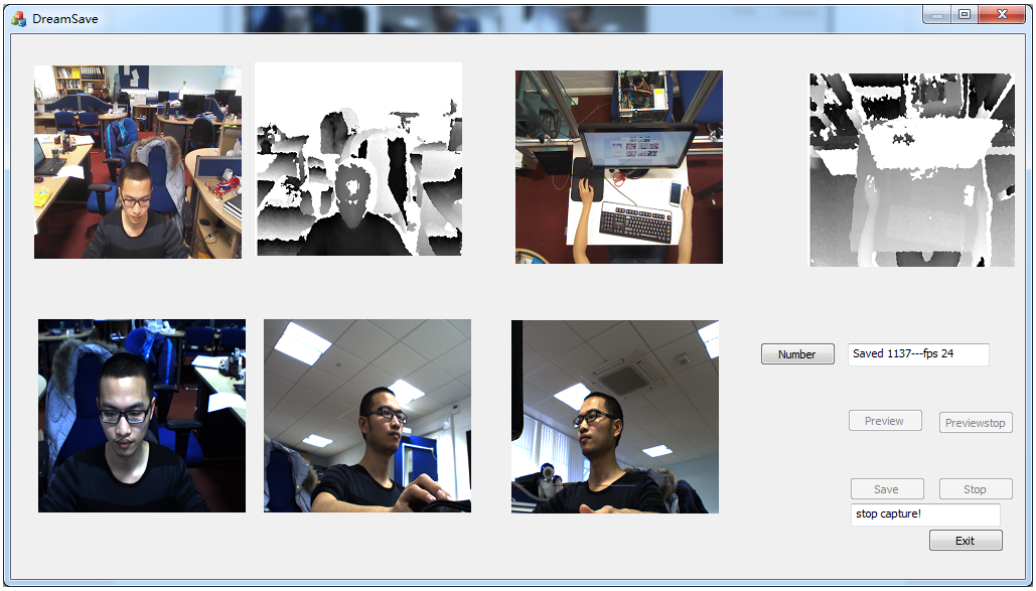
In order to save the video, we can simply click the Save button. Then the following image shows up.



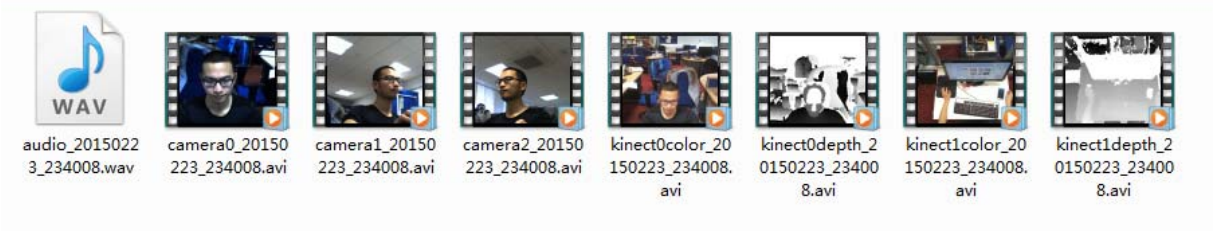
The saving time is also displayed at the right-bottom of the windows. Our program is able to save the data while at the same time displaying the image. If you want to check how many images have been saved, just click the number button. The number of images captured will display on the right of the button (click the button will not affect the performance of the program, you can click as many times as you want).



If you want to stop capturing the data, simply click the stop button.



Then click the exit button or red button to exit the program. The data will be saved at the path you selected.



Note: After preview, if we click the Exit button or the red button on the right-top of the window, the program will exit.

6.2.3. Sensor adjustment

6.2.3.1. Kinect adjustment

Once the Kinects are mounted on the bar, you may need to adjust the view angle of the sensor manually. You should ensure that the view range covers the NAO robot and the whole table plane. For the middle Kinect, you should ensure that the face is at the centre of the image when the child faces toward the camera.

6.2.3.2. Camera adjustment and testing

Once all the cameras are mounted on the bar, you may need to adjust the view angle of the sensor manually. You should ensure that the face is at the centre of the image when the child faces toward the camera.

Here is one example image from the testing of the sensors, all of which are working well:

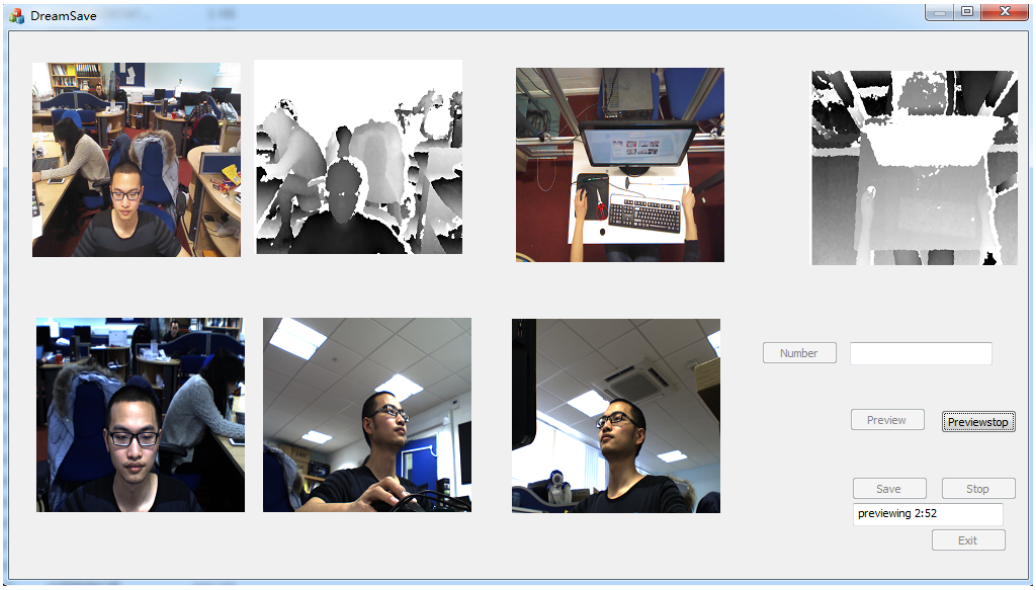


Figure 6.1

7. Method Development and Implementation

7.1. Camera Pose Estimation

7.1.1. Method

In this section, we need to determine the position and orientation of each camera, given its intrinsic parameters and a set of n correspondences between 3D points and their 2D projections. The 3D point coordinates are from the Kinect. The 2D projections on camera plane are obtained by matching 2D points between camera image and Kinect RGB image. Our

implementation is based on Efficient Perspective-n-Points (EPnP) algorithm proposed by Vincent, Francesc and Pascal, with the 2D-3D correspondence obtained mentioned above, the camera poses can be estimated accurately. The PnP algorithm employed in our implementation is a transformation-based PnP method. As shown in the following equation,

$$\mathbf{m}_i \approx \mathbf{K}(\mathbf{R},\mathbf{t})\mathbf{M}_i$$

m_i is the projection of the 3D point Mi onto the camera image with K being intrinsic parameters of the camera. R is the rotation matrix, t is the translation matrix. mi, K and Mi are known in the equation. With more than 3 pairs of mi-Mi correspondence, the R and t can be estimated using optimization algorithms. In our implementation, the mi-Mi correspondences are more than 20 pairs to improve the robustness of the process.

7.1.2. Results

Our camera pose estimation method has robust results for different camera poses. It only requires the user to mark the corresponding points between the Kinect image and Camera image manually for about 20 pairs. This approach is preferred as it's far more reliable than any other feature-matching algorithm. With the intrinsic parameters of the cameras, the poses of those cameras related to the Kinect can be found reliably. And we also provide an interface to transform the rotation across different coordinate systems. The experiment results are shown in Fig. 1. As illustrated, the poses of three cameras at three different positions can be estimated accurately.

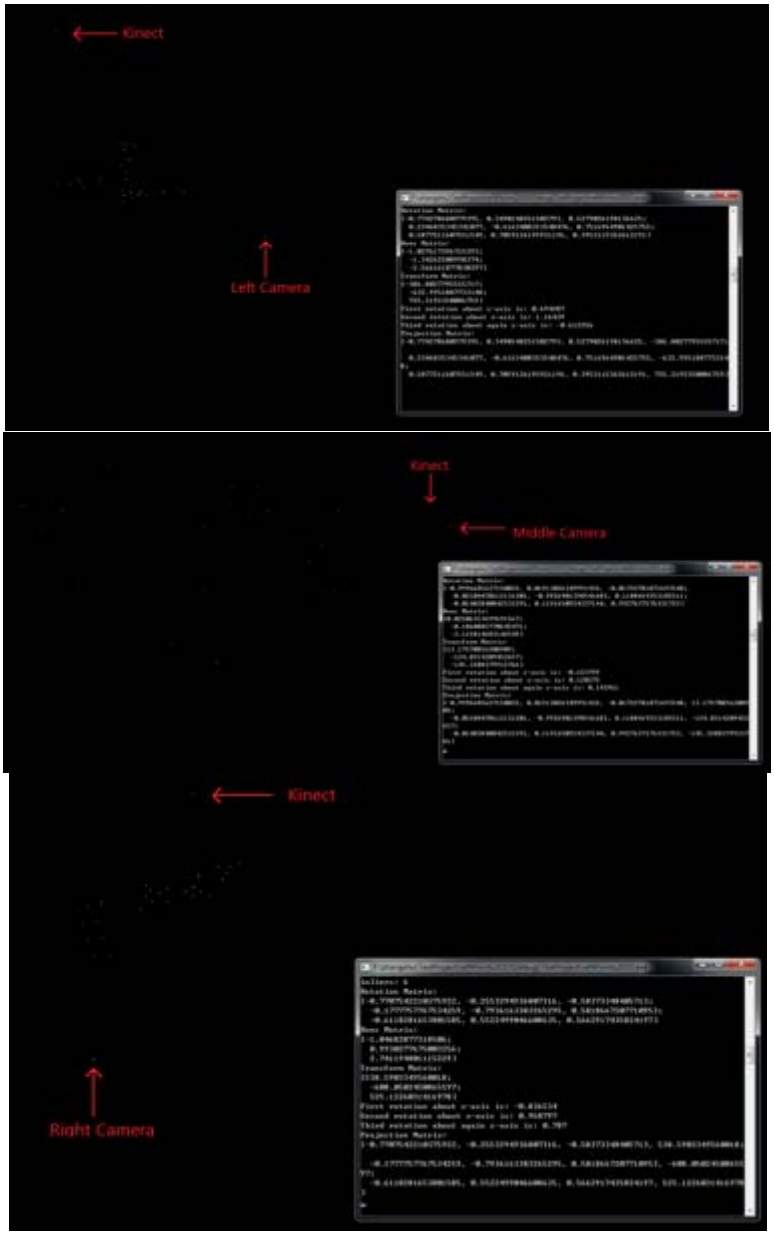


Figure 7.1

7.2. Gaze Estimation

7.2.1. Face Location Method

The boosted cascade face detector is employed with default parameters in order to obtain the approximate location of the face [98]. This method corresponds to the getFaces(x,y,z) function.

7.2.2. Eye Location Method

This correspond to the getEyes(eyeLx, eyeLy, eyeLz, eyeRx, eyeRy, eyeRz) function. In our project, we present an convolution based integro-differential eye center localization

method to localize the eye centers. The proposed method is computationally much cheaper than the original integro-differential method [99] and also achieves a higher accuracy in a public available lowe resolution image database.

The original integro-differential method is a very popular eye localization method in the literature which is defined as follows:

$$\max_{(r,x_0,y_0)} \left| G_\sigma(r) * \frac{\partial}{\partial r} \oint_{r,x_0,y_0} \frac{I(x,y)}{2\sigma r} ds \right| \qquad (1)$$

Where $G_\sigma(r)$ is a Gaussian smoothing function with a scale of σ. $I(x,y)$ represent for the eye image. ds is the contour of a circle with the center point of (x_0, y_0) and radius r . The convolution operation is denoted as $*$. The operator locate the eye centre by make use of the drastic intensity along the boundary of iris and cornea.

The following equation is the discrete implementation of the integro-differential operator:

$$\max_{(n\Delta r,x_0,y_0)} = \left| \frac{1}{\Delta r} \sum_k \left\{ \left(G_\sigma\big((n-k)\Delta r\big) - G_\sigma\big((n-k-1)\Delta r\big) \right) * \sum_m I[(k\Delta r \cos(m\Delta\theta) + x_0), (k\Delta r \sin(m\Delta\theta) + y_0)] \right\} \right| \qquad (2)$$

Where Δr and $\Delta\theta$ represent for small increments in radius and angular.

Instead of considering the small increments along the angular. We design two kinds of mask to convolute the eye image. The proposed method calculates a ratio derivative between neighbor curve of iris and cornea which is formulated as follows:

$$\begin{cases} I_r = K_r * I(x,y) \\ I'_{r+1} = K'_{r+1} * I(x,y) \\ D_r = \frac{I'_{r+1}}{I_r} \\ argmax_{(r,x,y)}(D_r) \\ r \epsilon [r_{min}, r_{max}] \end{cases} \qquad (2)$$

Where K_r and K'_{r+1} are two kind of designed masks. I_r and I'_{r+1} are the convolution result of the different eye image $I(x,y)$. And D_r is the ratio derivative. r_{min} and r_{max} represent for the minimum and maximum of the radius r. The computation complexity of the proposed eye localization method is greatly reduced by employing FFT in the realization of convolution

7.2.3. Gaze Estimation Method

We propose a real-time gaze estimation method by constructing multi-sensor fusion system to handle the large head movement. Three cameras and two Kinects are used in this system. In the gaze estimation task, the cameras are used to capture the face of the child. The frontal Kinect is used to capture the head position in world coordinate and the top Kinect is used. All the image data are captured simultaneously by creating 8 handles in programming. Each handle deals with difference kinds of data. The data captured in each handle are two Kinect RGB image data, two Kinect depth data, three camera data and one Kinect audio data. The resolution of the camera, Kinect RGB image, Kinect depth image are 1280*960, 640*480 and 640*480 separately.

To estimation the gaze direction. Firstly the facial features should be located. In pure method, we employ the method proposed by Xiong et al. [100] to locate the feature points in the human face. In order to deal with head movements, the head poses need to be determined.

We employ the object pose estimation method (POSIT) proposed by Dementhon et al. [101] is used to calculate the direction of head gaze (corresponding to the getHead(headx, heady, headz) function. Then the eye centre is located by applying the proposed convolution based intergo-differential eye centre localization method. It should be noted that the gaze direction differs from the head gaze by two angles, the horizontal direction θ and the vertical direction φ. The final gaze direction is finally determined by adding the angles to the head gaze. The following is the equation to calculate the gaze direction (corresponding to the getEyeGaze(eye, x, y, z) function).

$$\begin{cases} \theta = \tan^{-1}(\gamma * \sqrt{(x_p - x_c)^2 + (y_{p-}y_c)^2} * \frac{\cos\alpha}{L}) \\ \varphi = \tan^{-1}(\varepsilon * \sqrt{(x_p - x_c)^2 + (y_{p-}y_c)^2} * \frac{\cos\beta}{H}) \end{cases} \quad (4)$$

Where (x_c, y_c)means the the center of two eye corners, (x_p, y_p)means the center eye pupils, α is the angle between the line of two eye corners and the line of two centres. β is the complementary angle of α. L is the distance of the two eye corners, γ and ε are determined through experiment.

7.2.4. Results

The following is the result of gaze estimation. The green points are the located facial points. The located eye centres are marked in red. The red line which starts on the nose indicates the direction of the head pose. The white line which starts at the middle of two eye centres is the gaze direction.

Figure 7.2

7.3. Human Action Analysis

The aim of this section is mainly about recognising the behaviour of ASD children. To achieve this objective, this part will provide the requirements on action and event recognition based on multi-sensory data (task 4.4). The meaning of function and variables has been shown in data interpretation part which will adopt skeleton data. After acquiring skeleton information, the centre of body, hand joint and grip position will be obtained and output the 3D coordinates information. The body pose such as manipulate object and shaking hand can be estimated as well as arm angel through skeleton joint angel.

7.3.1. Method

To recognise the action of human body, this project will use the skeleton information acquired by Kinect, one main idea is to represent the movement of human body using the pairwise relative positions of the joints feature. For a human subject, 21 joint positions are tracked by the skeleton tracker and each joint i has 3 coordinates $p_i(t) = (x_i(t), y_i(t), z_i(t))$ at a frame t. The illustration of the skeleton joints are shown in Fig. 1. The coordinates are normalized so that the motion is invariant to the initial body orientation and the body size. For each joint i, we extract the pairwise relative position features by taking the difference between the position of joint i and any other joint j:

$$p_{ij} = p_i - p_j$$

The 3D joint feature for joint i is defined as:

$$p_i = \{p_{ij} | i \neq j\}$$

From the equations, enumerating all the joint pairs are used for introducing some information, this might be irrelevant compared with classification task, but our system can handle this difficulties by selecting most relevant joints for recognising task. Relative joint position is actually a quite intuitive way to represent human motions. Think about that, for example, the action "waving". It can be interpreted as "arms above the shoulder and moving left and right". This can be effectively characterized through the pairwise relative positions.

7.3.2. Results

Using the skeleton information of Kinect, some results are shown as Fig.6 to Fig.8, as well as the 3D joint position including stand model and seat model.

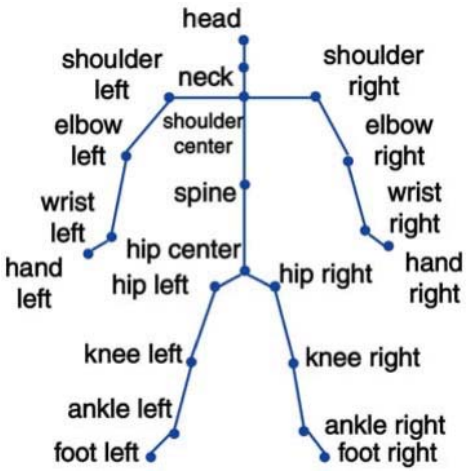


Figure 7.3 Human joints tracked with the skeleton tracker

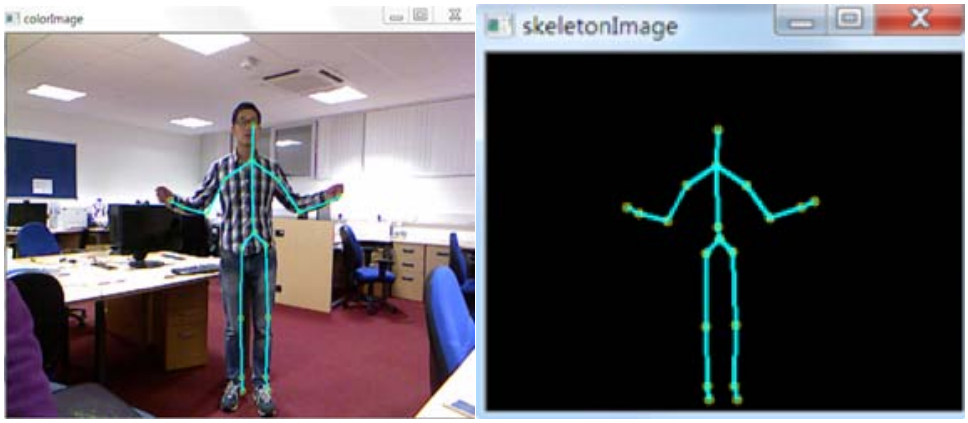


Figure 7.4 Action recognition using depth information and Kinect skeleton (stand model)

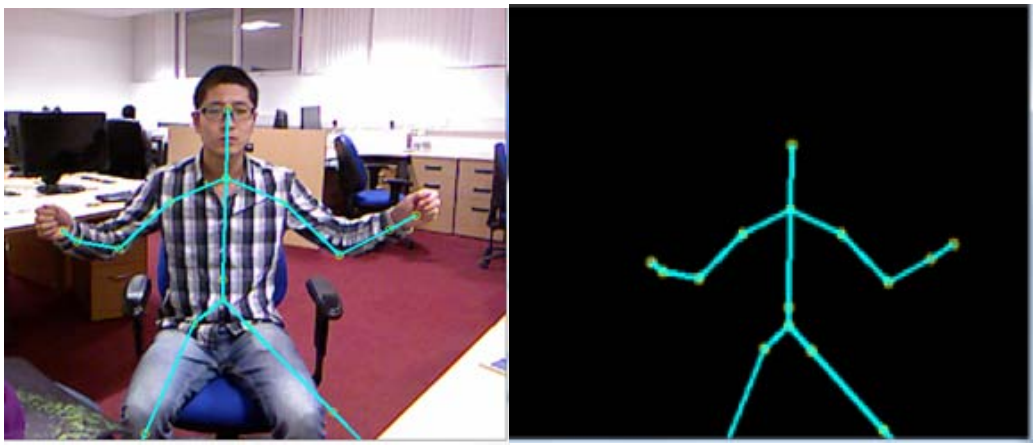


Figure 7.5 Action recognition using depth information and Kinect skeleton (seat model)

7.4. Face Expression Analysis

Face & facial expression recognition are the relevant components as mentioned in task 4.4. Task 4.4 provides some advice that the facial appearance cues should be captured and Support Vector Machine (SVM) is considered as classifier. So we use Local Binary Patterns (LBP) to represent facial texture cues and apply SVM for identity & facial expression classification.

7.4.1. Method

LBP is a nonparametric method and has been proved as a powerful descriptor in representing local textural structure [72]. The main advantages of LBP are its strong tolerance against illumination variations and computational simplicity. This method has been successfully used in both spatial and sptio-temporal domains in face recognition and facial expression recognition.

The original LBP operator labels the pixels of an image with decimal numbers. Each pixel is compared with its eight neighbors in a 3×3 neighbourhood, considering the center pixel value as threshold; bigger values are encoded with 1 and the others with 0. A binary number is obtained by concatenating all these values. Its corresponding decimal number is used to compute LBP histogram. Figure 9 shows an example of LBP operator.

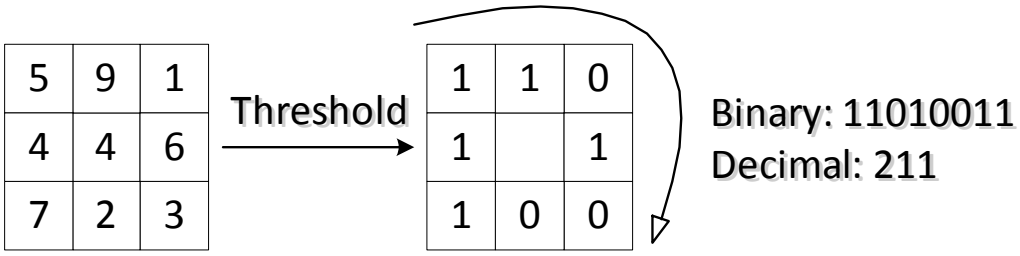


Figure 7.6 Example of LBP operator.

7.4.2. Support Vector Machine

SVM is considered as one of the most powerful machine learning techniques for data classification. It achieves a good balance between structural complexity and generalization error. It offers great performance under the circumstance of very few training samples, high dimensionality and nonlinear classification.

In a two-class learning task, SVM find a maximal margin hyperplane as its decision boundary. For a linear separable dataset, SVM assumes that the best classification results are obtained by maximizing the margin of hyperplane between two classes. It allows not only the best partition on the training data, but also leaves much room for the correct classification of the future data. In order to guarantee the maximum margin hyperplanes to be actually found, an SVM classifier attempts to maximize the following function with respect to $\vec{w}$ and b:

$$L_P = \frac{1}{2}\|\vec{w}\| - \sum_{i=1}^{t} \alpha_i y_i (\vec{w} \cdot \vec{x_i} + b) + \sum_{i=1}^{t} \alpha_i$$

where t is the number of training examples, α_i are the Lagrange multipliers. The vector $\vec{w}$ and constant b define the hyperplane.

SVM only makes binary decisions. For multi-class classification problem, one-vs-rest technique provides a computational simpler and flexible strategy, which trains binary classifiers to one class from all the others

7.4.3. Result

We evaluate our system on CK+ database using 10-fold cross-validation. CK+ is a set of image sequences in which the facial expressions of subjects are displayed from neutral to target emotions. For our experiment, the first neutral face and three peak frame are used, which results in 1236 images (135 Angry, 177 Disgust, 75 Fear, 207 Happy, 84 Sadness 249 Surprise and 309 Neutral). The confusion matrix is shown below.

Table 1 Facial expression recognition rate

	Neutral	Angry	Disgust	Fear	Happy	Sadness	Surprise
Neutral	50.15	15.32	9.61	3.60	9.61	8.11	3.60
Angry	26.15	36.15	9.23	4.62	1.00	9.23	4.62
Disgust	20.44	6.63	56.91	0.55	8.84	4.42	2.21

Fear	13.33	8.33	6.67	43.33	13.33	10.00	5.00
Happy	13.04	4.83	8.21	6.76	59.42	3.38	4.35
Sadness	31.17	7.79	10.39	12.99	9.09	24.68	3.90
Surprise	4.84	1.61	0.40	2.42	3.23	2.02	85.48

7.5. Object Tracking

The aim of this section is mainly focus on tracking part including tracking ASD child hand, the trajectory of hand, objects holding by ASD child and getting the distance between the object and table. The complication of tracking part can assist analysing the behaviour of ASD children which is part of task 4.4. The main task of this part if outputting 3D coordinates of object the position of children in real world and acquiring the trajectory of hand of ASD children. The key technology for achieving this aim is as follows.

7.5.1. Method

Despite that numerous algorithms [102] have been proposed in the literature, object tracking remains a challenging problem due to appearance change caused by pose, illumination, occlusion, and motion, among others. In this project, a compressive tracking algorithm will be used which can handle referred problems. The tracking problem is formulated as a detection task and the main steps of the proposed algorithm. We assume that the tracking window in the first frame is given by a detector or manual label. At each frame, we sample some positive samples near the current target location and negative samples away from the object center to update the classifier. To predict the object location in the next frame, we draw some samples around the current target location and determine the one with the maximal classification score.

To account for large scale change of object appearance, a multiscale image representation is often formed by convolving the input image with a Gaussian filter of different spatial variances. The Gaussian filters in practice have to be truncated which can be replaced by rectangle filters.

For each sample $Z \in R^{\omega \times h}$, its multiscale representation is constructed by convolving Z with s set of rectangle filters at multiple scales $\{F_{1,1}, \dots, F_{\omega,h}\}$ defined by

$$F_{\omega,h}(x,y) = \frac{1}{\omega h} \times f(x) = \begin{cases} 1, & 1 \le x \le 0, 1 \le y \le h, \\ 0, & otherwise. \end{cases}$$

Where ω and h are the width and height of a rectangle filter respectively.

Then we represent each filtered image as a column vector in $R^{\frac{\omega h}{}}$ and concatenate these vectors as a very high-dimensional multiscale image feature vector $x = (x_1, \dots, x_m)^T \in R^m$ where $m = (\omega h)^2$. The dimensionality m is a typically in the order of 10^6 to 10^{10}. We adopt a sparse random matrix **R** to project x onto a vector $v \in R^n$ in a low-dimensional space. The random matrix **R** needs to be computes only once offline and remains fixed throughout the tracking process. For the sparse matrix **R**, the computational load is very light. And we only need to store the nonzero entries in **R** and the positions of rectangle filters in an input image

corresponding to the nonzero entries in each row of **R**. Then, v can be efficiently computed by using **R** to sparsely measure the rectangular features which can be efficiently computed using the integral image method.

It is easy to show that the low-dimensional feature v is scale invariant. Each feature in v is a linear combination of some rectangle filters convolving the input image at different positions. Therefore, without loss of generality, we only need to show that the j-th rectangle feature x_j in the i-th feature v_i in v is scale invariant. We have

$$x_j(xy) = F_{sw_j,sh_j}(sy) \otimes Z(sy)$$
$$= F_{sw_j,sh_j}(a) \otimes Z(a)|_{a=sy}$$
$$= \frac{1}{s^2\omega_i h_i}\int_{u\epsilon\Omega_s} Z(a-u)du$$
$$= \frac{1}{s^2\omega_i h_i}\int_{u\epsilon\Omega} Z(y-u)|s^2 du$$
$$= \frac{1}{\omega_i h_i}\int_{u\epsilon\Omega} Z(y-u)du$$

$$=F_{w_j,sh_j}(y) \otimes Z(y) = x_j(y)$$

Where $\Omega = \{(u_1,u_2)|1 \le u_1 \le \omega_i, 1 \le u_2 \le h_i\}$ and $\Omega_s = \{(u_1,u_2)|1 \le u_1 \le s\omega_i, 1 \le u_2 \le sh_i\}$

We assume all elements in v are independently distributed and model them with a naive Bayes classifier

$$\mathrm{H(v)} = \log\left(\frac{\prod_{i=1}^{n} p(v_i|y=1)p(y=1)}{\prod_{i=1}^{n} p(v_i|y=0)p(y=0)}\right) = \sum_{i=1}^{n}\log\left(\frac{p(v_i|y=1)}{p(v_i|y=0)}\right)$$

Where we assume uniform prior, $p(y=1) = p(y=0)$, and $y\epsilon\{0,1\}$ is a binary variable which represents the sample label.

The random projections of high dimensional random vectors are almost always Gaussian. Thus the conditional distributions $p(v_i|y=1)$ and $p(v_i|y=0)$ in the classifier H(v) are assumed to be Gaussian distributed with four parameters $(\mu_i^1, \sigma_i^1, \mu_i^0, \sigma_i^0)$,

$$p(v_i|y=1) \sim N(\mu_i^1, \sigma_i^1), p(v_i|y=0) \sim N(\mu_i^0, \sigma_i^0)$$

Where $\mu_i^1(\mu_i^0)$ and $\sigma_i^1(\sigma_i^0)$ are mean and standard deviation of the positive (negative) class. The scalar parameters in last equation are incrementally updated by

$$\mu_i^1 \leftarrow \lambda\mu_i^1 + (1-\lambda)\mu^1$$

$$\sigma_i^1 \leftarrow \sqrt{\lambda(\sigma_i^1)^2 + +(1-\lambda)(\sigma^1)^2 + \lambda(1-\lambda)(\mu_i^1-\mu^1)^2}$$

Where $\lambda > 0$ is a learning parameter,

$$\sigma^1 = \sqrt{\frac{1}{n}\sum_{k=0|y=1}^{n-1}(v_i(k) - \mu^1)^2}$$

And $\mu_1 = \frac{1}{n}\sum_{k=0|y=1}^{n-1} v_i(k)$. Parameters μ_i^0 and σ_i^0 are updated with similar rules. The above equations can be easily derived by maximum likelihood estimation.

Because the variables are assumed to be independent in our classifier, the n-dimensional multivariate problem is reduced to the n univariate estimation problem. Thus, it requires fewer training samples to obtain accurate estimation than estimating the covariance matrix in the multivariate estimation. Furthermore, several densely sampled positive samples surrounding the current tracking result are used to update the distribution parameters, which is able to obtain robust estimation even when the tracking result has some drift. In addition, the useful information from the former accurate samples is also used to update the parameter distributions, thereby facilitating the proposed algorithm to be robust to misaligned samples. Thus, our classifier performs robustly even when the misaligned or the insufficient number of training samples is used.

7.5.2. Results and explanation of function

This part will mainly focus on functions F13, F14, F15, F19, F20 and F25. The meaning of these functions has been explained in data interpretation part. To complete these functions, some programming will be designed and output the 3D coordinates of object and hand. To show the result vividly, image result which is the preliminary result shown as follows which indicates that the object tracking can work well when therapist interact with ASD child.

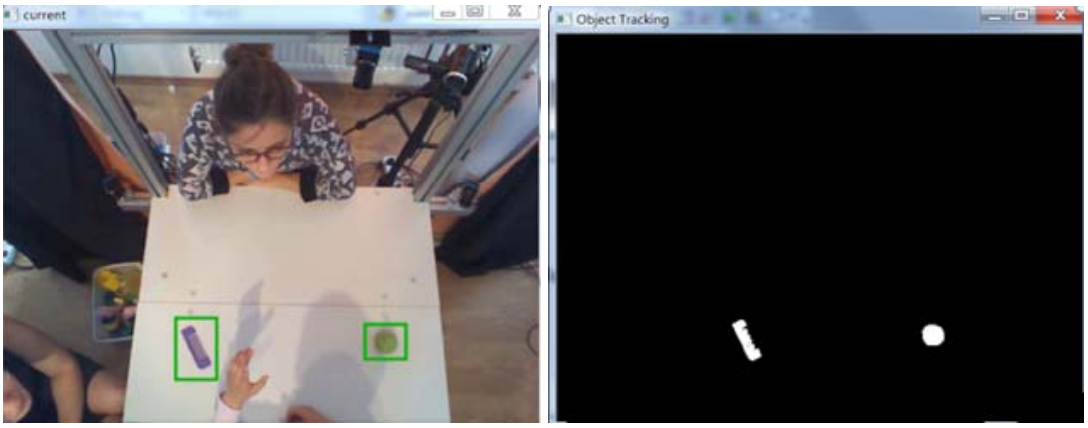


Figure 7.7 The result one of object tracking

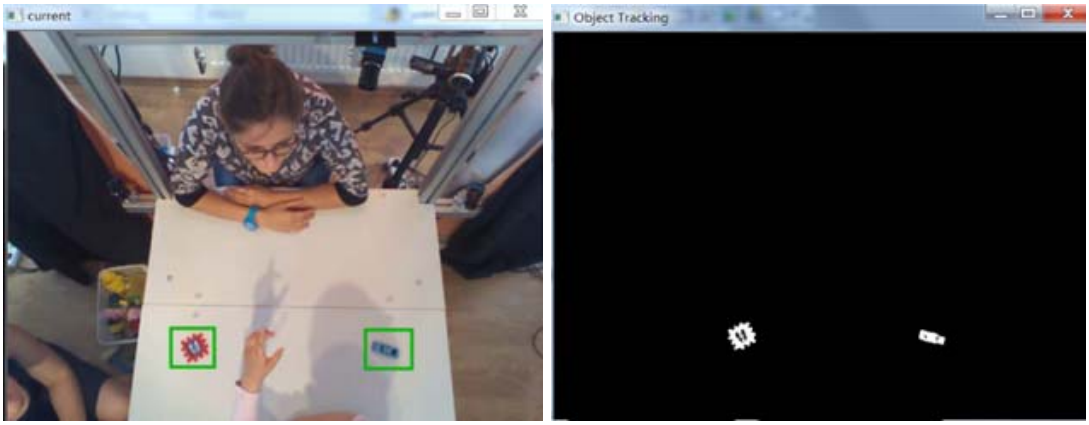


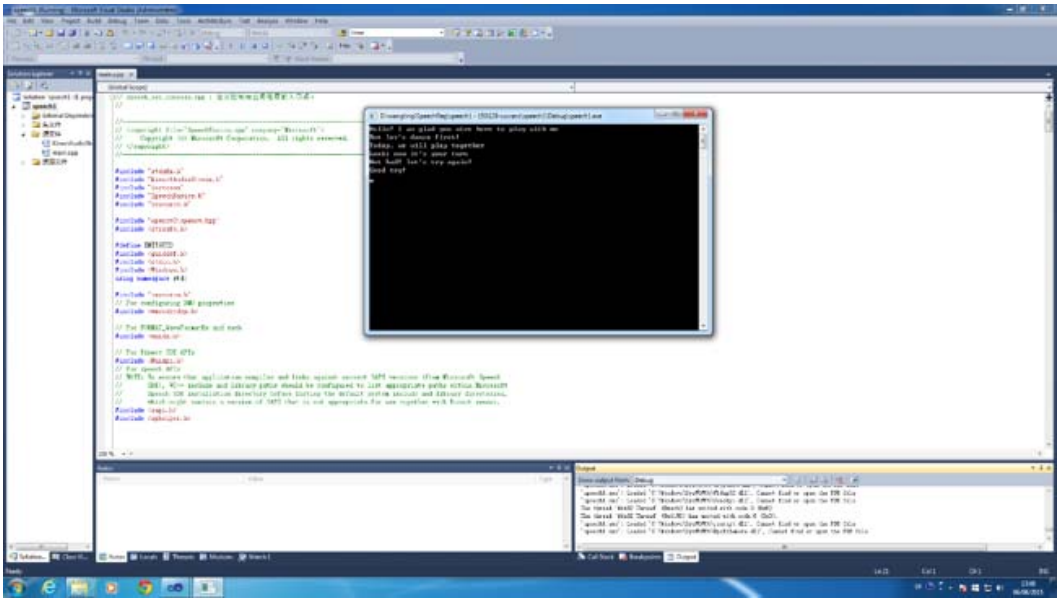
Figure 7.8 Result two of object tracking

7.6. Speech Recognition

In DREAM project, the effective child-robot social interactions in supervised autonomy RET requires the robot to be able to infer the psychological disposition of the child. The speech recognition can help the system to understand the psychological disposition of the child better. Therefore, the implementation of the speech recognition is given in the system. With the speech recognition functions, the specified words and sentence spoken by child can be transformed into plain texts for easier understanding what child have expressed. Our implementation is based on Microsoft Kinect SDK.

7.6.1. Results

The isolated words or continuous sentences can be recognized with our implementation of speech recognition that built on top of word recognition technology. Our speech recognition system displays the content of recognition results in plain text as output when the speaker is speaking. The follow figures are the experiment results.



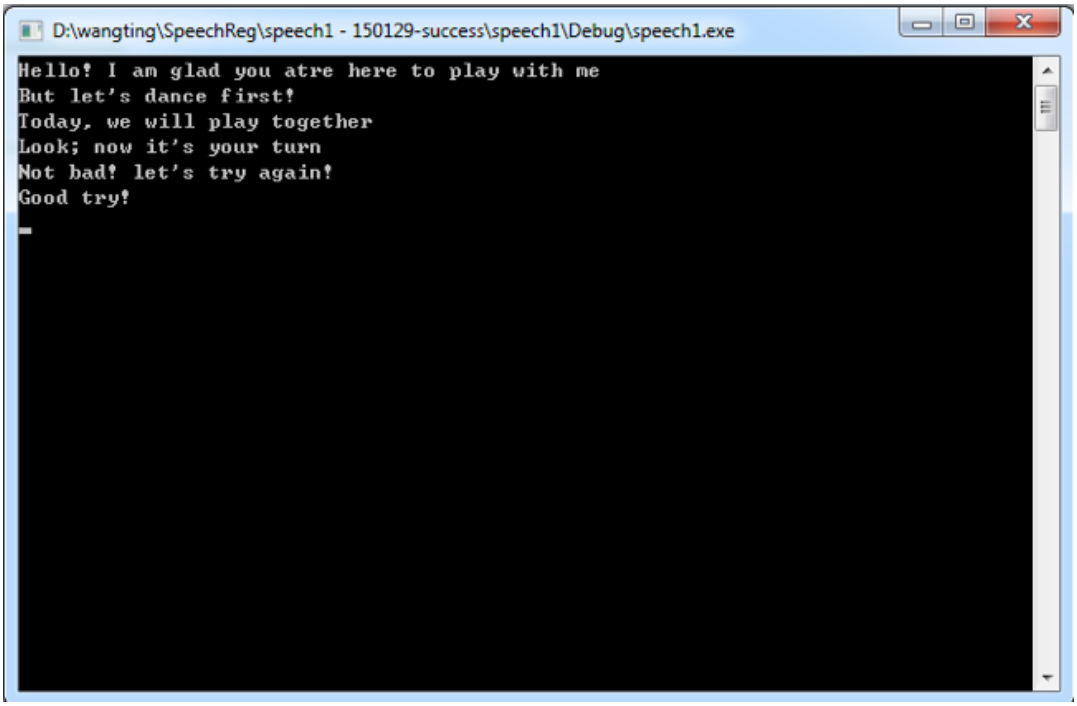


Figure 7.9

7.7. Multiple Sensors Capturing and Fusion

As described in the above section, the hardware configuration was introduced, which provides the platform for *Task 4.1: design multi-camera system for data acquisition of smart space* in software. This section describes the design of synchronising all the sensors in software, not only for multi-sensory data acquisition but also for *Task 4.2 vision signal processing and analyses*. In order to synchronise and fuse multiple sensor data, a framework for coordinating multiple sensors is presented in Figure 4. This framework starts from five individual sensors: Camera 1, Camera 2, Camera3, Kinect 1 and Kinect 2. The modules from 1 to 5 are responsible for gaze estimation, face and expression recognition, audio processing, body pose recognition and object tracking, respectively. It is worth noting that only 1 of 3 cameras is activated at a time for gaze estimation and face & expression recognition, and thus a special module of face detection & camera selection is proposed.

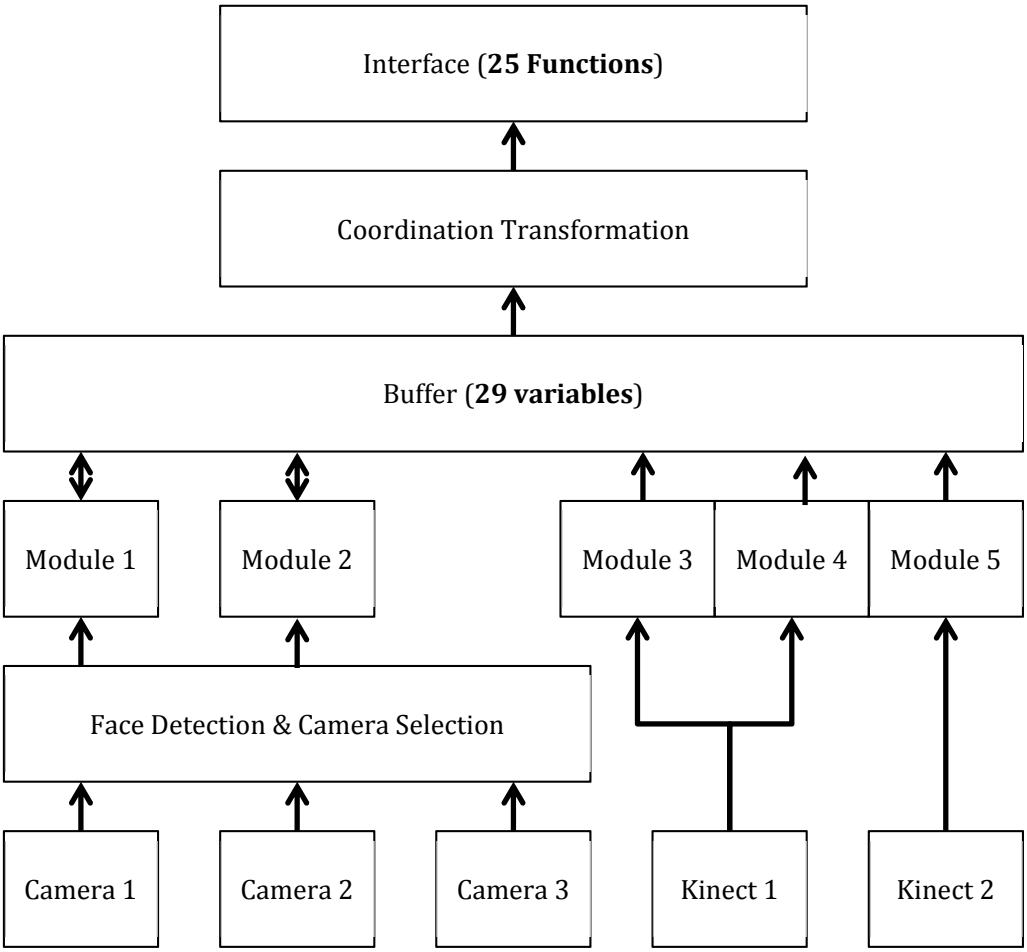


Figure 4: a framework for coordinating multiple sensors

7.7.1. Face Detection & Camera Selection

Camera 1, Camera 2 and Camera 3 form a functional unit to get the face location, eye locations, gaze direction, head direction, etc. The Face Detection & Camera Selection Module (FDCS) captures image frames from three sensors and selects one camera by the highest face detection probability, and meanwhile FDCS module also functions to obtain facial feature points from the selected frame. The selected camera ID, the original frame and the calculated feature points will be simultaneously saved in the Global Buffer and be updated according to the speed (fps). Module 1 and Module 2 serve to implement the primary functions, like calculating face/eye location, head/gaze directions, face ID, facial expression ID and etc. These separated modules will run through the main algorithms, which has been proposed/employed by this project.

The function of Kinect1 are two-folds: voice analysis (Module 3) and subject's skeleton joints extraction (Module 4). Module 3 can be further separated into two parts: speech recognition and speech direction tracking. Kinect2 focuses on object tracking, and the objective is to get object (a toy) location, object ID and head location of a robot.

7.7.2. Global Buffer Module

As described in Task 3.1, the YARP will be used to deliver information among components. Therefore, a global buffer is proposed to store a number of variables, and these variables can be considered as parameters or arguments.

global_buffer

{

V1: *cv::VideoCapture cap;*
V2: *CameradeviceID*
V3: *Mat ::image*
V4: *Eyes(rightx, righty, rightz, left x,left y, left z)*
V5: *Head pose(roll, yaw, pitch)*
V6: *Face(vector(x,y,z))*
V7: *Gaze(roll, yaw, pitch)*
V8: *Coordinate transform Mat(R,T)---camera1,camera2,camera3,Kinect2*
V9: *Frame 3D points(x,y,z)*
V10: *Object position(x,y,z)*
V11: *Head position(x,y,z)*
V12: *Hand position(x,y,z)*
V13: *cv::Mat X; //face 49 feature points*
V14: *int numberkernel; //number of kernel used for eye center detection*
V15: *vector<cv::Mat> kernel_filter; //kernel used for eye center detection*
V16: *Robot head position*
V17: *Sound Direction*
V18: *Face sorce*
V19: *Desk_point vector // 3 points*
V20: *Skelton joint vector*
V21: *Object_location*
V22: *Object_id*
V23: *Face_id*
V24: *Face_expression_id*
V25: *Object_id*
V26: *Object_history_location vector*
V27: *Voice_descriptor_id*
V28: *Voice_text_id*
V29: *Skelton_history_joint vector*

}

7.7.3. Functions and Global Variables

The relationship between 25 functions (as described in D3.1) and 29 variables is listed in Table 1. With 29 variables, predefined 25 interface functions can be easily implemented through accessing the variables based on arithmetic and coordinate transformation.

Table 2 Functions and global variables

25 Interface functions	Related Variables
F1: checkMutualGaze	V7,V8,V17
F2: getArmAngle	V20
F3: getBody	V20
F4: getBodyPose	V20
F5: getEyeGaze	V4,V7,V8,V13
F6: getEyes	V3,V4,V6,V8,V13,V14,V15
F7: getFaces	V3,V6,V8
F8: getGripLocation	V22,V21,V22
F9: getHands	V20
F10: getHead	V6,V8,V13
F11: getHeadGaze1	V6,V8,V11,V13,V19
F12: getHeadGaze2	V6,V8,V13
F13: getObjects1	V21,V22
F14: getObjects2	V19,V21,V22
F15: getObjectTableDistance	V21,V22
F16: getSoundDirection	V8,V17
F17: indntifyFace	V11,V23
F18: IdentifyFaceExpression	V11,V23,V24
F19: identifyObject	V21,V25
F20: identifyTrajectory	V25,V26
F21: identifyVoice	V27
F22: recognizeSpeech	V28
F23: trackFace	V6,V8,V18
F24: trackHand	V29
F25: trackObject	V21,V22

8. References:

1. Zhu, Z.W. and Q. Ji, *Eye gaze tracking under natural head movements.* 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol 1, Proceedings, 2005: p. 918-923.
2. Valenti, R., N. Sebe, and T. Gevers, *Combining Head Pose and Eye Location Information for Gaze Estimation.* IEEE Transactions on Image Processing, 2012. **21**(2): p. 802-815.
3. Lu, F., et al., *Inferring Human Gaze from Appearance via Adaptive Linear Regression.* 2011 Ieee International Conference on Computer Vision (Iccv), 2011: p. 153-160.
4. Sugano, Y., Y. Matsushita, and Y. Sato, *Appearance-Based Gaze Estimation Using Visual Saliency.* Ieee Transactions on Pattern Analysis and Machine Intelligence, 2013. **35**(2): p. 329-341.
5. Williams, O., A. Blake, and R. Cipolla, *Sparse and Semi-supervised Visual Mapping with the S^ 3GP.* In Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. IEEE., 2006. **1**: p. 230-237.
6. Lu, H.C., et al., *A novel method for gaze tracking by local pattern model and support vector regressor.* Signal Processing, 2010. **90**(4): p. 1290-1299.
7. Xiong, X., et al., *Eye gaze tracking using an RGBD camera.* 2014: p. 1113-1121.
8. Mora, K.A.F. and J.M. Odobez, *Geometric generative gaze estimation (G3E) for remote RGB-D cameras.* In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on (pp. 1773-1780). IEEE., 2014.
9. Shivappa, S.T., M.M. Trivedi, and B.D. Rao, *Audiovisual information fusion in human--computer interfaces and intelligent environments: A survey.* 2010. **98**(10:1692-1715).
10. Smith, D. and S. Singh, *Approaches to multisensor data fusion in target tracking: A survey.* 2006. **18**(12:1696-1710).
11. Jaimes, A. and N. Sebe, *Multimodal human-computer interaction: A survey.* 2007. **108**(1:116-134).
12. Sun, S., et al. *Adaptive sensor data fusion in motion capture*. 2010. Information Fusion (FUSION), 2010 13th Conference on.
13. Matzka, S. and R. Altendorfer, *A comparison of track-to-track fusion algorithms for automotive sensor fusion.* 2009(69-81).
14. Lazarus, S.B., et al., *Vehicle localization using sensors data fusion via integration of covariance intersection and interval analysis.* 2007. **7**(9:1302-1314).
15. Luo, R.C., Y.C. Chou, and O. Chen. *Multisensor fusion and integration: algorithms, applications, and future research directions*. 2007. Mechatronics and Automation, 2007. ICMA 2007. International Conference on.
16. Rattani, A., et al. *Feature level fusion of face and fingerprint biometrics*. Biometrics: Theory, Applications, and Systems, 2007. BTAS 2007. First IEEE International Conference on.
17. Lavanya, A., et al. *Image fusion of the multi-sensor lunar image data using wavelet combined transformation*. 2011. Recent Trends in Information Technology (ICRTIT), 2011 International Conference on.
18. Li, Z., et al. *Using semi-blind source separation in multi-sensor fault location of satellite attitude determination system*. 2010. Information Computing and Telecommunications (YC-ICT), 2010 IEEE Youth Conference on.
19. cheng, J., et al., *Feature fusion for 3D hand gesture recognition by learning a shared hidden*

space. 2012. **33**(4:476-484).

20. Thiemjarus, S., A. James, and G.Z. Yang, *An eye--hand data fusion framework for pervasive sensing of surgical activities.* 2012.
21. Starzacher, A. and B. Rinner. *Embedded realtime feature fusion based on ANN, SVM and NBC*. 2009. Information Fusion, 2009. FUSION'09. 12th International Conference on.
22. Zhang, Y. and Q. Ji, *Efficient sensor selection for active information fusion.* 2010. **40**(3:719-728).
23. Zhang, X., et al., *A Framework for Hand Gesture Recognition Based on Accelerometer and EMG Sensors.* 2011(99:1-13).
24. Garg, A., et al., *Frame-dependent multi-stream reliability indicators for audio-visual speech recognition.* 2003. **3**(605-615).
25. Zeng, Z., et al., *Audio--visual affective expression recognition through multistream fused HMM.* 2008. **10**(4:570-577).
26. Liu, S., et al., *Multi-Sensor Data Fusion for Physical Activity Assessment.* 2011. **99**(1-10).
27. Zhan, Y., et al., *Automated speaker recognition for home service robots using genetic algorithm and Dempster--Shafer fusion technique.* 2009. **58**(9:3058-3068).
28. Stergiou, A., A. Pnevmatikakis, and L. Polymenakos, *A decision fusion system across time and classifiers for audio-visual person identification.* 2007(223-232).
29. Zouba, N., F. Bremond, and M. Thonnat. *Multisensor fusion for monitoring elderly activities at home*. in *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*. 2009. IEEE.
30. Keller, Y., et al., *Audio-visual group recognition using diffusion maps.* 2010. **58**(1:403-413).
31. Zhang, Z., Z. Huang, and J. Wu. *Hierarchical information fusion for human upper limb motion capture*. in *Information Fusion, 2009. FUSION'09. 12th International Conference on*. 2009. IEEE.
32. Luo, R.C. and K.L. Su, *Multilevel multisensor-based intelligent recharging system for mobile robot.* 2008. **55**(1:270-279).
33. Heracleous, P., et al. *Exploiting multimodal data fusion in robust speech recognition*. in *Multimedia and Expo (ICME), 2010 IEEE International Conference on*. 2010. IEEE.
34. Hartley, R. and A. Zisserman, *Multiple view geometry in computer vision*. 2003: Cambridge university press.
35. Gao, X.-S., et al., *Complete solution classification for the perspective-three-point problem.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2003. **25**(8): p. 930-943.
36. Kneip, L., D. Scaramuzza, and R. Siegwart. *A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation*. in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. 2011. IEEE.
37. Ansar, A. and K. Daniilidis, *Linear pose estimation from points or lines.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2003. **25**(5): p. 578-589.
38. Garro, V., F. Crosilla, and A. Fusiello. *Solving the pnp problem with anisotropic orthogonal procrustes analysis*. in *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*. 2012. IEEE.
39. Lu, C.-P., G.D. Hager, and E. Mjolsness, *Fast and globally convergent pose estimation from video images.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2000. **22**(6): p. 610-622.

40. Schweighofer, G. and A. Pinz. *Globally Optimal O (n) Solution to the PnP Problem for General Camera Models*. in *BMVC*. 2008.
41. Laptev, I., *On space-time interest points.* International Journal of Computer Vision, 2005. **64**(2-3): p. 107-123.
42. Laptev, I., et al., *Learning realistic human actions from movies.* 2008 Ieee Conference on Computer Vision and Pattern Recognition, Vols 1-12, 2008: p. 3222-3229.
43. Dalal, N. and B. Triggs, *Histograms of oriented gradients for human detection.* 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol 1, Proceedings, 2005: p. 886-893.
44. Campbell, L.W. and A.F. Bobick, *Recognition of human body motion using phase space constraints.* Fifth International Conference on Computer Vision, Proceedings, 1995: p. 624-630.
45. Lv, F. and R. Nevatia, *Recognition and segmentation of 3-D human action using HMM and multi-class AdaBoost.* Computer Vision - Eccv 2006, Pt 4, Proceedings, 2006. **3954**: p. 359-372.
46. Han, L., et al., *Discriminative human action recognition in the learned hierarchical manifold space.* Image and Vision Computing, 2010. **28**(5): p. 836-849.
47. Ning, H., et al., *Latent pose estimator for continuous action recognition*, in *Computer Vision–ECCV 2008*. 2008, Springer. p. 419-433.
48. M, M., et al., *Motion templates for automatic classification and retrieval of motion capture data*, in *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 2006, Eurographics Association: Vienna, Austria.
49. Li, L. and B.A. Prakash. *Time series clustering: Complex is simpler!* in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 2011.
50. Li, W., Z. Zhang, and Z. Liu. *Action recognition based on a bag of 3d points*. in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*. 2010. IEEE.
51. Wang, J., et al., *Robust 3D Action Recognition with Random Occupancy Patterns.* Computer Vision - Eccv 2012, Pt Ii, 2012. **7573**: p. 872-885.
52. Yang, X. and Y. Tian. *Eigenjoints-based action recognition using naive-bayes-nearest-neighbor*. in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. 2012. IEEE.
53. Vieira, A.W., et al., *Stop: Space-time occupancy patterns for 3d action recognition from depth map sequences*, in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. 2012, Springer. p. 252-259.
54. Yun, K., et al. *Two-person interaction detection using body-pose features and multiple instance learning*. in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. 2012. IEEE.
55. Yang, X., C. Zhang, and Y. Tian. *Recognizing actions using depth motion maps-based histograms of oriented gradients*. in *Proceedings of the 20th ACM international conference on Multimedia*. 2012. ACM.
56. Chaudhry, R., et al. *Bio-inspired dynamic 3d discriminative skeletal features for human action recognition*. in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*. 2013. IEEE.
57. Raptis, M., D. Kirovski, and H. Hoppe. *Real-time classification of dance gestures from*

skeleton animation. in *Proceedings of the 2011 ACM SIGGRAPH/Eurographics symposium on computer animation*. 2011. ACM.

58. Jain, A.K., B. Klare, and U. Park. *Face recognition: Some challenges in forensics*. in *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*. 2011. IEEE.
59. Sariyanidi, E., H. Gunes, and A. Cavallaro, *Automatic analysis of facial affect: A survey of registration, representation and recognition.* 2014.
60. Zeng, Z., et al., *A survey of affect recognition methods: Audio, visual, and spontaneous expressions.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2009. **31**(1): p. 39-58.
61. Yeasin, M., B. Bullot, and R. Sharma. *From facial expression to level of interest: a spatio-temporal approach*. in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. 2004. IEEE.
62. Ojala, T., M. Pietikainen, and T. Maenpaa, *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2002. **24**(7): p. 971-987.
63. Ojansivu, V. and J. Heikkilä, *Blur insensitive texture classification using local phase quantization*, in *Image and signal processing*. 2008, Springer. p. 236-243.
64. Ahonen, T., A. Hadid, and M. Pietikainen, *Face description with local binary patterns: Application to face recognition.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2006. **28**(12): p. 2037-2041.
65. Valstar, M.F., et al. *The first facial expression recognition and analysis challenge*. in *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*. 2011. IEEE.
66. Zhao, G. and M. Pietikainen, *Dynamic texture recognition using local binary patterns with an application to facial expressions.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2007. **29**(6): p. 915-928.
67. Jiang, B., et al., *A dynamic appearance descriptor approach to facial actions temporal modeling.* Cybernetics, IEEE Transactions on, 2014. **44**(2): p. 161-174.
68. Jiang, B., M.F. Valstar, and M. Pantic. *Action unit detection using sparse appearance descriptors in space-time video volumes*. in *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*. 2011. IEEE.
69. Valstar, M.F. and M. Pantic, *Fully automatic recognition of the temporal phases of facial actions.* Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 2012. **42**(1): p. 28-43.
70. Rudovic, O., M. Pantic, and I. Patras, *Coupled Gaussian processes for pose-invariant facial expression recognition.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2013. **35**(6): p. 1357-1369.
71. Wang, Z., S. Wang, and Q. Ji. *Capturing complex spatio-temporal relations among facial muscles for facial expression recognition*. in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. 2013. IEEE.
72. Huang, D., et al., *Local binary patterns and its application to facial image analysis: a survey.* Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 2011. **41**(6): p. 765-781.
73. Wu, Y., J. Lim, and M.H. Yang, *Online Object Tracking: A Benchmark.* 2013 IEEE

Conference on Computer Vision and Pattern Recognition (CVPR), 2013: p. 2411-2418.

74. Babenko, B., M.H. Yang, and S. Belongie, *Visual Tracking with Online Multiple Instance Learning.* Cvpr: 2009 Ieee Conference on Computer Vision and Pattern Recognition, Vols 1-4, 2009: p. 983-990.
75. Comaniciu, D., V. Ramesh, and P. Meer, *Kernel-based object tracking.* Ieee Transactions on Pattern Analysis and Machine Intelligence, 2003. **25**(5): p. 564-577.
76. Wang, D., H.C. Lu, and M.H. Yang, *Online Object Tracking With Sparse Prototypes.* Ieee Transactions on Image Processing, 2013. **22**(1): p. 314-325.
77. Fan, J.L., X.H. Shen, and Y. Wu, *Scribble Tracker: A Matting-Based Approach for Robust Tracking.* Ieee Transactions on Pattern Analysis and Machine Intelligence, 2012. **34**(8): p. 1633-1644.
78. Bao, C.L., et al., *Real Time Robust L1 Tracker Using Accelerated Proximal Gradient Approach.* 2012 Ieee Conference on Computer Vision and Pattern Recognition (Cvpr), 2012: p. 1830-1837.
79. Ross, D.A., et al., *Incremental learning for robust visual tracking.* International Journal of Computer Vision, 2008. **77**(1-3): p. 125-141.
80. Mei, X., et al., *Efficient Minimum Error Bounded Particle Resampling L1 Tracker With Occlusion Detection.* Ieee Transactions on Image Processing, 2013. **22**(7): p. 2661-2675.
81. Qi, Z.Q., et al., *Online multiple instance boosting for object detection.* Neurocomputing, 2011. **74**(10): p. 1769-1775.
82. Kalal, Z., J. Matas, and K. Mikolajczyk, *P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints.* 2010 Ieee Conference on Computer Vision and Pattern Recognition (Cvpr), 2010: p. 49-56.
83. Babenko, B., M.H. Yang, and S. Belongie, *Robust Object Tracking with Online Multiple Instance Learning.* Ieee Transactions on Pattern Analysis and Machine Intelligence, 2011. **33**(8): p. 1619-1632.
84. Zhang, K.H., L. Zhang, and M.H. Yang, *Real-Time Compressive Tracking.* Computer Vision - Eccv 2012, Pt Iii, 2012. **7574**: p. 864-877.
85. Candes, E.J. and T. Tao, *Decoding by linear programming.* Ieee Transactions on Information Theory, 2005. **51**(12): p. 4203-4215.
86. Zhang, K.H., et al., *Fast Visual Tracking via Dense Spatio-temporal Context Learning.* Computer Vision - Eccv 2014, Pt V, 2014. **8693**: p. 127-141.
87. Bengio, Y., R. De Mori, and R. Cardin. *Speaker independent speech recognition with neural networks and speech knowledge*. in *Advances in neural information processing systems*. 1990.
88. Giacobello, D., et al., *Sparse linear prediction and its applications to speech processing.* Audio, Speech, and Language Processing, IEEE Transactions on, 2012. **20**(5): p. 1644-1657.
89. Rakesh, K., S. Dutta, and K. Shama, *Gender Recognition using speech processing techniques in LABVIEW.* International Journal of Advances in Engineering & Technology, 2011. **1**(2): p. 51-63.
90. Polur, P.D. and G.E. Miller, *Experiments with fast Fourier transform, linear predictive and cepstral coefficients in dysarthric speech recognition algorithms using hidden Markov model.* Neural Systems and Rehabilitation Engineering, IEEE Transactions on, 2005. **13**(4): p. 558-561.
91. Hai, J. and E.M. Joo. *Improved linear predictive coding method for speech recognition*. in

Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on. 2003. IEEE.

92. Sayem, A., *Speech Analysis for Alphabets in Bangla Language: Automatic Speech Recognition.* International Journal of Engineering Research, 2014. **3**(2): p. 88-93.
93. Sunny, S., S. David Peter, and K.P. Jacob, *Design of a Novel Hybrid Algorithm for Improved Speech Recognition with Support Vector Machines Classifier.* 2013.
94. Sunny, S., S. David Peter, and K.P. Jacob, *Discrete Wavelet Transforms and Artificial Neural Networks for Recognition of Isolated Spoken Words.* International Journal of Computer Applications, 2012. **38**(9): p. 9-13.
95. Juang, B.-H.F., *On the hidden Markov model and dynamic time warping for speech recognition—A unified view.* AT&T Bell Laboratories Technical Journal, 1984. **63**(7): p. 1213-1243.
96. Shady, Y. and S.H.H. Zayed, *Speaker independent Arabic speech recognition using support vector machine*. 2009, Department of Electrical Engineering, Shoubra Faculty of Engineering, Benha University, Cairo, Egypt.
97. Priyadarshani, P., N. Dias, and A. Punchihewa. *Dynamic Time Warping based speech recognition for isolated Sinhala words*. in *Circuits and Systems (MWSCAS), 2012 IEEE 55th International Midwest Symposium on*. 2012. IEEE.
98. Viola, P. and M.J. Jones, *Robust real-time face detection.* International Journal of Computer Vision, 2004. **57**(2): p. 137-154.
99. Daugman, J.G., *High Confidence Visual Recognition of Persons by a Test of Statistical Independence.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1993. **15**(11): p. 1148-1161.
100. Xiong, X.H. and F. De la Torre, *Supervised Descent Method and its Applications to Face Alignment.* 2013 Ieee Conference on Computer Vision and Pattern Recognition (Cvpr), 2013: p. 532-539.
101. Dementhon, D.F. and L.S. Davis, *Model-Based Object Pose in 25 Lines of Code.* International Journal of Computer Vision, 1995. **15**(1-2): p. 123-141.
102. Zhang, K.H., L. Zhang, and M.H. Yang, *Fast Compressive Tracking.* Ieee Transactions on Pattern Analysis and Machine Intelligence, 2014. **36**(10): p. 2002-2015.