GAME-ABLING: Platform of Games for People with Cerebral Palsy

Jaume Vergés-Llahí[1], Hamed Habibi[2], Fran Casino[2] and Domènec Puig[2]

[1]Ateknea Solutions Catalonia (ASC),
Victor Pradera, 45, ES-08940 Cornell de Llobregat, Barcelona, Spain
[2]Universitat Rovira i Virgili (URV),
Campus Sescelades, Av. Pasos Catalans, 26, ES-43007 Tarragona, Spain
jaume.verges@ateknea.com,
{hamed.habibi, fran.casino, domenec.puig}@urv.cat
http://www.ateknea.com, http://www.urv.cat

Abstract. We present the FP7 European Project GAME-ABLING developed from December 2012 to January 2015. This project aimed at the development of a platform for the creation of games for patients with Cerebral Palsy (CP). A key point of the platform is that the framework can be used by personal with no specific skill in game creation, permitting caregivers and parents its utilization. The system is composed of (i) a framework that encompasses the several tools developed to run and control the games, (ii) the authoring tool to easily allows the creation of new games, and (iii) the analyzing tool that generate statistics on the impact of the games in CP patients. Due to motor and cognitive constraints of CP patients, specific sets of games were developed. Also an extensive group of peripherals can be employed beyond the usual game controllers, including color and depth cameras, Nintendo Wiimote and balance boards. This article describes the system elements and the results obtained during the evaluation of the games with real patients. Special detail is given to the analysis of the movements of the user's head and hands that is employed to control the games.

1 Introduction

Cerebral Palsy (CP) is one of the most frequently conditions in childhood, with an incidence of 2 per 1,000 live births. In the EU there is 1.3 out of 15 million persons with CP in the world. This neurological disorder affects body movement, balance and posture and almost always is accompanied by other cognitive or sensory impairments like mental retardation, deafness and vision problems. The severity of these problems varies widely, from very mild and subtle to very profound. These disabilities lead to an inactive lifestyle which reduces the patients physical health, social participation, and quality of life. Therapy costs (up to 45,000 by year) cannot be afforded by most of the families. Playing Video games is a useful treatment that promotes and maintains more active and healthful lifestyle in these persons. However accessibility to videogames is hardly applied for them.

The FP7 Project GAME-ABLING has developed a software tool for creating interactive video games in an intuitive manner in such a way that non-expert personnel

Puig D., VergÃĺs-LlahÃ■ J., Casino F. and Aghdam H.
GAME-ABLING: Platform of Games for People with Cerebral Palsy.
DOI: 10.5220/0006164300640087
In *European Project Space on Computer Vision, Graphics, Optics and Photonics (EPS Berlin 2015)*, pages 64-87
ISBN: 978-989-758-156-4

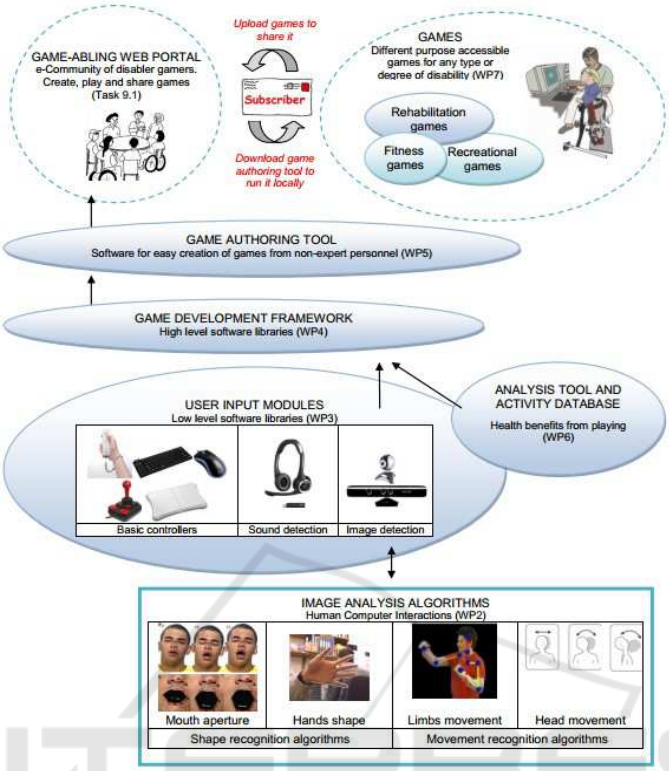

Fig. 1. Different elements composing the GAME-ABLING system.

(caregivers and parents) can develop customized games. GAME-ABLING are accessible games with the aim of improve physical activity of disabled people. This is a different approach that benefits both caregivers and therapists, who are able to design and easily customize games for their patients, and also for disabled people, who are able to play with the games developed by caregivers of therapist improving their fitness while playing.

Games created by GAME-ABLING can be controlled in a variety of ways, allowing the use of body movements and voice. The images recorded by the webcam are processed using computer vision techniques in order to track head, eyes, and hands as well as to detect some basic gestures like hand shapes (open hand, closed fist or hand pointer shape) and face gestures (aperture of mouth and opening/closing the eyes). Voice tones and all these movements and gestures are also translated into user actions that could be used to control the videogames, similar to a joystick or a mouse.

GAME-ABLING also supports other type of standard input game controllers such as mouse, keyboard, joystick, Wii controller, and balance board. According with the type and severity of disability of patient, caregivers can decide which type of game design and which type of game controller input can be used. Game performance and player feedback is registered in a database and analyzed by caregivers or therapists to determine the physical fitness improvement.

It can be argued against the need of creating a new platform to create games without taking into account other similar existing solutions such as SCRATCH. The reason to do that is double folded. First, our approach aimed at a user segment that did not require any specific training in the creation of games. Secondly, the specific requirements of CP patients obliged to deal only with a limited family of games, which accordingly to experts, were proven to have a usability within our sector.

The article is distributed as follows. First, In Section 2 a general overview of the project and the system developed in the GAME-ABLING platform is presented. In Sections 3 and 4 a more detailed description of the algorithms and techniques employed to extract the motion corresponding to the user's head and hands are taken into account. Finally, the article ends with the Conclusions are in Section 5.

2 System Overview

The GAME-ABLING system is composed of the following elements from bottom to top: (i) image analysis module, (ii) user input modules, (iii) game development framework, (iv) game authoring tool, (v) analysis tool and activity database, and (vi) games. The structure of the complete system can be seen in Fig. 1.

The idea of the whole system is to provide the tools to create, use, evaluate, and share the games by a community of users with specific requirement (CP patients) and non-expert programming skills (caregivers and parents). We are going to describe each part in the following subsections.

2.1 Image Analysis Module

This module is able to capture movements of the different parts of the body, especially focusing in the obtaining of head and hands movement. This module was developed in four phases, namely, data collection, developing image analysis algorithms, code optimization and testing. In addition, we carried out the analysis of head and hands in parallel since they are two independent problems, and employed two different types of camera, generalist RGB cameras and color and depth cameras, such as Kinect.

Data Collection. During the GAME-ABLING project, we have created a database of videos recorded at gaming sessions in the facilities of one of our partners, the Associació Provincial de Paràlisi Cerebral (APPC) in Tarragona, Spain. This database allows testing the algorithms using video sequences in similar conditions as those encountered during the use of the games. Furthermore, we annotated the some images of this dataset manually to train our head and body-part detectors. We also collected another dataset in APPC and developed an annotation software and annotated them in URV. Later, we used the database for refining our classification model. These databases were created scrupulously following the ethical issues involved in the obtaining and processing of such data from Cerebral Palsy patients for image analysis purposes.

Image Analysis. A series of algorithms for analyzing and tracking the head movements based on skin color segmentation and a state-of-art face detection method were developed. These algorithms work with standard color webcams and the depth cameras. Although the algorithms were accurate and fast with health people, there were

some practical challenges when tested in real scenarios on the patients with various severity level. We observed that the algorithm worked accurately with level 1 patients, which are those with the least level of impairment. Notwithstanding, level 4 and level 5 patients (the highest levels) were not able to control the games due to the fact that their facial and body-part appearance as well as movements are much more different than the patients with severity level 1 and 2.

In order to solve such difficulties we also utilized the depth information for segmentation purposes. By this way, we are able to increase the computational efficiency and also make the algorithm more robust against illumination. We also make use of the shape information to find the head of the patients. This approach can deal with one of the important challenges that we faced in the first period: the fact that patients with higher severity level are barely able to control their heads and hold it in a frontal or near-frontal position. As the results, the state-of-are methods were not able to detect the face and, consequently, the users were not able to control the games.

Fig. 2. APPC patients playing during the database creation.

2.2 User Input Modules

A series of libraries were developed in the project to access different input devices to control the game through a wide range of possibilities, i.e., direct motion controllers, image-based controllers, and audio-based controller. These libraries allow an application to read inputs from the following list of devices: Nintendo Balance board, usual game controllers (joysticks, gamepads, and switch buttons), Nintendo Wiimote controller, keyboard and mouse, color and depth cameras, and a microphone. The inputs from these devices can now be read, adapted when necessary and used by games to

perform different game actions. As regards the audio devices, a simple method to capture and measure the intensity of the voice or blows into the microphone allowing the device to control some game actions. The inclusion of cameras as game controllers was accomplished as two modules for integrating Microsoft Kinect and color cameras.

2.3 Game Development Framework

The main core of the system consists of the common game framework. A configurable launcher was created for two different kinds of games, the XY and the ”Spot-the-difference” games. The launcher is capable of loading the game description and assets, loading different input modules, displaying a control panel and finally allowing the user to play the game. The work was focused on the development of a software framework that assisted in the development of configurable games and the integration of general purpose and specialized input devices in order to support a wide range of patients/disabilities. The approach followed was to abstract user input and to decouple devices from games, which facilitates the adaptation of games to future use cases. Moreover, a modular architecture was designed to allow for the independent development and deployment of base games and input modules. This is important as it helped the development throughout the project lifetime but also facilitates future developments and allows the introduction of a licensed development scheme; allows the commercialization of modules; helps the deployment process when different platforms are involved and when upgrading an existing installation. Finally, a central application is provided that coordinates all existing modules and presents the user with a single environment through which to configure and control game sessions

2.4 Game Authoring Tool

The main objective of this tool is to provide an intuitive, easy to use with no special expertise required, authoring environment that combines configurable games and the various game elements into complete ready to played - games. The approach followed was the development of a web application, capable also to be executed offline and supporting all major browsers and a step by step approach in developing the games. A common look and feel was adapted across all base games, although the game play of each base game is considerable different. The authoring tool allows to load user-created assets, like avatars, sounds, backgrounds, and their position in the window.

A number of considerations were taken into account in the design of the authoring tool: (i) usability: the solution is very easy to use and self-explanatory, (ii) deployment: the solution is hassle-free of installation and OS considerations, and (iii) effectiveness: The solution is effective in generating a wide range of games.

2.5 Analysis Tool and Activity Database

This module generates for each patient databases with data of the performance of the patients during the gamming sessions and an activity report that visualizes it creating graphs related with the patient performance. The relational database is capable of storing two set of information: configuration of individual games/users and the actual data

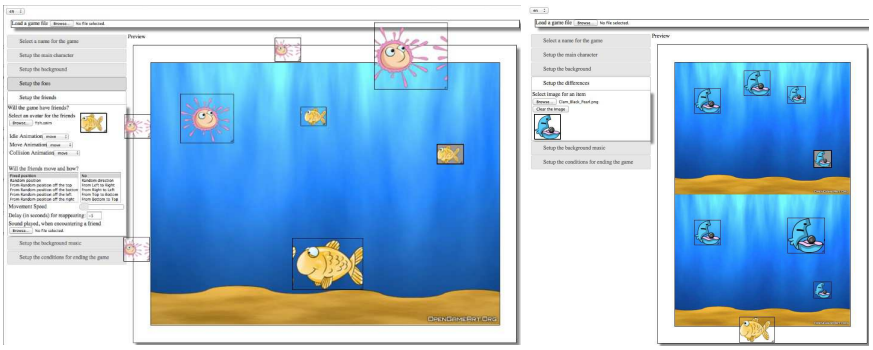

Fig. 3. Snapshots of the Authoring Tool creating an XY game.

of the played games. The purpose of this database is to use data to analyze the development of the patient. The analysis tool was developed in order to study the information stored in the database. This tool allows to access, combine and visualize information stored in the database according to criteria of specialized personnel (therapists, caregivers, and psychologists), so they get the best possible profit of the activity database. It is a graphical web interface tool with diagrams and statistics, which inform the interested users of the evolution of each patient, and based on this information, they will be able for further customization the games for each patient. The data generated during the games is anonymized and follows the strict requirements of the Spanish Data Protection Act (LOPD).

2.6 Implemented Games

Following the suggestions of the CP experts involved in the project, three main families of game typologies were implemented in the project, namely, XY games, spot-the-difference games, and memory games:

XY Games: these are games where an avatar is moving in one or two axes: vertical and/or horizontal and the main goal of the player is to collect items (Friends) and avoid enemies (foes). With each Friend collect it, he/she gains points and with each foe hit he/she loses lives.

Spot-the-Difference Games: in these games the player needs to find small differences in two images that look identical.

Memory Game: Here the player is presented with a grid of images, where every image exists twice. After few seconds all the images are hidden and the player needs to select the same images sequentially.

Each one of these groups of games envisages to create an improvement in either motor or cognitive capabilities of the patients, such as coarse/fine hand and limb dexterity, cause/effect, or visual memory. A series of scored and game variables are recorded during the gamming sessions in order to be analyzed by the caregivers or doctors later on and objectively evaluate the effect and of the game on the patient.

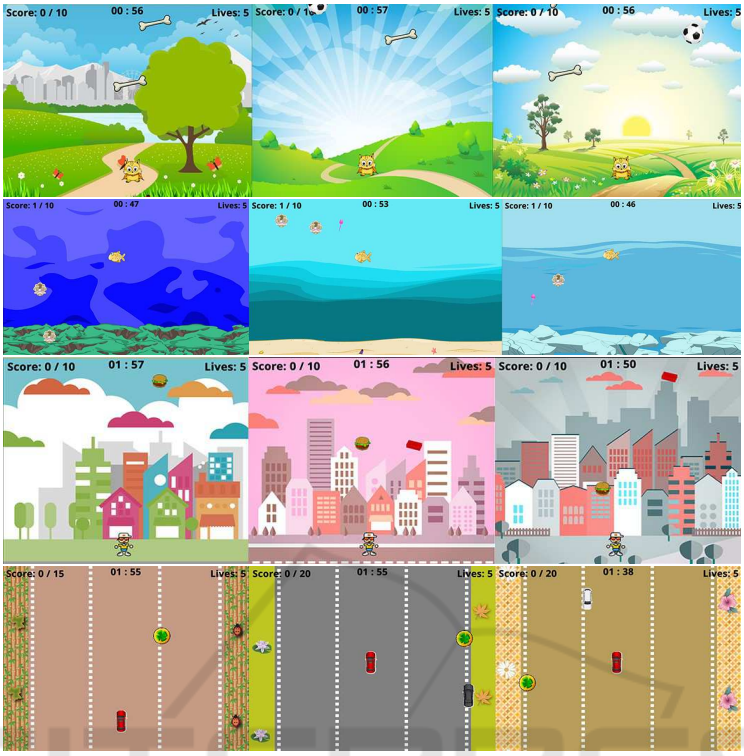

Fig. 4. Snapshots of some of the games developed with the Authoring Tool.

2.7 Evalutation

Testing of the Game Authoring Tool (GAT) and Rehabilitation Games developed for patients with Cerebral Palsy of different levels of motor impairment and age was done. This was done by specialists from International Clinic of Rehabilitation (ICR) in Ukraine and the Associaci Provincial de Parlisi Cerebral de Tarragona (APPC) in Span, both partners of the project.

Evaluation of the GAT was done by the following procedure. Therapists were taught how to use the GAT and supplied with the assets to develop games. They also learned how to use games with different gaming hardware balance board, Kinect sensor, camera, special goniometer joystick, and others. They developed their own games that were checked by a supervisor.

Totally 12 games were selected for testing on patients with Cerebral Palsy. Therapists filled out the questionnaire that was analyzed and conclusions about the GAT usability have been made. Evaluation of the games was done on 32 patients with Cerebral Palsy of different level of motor disability. Every patient participated in 6 to 8 gaming sessions of 15-20 min duration under the supervision of the therapist. During training sessions different gaming hardware has been used Balance board, Kinect sensor, camera, special goniometer-joystick and keyboard or mouse. After the sessions parents

were interviewed and filled out the questionnaire that was later analyzed. Parents and therapists gave important suggestions for future development.

Analysis of the questionnaires indicates that the majority of the patients highly rated rehabilitation games - 82% ranked the rehabilitation games as excellent and 18% as good, and nobody stated that the games are of poor or average quality. Also the majority of the parents liked graphical and sound effects of the games and stated that the game difficulty was adjusted properly to the motor abilities of the child. Parents suggested in their comments that there should be a larger collection of the games of different difficulty levels with nice animated characters. They suggested that there also should be games aimed at not only motor development but also at training of cognitive functions. There were also suggestions to integrate the games into social networks for disabled children.

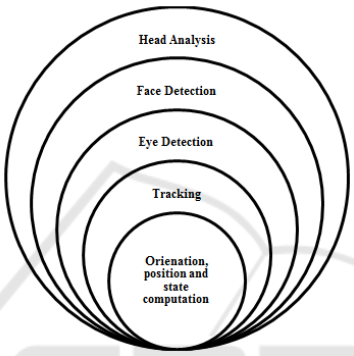

Fig. 5. Head Analysis Problem in GAME-ABLING project.

3 Head Analysis

3.1 Problem Formulation

One of the machine vision tasks in the GAMEABLING project is to obtain information in regards to the head. This information includes position and orientation of the head, short-term and long-term tracking of the head and tracking the location and state of eyes. Fig.5 shows the relationship between such different problems.

In this project, the Microsoft Kinect sensor was utilized for capturing the images. One of the main reasons was that the use of depth information helped to solve hand analysis problem. Therefore, we also take advantage of such information in order to speed up the processing and to calculate the pose (orientation and position) of the head in 3D Cartesian space. Regardless of the type of the video stream, the developed algorithm has some mandatory requirements which are depicted in Fig.6

According to this figure, the algorithm must be real-time. In other words, the developed algorithm must be able to analyze more than 15 frames per second. Moreover, it should calculate the pose of the head accurately and track the facial components robustly. However, considering that CP patients are not capable of controlling their head accurately and Kinect is a low-end camera (with considerable noise in the video stream), we cannot expect the system to have 5 degree (or less) accuracy in orientation.

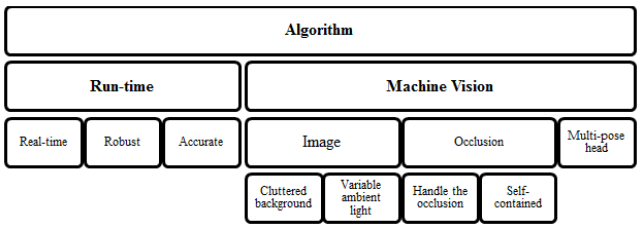

Fig. 6. Head Analysis Problem in GAME-ABLING project.

Regarding the machine vision requirements, the algorithm must be able to cope with cluttered background and variable ambient light. Also, it must be able to deal with partial occlusion (e.g. moving the hand in front of the eye) which is a frequent situation. Finally, the most critical assumption about the algorithm is the pose of the head in relation to the camera.

3.2 Technical Review of the Algorithms

The first step in estimating the head pose is to localize the face in the input image. One of the most successful and most widely used algorithms for detecting the near- frontal faces in the given image is the Viola-Jones detector [14]. This algorithm uses a bank of Haar-like rectangular filters with a cascade classifier to find the faces in the image. Although this algorithm works well with frontal faces and static images, it is impractical in the GAME-ABLING project. This is due to the fact that the depth of the cascade classifier is large and, hence, it is computationally intractable in real-time applications. Moreover, it may produce false-negative outcomes in cluttered background. Notwithstanding, there is a built-in function in OpenCV for face detection. According to our experiments, this algorithm can run with the rate of 8-9 frames-per-second (fps). For this reason, we developed our own face detection and tracking algorithm which utilizes the depth information and achieves a real-time performance.

The next step for estimating the head pose is to localize the facial components such as eye-corners on the face image. Vukadinobic and Pantic [15] proposed a method for facial feature point detection based on bank of Gabor filters which is able to detect 20 facial feature points in neutral face images. They used 48 Gabor filters for extracting features of a 1313 image patches. Their feature vector comprises from gray-level values of the patch plus the results obtained by applying Gabor filters on the patch. This makes a feature vector containing 8281 features. However, convolving the all Gabor filters with the input image is a time consuming task. Moreover, the length of the feature vector is too large that increases it computational cost dramatically.

In GAME-ABLING project we used local binary pattern (LBP) [16] for extracting features of the image patches and localizing the facial components. LBP is a computationally efficient algorithm for extracting texture features. In this method, for each pixel in the image patch, we scan its surrounding pixels. If the pixel has greater value than the central pixel it is replaced with 1 otherwise it is replaced with 0. After all pixels are checked, an 8-bit binary number is produced which is converted to a decimal number. Each pixel produces a decimal number. Histogram of these decimal numbers in the image patch is called local binary pattern feature. It is clear that using 8bit number, the

LBP histogram will contain 256 bins. Later, this definition was refined and another LBP histogram called uniform local binary pattern was proposed [16]. Using 8-bit number, the uniform LBP produces a histogram with 59-bins. In this project, we implemented a highly optimized version of LBP algorithm which is computationally efficient.

Having the face and facial component localized, the final step is to estimate the head pose. Efforts for solving this problem break down into 8 different classes [1]: template based methods, detector arrays, nonlinear regression, nonlinear dimension reduction, model based, geometry based, tracking methods and hybrid methods. As a template based method, Baily and Milgram [2], first, generated several images of a person in different poses. Estimating the head pose of an input image was done by matching it with all images in database. There are several studies [3, 4] which have employed nonlinear reduction techniques [5–7] for nonlinear mapping of high dimensional image onto a low dimensional manifold. After mapping the training images onto the manifold, system must learn how to embed the new input image onto the manifold. Erik and Mohan [8] proposed a nonlinear regression technique. In this method, localized gradient orientation histogram feature extraction method is applied on the input image. Then, using this feature vector and three different support vector regressors head pose is estimated. Baily et al [9] used Haar like rectangular filters for feature extraction and generalized neural network for head pose estimation.

In addition, Tedora et al [10] utilized feed-forward neural network and facial feature points. Model based approaches tries to learn a model for face image. Among object modeling methods, active shape model [17] and active appearance model [11] are popular methods for modeling human face. After the model is learned using training images, it is fitted on input image. Then, we can use similar approach in [10] for head pose estimation. For a complete survey on head pose estimation refer to [1]. Martins and Batista [11, 12] employs active appearance model for locating facial feature points. Then, they mapped 2D facial feature points to specialized 3D rigid facial model. Finally, head pose estimated using POSIT algorithm. This method cannot deal with non-rigid face motions sine it does not fit 3D model on 2D facial points. In order to estimate the head pose in the GAME-ABLING project, we implemented our own algorithm which utilizes the depth information to achieve its goal. The developed algorithm runs in real-time and it is able to robustly estimate the head pose.

Table 1. Publicly available database for research.

Database	No. of images	Image size	Landmarks	Color
CIE-Biometrics	9951	2048×1536	30	Color
BioID	1521	384276	20	Grayscale
FDDB	28204	Variable	NA	Color
MUCT	3755	480640	76	Color
IMM	240	648480	58	Color/Grayscale

3.3 Databases

Before implementing an algorithm, we need a database to train our classifiers. There are different public databases in Internet which can be used for this purpose. However,

there are a few databases that include the information about the landmarks of the faces. The collected databases are listed in Table 1

The important issue about these databases is that only a few of them are useful in GAME-ABLING project. To be more specific, FDDB is useful for training a RGB based face detector. However, because our algorithm does not strongly rely on face detection, we did not use this database in this project. The databases CIE-Biometrics, MUCT and IMM are useful for research purposes. This is due to the fact that the images of these databases have been taken in controlled environment and in a neutral facial expression. Notwithstanding, neutral expression of CP patients usually is different from healthy people. For this reason, the geometry and the texture of the landmarks is different from the images in this databases. Moreover, these databases have utilized high-resolution cameras to capture the images. But, the quality of the images taken by cheap RGB cameras and Kinect sensor are low and for this reason the texture of the face will be different. Consequently, these databases were not used in this project.

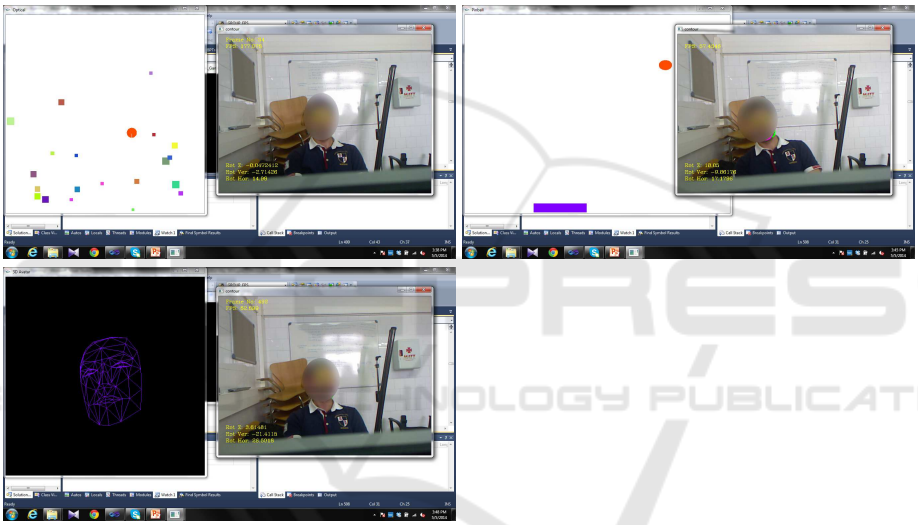

Fig. 7. Screen shot of the developed games for collecting realistic data.

Instead, we collected our own database in APPC and URV. To this end, we developed three games in which the user can control the hero of the game using head motions and head posture. Fig.7 illustrates the screen shot of the developed games. In the game shown in the first row the user controls the game character (red circle) to collect the moving squares. The hero has four different motions namely up, down, left and right. Also, the speed of the hero is controlled using the distance of the user from camera. The movements of the hero are controlled using the head motions or head posture.

The second row shows a game in which there is a board in the bottom of the screen that can move to left and right. The user controls the movement of the board using head posture or moving his/her body to right or left. Finally, the third row is a 3D face mask which is controlled by head posture and head position.

In all three cases we asked the patients to play the games. Then, the user interactions

were recorded while he/she was playing the games. Using this approach we were able to collect the images in realistic conditions. We repeated the same procedure in our laboratory and collected some data from ourselves by playing the same games.

After collecting the new images in APPC and URV, we developed another application to annotate the images. The screen shot of the application is shown in Fig.8. Here, we load the images and determine the location of the eyes, nose and the head manually. In addition, we also annotate some regions to emphasize that these regions are not the facial components or the face. Using this approach, we could collect some positive and some negative data for refining our mathematical models.

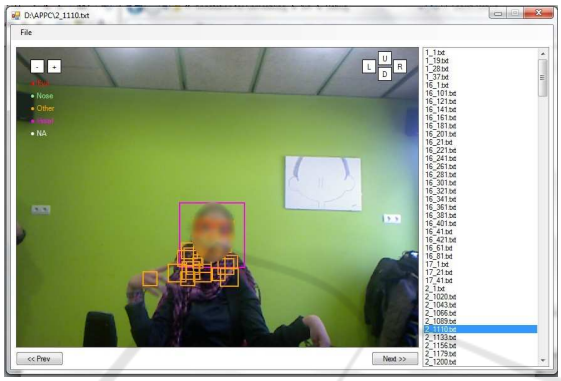

Fig. 8. Screen shot of the annotation application.

3.4 Head Analysis Framework

Analyzing the head is a multi-stage process. There are different methodologies for estimating pose of the head. The simplest method is template matching in which some templates are stored in the database and are compared with the input image to find the pose of the head. Clearly, this method is very limited since it is very sensitive to lighting and the persons identity. Another approach is to train different detectors from the images collected in different poses. Although this method is simple to implement, it has some serious problems. First, the few number of different poses which can be detected. Second, there are cases in which more than one detector returns a positive result.

Other methods for solving this problem are manifold embedding approaches. In these methods, the image of the face is mapped into a manifold with few dimensions. Then, the pose of the head is estimated on this manifold. One of the most important problems is the way that new images are going to be mapped on the manifold. Inaccurate mapping causes inaccurate pose estimation. Nonlinear regression methods are also a considerable approach to map the input image into the precise orientation along each axis in the 3D Cartesian space. However, while the method is rational and admissible from a theoretical perspective, in real applications we need a rich database containing persons with varied races, genders and ages gathered from different poses in 3D space. Therefore, the collection of this information is rather difficult.

In order to find the pose of the head, we may also use geometrical configuration of facial feature points. In this method, facial parts such as eyes, nose and mouth are

accurately localized on the input image. Then, using the configuration of the parts it is possible to deduce the head's pose in 3D space. It is clear that this method requires an accurate facial part localization algorithm. Moreover, mapping geometrical configuration into continuous pose values is very difficult. In this project we have utilized pictorial structures for localizing the head as well as the facial parts in the input image. To achieve this goal, we developed the framework showed in Fig.9.

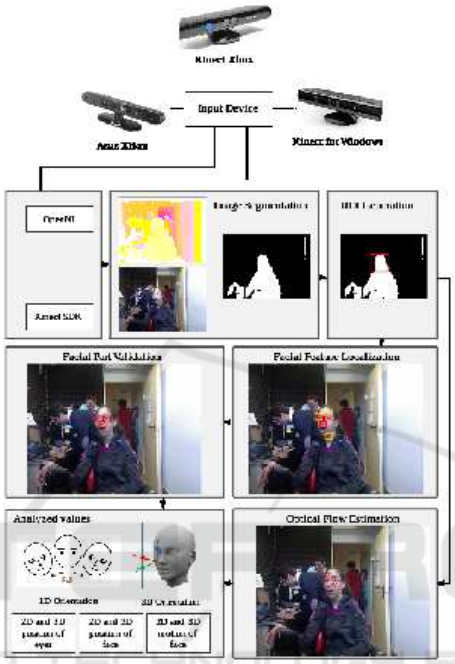

Fig. 9. Overall framework developed in this project.

Our algorithm starts by reading the images from the camera stream. We use both the depth and the RGB streams of the Kinect sensor. The raw image is preprocessed to reduce the noise and transform it into gray scale color space. Then, we use the depth information to segment out the image and generate a mask. This mask is used the find the region of interests (ROIs). The ROIs are processed to localize the face and then localize the facial components. Finally, the geometrical configuration of the components is validated and the head pose is estimated.

Camera Interface. Currently, there are different RGB-D cameras in the market. Among different brands, Kinect Xbox360, Kinect for Windows and Asus Xiton are three popular choices because of their reasonable price and accuracy.

In order to read the stream of data from these cameras, we have utilized OpenNI and Kinect SDK drivers. OpenNI is platform independent and can be used in different operating systems including Microsoft Windows and Linux. However, Kinect SDK is only compatible with Microsoft Windows.

For the sake of compatibility and portability we have developed an interface which

is able to automatically find the installed driver and open the data stream between the software and the camera. After the camera streams are successfully opened, the RGB and the depth image are captured iteratively to analyze the head motions.

Segmentation. It should be noted that using depth information helps us to have a more robust segmentation which is independent of identity, gender, age and race of the patient. Another segmentation technique is skin segmentation. Even though such method is useful for RGB images, the outcomes highly depend on the lighting, race, age and other physical factors. For this reason, when using skin segmentation the system will require a calibration step before starting the game in order to model the skin color more accurately, reducing the false-positive results. Notwithstanding, the depth information in RGB-D cameras is obtained using structured infrared lights, being user-independent. The only restriction is that RGB-D cameras can only work properly in indoor environments. Another advantage of depth information over skin segmentation is that depth segmentation is computationally efficient, which makes it suitable for real-time applications. In this project, any pixel out of an established operational range (i.e. $[80cm...170cm]$) is discarded from processing pipeline. The result of the depth segmentation stage is shown in Fig.9

ROI Generation. Segmented depth information is further processed to find the most probable location of the head. In this context, our definition of head is as follows: head of the patient has a roundish shape and it is located near the center. To be more specific, we extract the edges of the depth image taking into account the generated mask in the previous step. Then, we trace the edge pixels and localize the face by calculating the Chamfer distance between the edge pixels connected to the current pixel and our template model. Based on this definition, we first find the candidate regions and select the one which is closer to the camera.

Facial Component Detection. Two eyes and the tip of the nose make a triangle in 3D space which has a small degree-of-freedom due to inflexibility of the nose. In addition, while the state of the eye can be changed between open and close, the distance between two eyes is invariant. Therefore, the triangle obtained from two eyes and the tip of the nose can thought as a rigid object in space.

Having this assumption, if we can detect these three components in the image we can also find the pose of the head in 3D space. Moreover, the state of the eyes can be detected in parallel without any need to compute the feature vectors and scan the image again. Such efficient detections are very relevant due to the real-time requirements.

To detect the facial components, the local binary pattern (LBP) feature extraction method was used in this project. The reasons for using this method are its high discriminative power and fast calculation. There is a simple implementation of LBP in OpenCV. However, this implementation is not flexible and it is not good for real-time applications.

To have a more structured and well-implemented software, three basic LBP methods were implemented. These methods include LBP, Uniform LBP and Rotation Invariant Uniform LBP. In order to model the facial component we examined support vector

machines (SVM), Adaboost, Logitboost and RandomForest models. We found out that SVM is not a good choice in this project since the number of supports vector was high and for this reason the calculation speed was low. RandomForest had a good accuracy and it was faster than SVM but because the number of the trees and their depth was too high, hence, the computation speed was not high enough for our purpose. Adversely, Adaboost and Logitboost had very close performances but the number of the linear models in Adaboost was less than Logitboost. As the result, we used Adaboost as the facial component detector models in this project.

The reason for selecting these classifiers is the fact that using their score it is possible to calculate the level of confidence on the input feature vector. This measure is useful to discard false negative results. To deal with different sizes of the facial components, the detection algorithm is performed in a multi-scale iteration. The number of scales and the reduction coefficient is adjustable. After detection of the parts, their configuration is evaluated using simple geometry of the face.

Facial Part Validation. False-positive results are inevitable in real applications. We also had some false-positive results in the facial detection stage. To find out which component are true-positive we analyze the geometrical configuration between the three facial components and select the most probable configuration. Finally, the posture of the head is estimated using the depth information and the location of the facial component.

Head Pose Estimation. After detecting the facial components, their corresponding 3D information is calculated using the focal length of the Kinect and the depth information. It should be noted that, although there is a stream in the Kinect pipeline which calculates the position of every pixel in 3D coordinate system, this calculation is very CPU demanding and it affects the overall performance. For this reason, we only calculate the 3D coordinate of the left eye, the right eye and the nose (3 pixels in total).

Once the 3D coordinate of the points is estimated, the orientation of the head is calculated using a non-linear numerical optimization method. The 3D coordinate of the facial components are fitted on this model using an optimization method. Although the equation system is small and, hence, the total number of iterations is also small, the algorithm is still accurate.

Tracking. Tracking is an essential step in this project. While it provides very useful information for controlling the games, it can also be utilized to increase the robustness

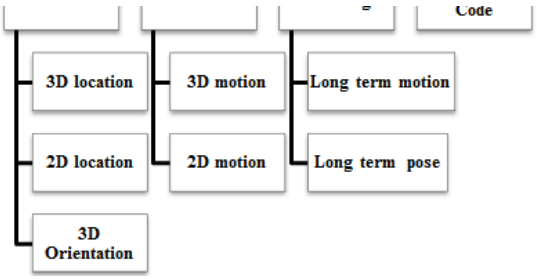

Fig. 10. Summary of the information produced by the URV and modified Kinect algorithms.

and computational efficiency of the image analysis algorithm. In this project, we implemented a linear tracking model since it is computationally efficient and it produces good results because the occlusion of the face rarely happens during the game play.

We store the tracking information including the 3D posture, 2D location and 3D location of the head as well as the 2D/3D optical flow history of the face. This information can be utilized to develop more advanced game control movements.

Head Optical Flow Estimation. In order to speed up the system and provide more information that can be used in controlling the game, we also compute the optical flow the face region. If the mean optical flow is less than some threshold, we use the analyze information from the previous frame as the results of the current frame. Otherwise, we analyze the image to estimate the posture. In addition, the optical flow information can be used to control the game. For example, it can be used to do the left, right, up and down movements.

4 Hands Analysis

This section deals with the extraction of information from the video sequences related to the position of hands in video streams. Despite hands are our main interest, the procedures described below do not limit their scope only to hands and are also capable of producing data of other parts of the body, such arms or head.

The outline of this procedure consists in detecting areas in the image likely to contain a hand based on color information and motion. We also use depth information in case it is available from a RGB-D camera (Kinect-style camera). In the following subsections we describe the methods employed.

Fig. 11. Some color transformations: (a) original image, (b) HSV transformation, and (c) YCrCb transformation.

4.1 Color Analysis

Despite a great amount of research was done so far to find effective methods to detect hands in a similar way as faces or objects, hands still suppose a great challenge due to their variable shape and finger configuration. This fact, together with the speed of their movement and the small size in which they are usually viewed in conventional video sequences, makes hand detection and tracking with low-cost color cameras a hard problem. With the irruption of affordable depth cameras, such as Kinect, the analysis of hands has certainly increased its precision and robustness. Nevertheless, this project made use of conventional color cameras.

We handle the task of determining the movement of hands by first finding image regions that are likely to contain a hand based on their appearance and, by imposing some constraints on the candidate regions, reducing to a small number of very plausible candidates, which finally will be considered hands. The main feature to define such regions is color and its similarity to *skin color*.

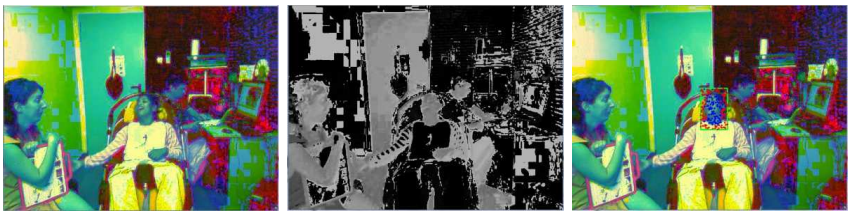

Fig. 12. Resulting H channel image (b) generated from a HSL image (a) after removing from the foreground those pixels with extreme values in the L channel. Image (c) shows the resulting pixels from an automatic sampling within the green rectangle where blue dots represent foreground samples and red dots are background.

Color Transformations. Color in images is usually coded by cameras as a series of RGB values. This straightforward color codification usually is not adequate to deal with changes provoked by illumination. Moreover, single color objects, such as hands, do not usually generate a simple region in the RGB color space but one which may be difficult to model instead. In order to reduce the complexity of the skin color region and increase its compactness and invariability to illumination we tested several color transformations Lab, Luv, HSV, HLS, and YCrCb as shown in Fig.11.

The best color transformations for our purposes are *Perceptual* color spaces HSV or HLS. One of the advantages of these color spaces in skin detection is that they allow to intuitively specify the boundary of the skin color class in terms of the hue and saturation. As I, V or L give the brightness information, they are often dropped to reduce illumination dependency of skin color. We use them to remove pixels that are either too bright or too dark. The resulting image (Fig.12) is a gray-scaled image where hands and other skin-colored parts of the body are quite distinguishable. These images will be later used to automatically learn subject-dependent skin color models.

Fig. 13. Binarization obtained using different procedures: (a) Thr+EM, (b) Thr+Knn, (c) Thr+SVM. The data gathered by the previous methods is used later in the learning steps of classification methods like Support Vector Machines (SVM), Expectation-Maximization (EM), K-Nearest Neighbors (KNN) that will be used in the following subsection to segment images based on skin color.

Color Segmentation. Several methods were developed to segment the skin color content in video sequences. All the methods described hereafter proceed in the same way, that is, first an automatic sample of pixels is generated separating them into two categories, i.e., skin and background, according to a coarse separation of colors in preestablished intervals. This sample is used for training a skin color model using a machine learning (ML) algorithm. Afterwards, a label is obtained for each pixel according to the model learnt resulting in a binarized image of the scene. Several approaches to compute the binarized image were attempted not only by varying the ML method, but also modifying the process by which image pixels are labeled.

We employed the following ML methods to learn the skin-color models:

Support Vector Machines (SVM): it is originally a technique for building optimal 2-class classifiers which was later extended to regressions and clustering problems. SVM is a partial case of kernel-based methods which maps feature vectors into a higher-dimensional space using a kernel and builds an optimal linear discriminant function in this space and an optimal hyper-plane that fits into the training data.

Expectation-Maximization (EM): it is an algorithm that estimates the parameters of a multivariate probability density function in the form of a Gaussian Mixture distribution with a specific number of mixtures. One of the main problems of the EM algorithm is a large number of parameters to estimate. We employ a small number of mixtures and diagonal covariance matrices to speed up the learning process.

K-Nearest Neighbors (KNN): the algorithm caches all training samples and predicts the response for the new sample by analyzing a certain number (K) of the nearest neighbors of the samples using voting, calculating weighted sum, and so on. It might be seen as learning by example because predictions are obtained by looking for the feature vector with a known response that is closest to the given vector.

Once a certain model is learned, skin pixels in an image are segmented straightforward by computing the class to which the belong. This procedure can be quite time-consuming and depends on the size of the image and the complexity of the class retrieval stage for each of the above methods, which is different and can vary greatly.

Raster Processing: According to our experimentation, in the case of EM processing in a raster manner is not adequate since it takes more than 1 second per frame. On the other hand, SVM and KNN are far faster and they take around a tenth of a second per frame.

Color Back-projection: To produce faster segmentations our approach consists in computing intervals in the color space and back-projecting their corresponding label onto the image, which speed up the process of pixel labeling. Two routines were implemented:

- Thresholding (Thr): We compute a set of interval limits for the foreground class (skin) in the gray scale range image that represents the color component used afterward as thresholds to segment the image. This approach is remarkably faster and provides fairly similar segmentations to those obtained by pixel-by-pixel labeling, preventing a performance bottleneck.
- Look-Up-Table (LUT): To cope with color regions which do not define a connected interval, a Look-Up Table (LUT) is used to convert colors in the image

into labels. This approach also copes with multi-channel color images such raw RGB or HSV. This approach is very efficiently in the case color is converted into a single-channel image like hue (H) component.

Fig. 14. Skin color segmentation results. Top left: original image after depth segmentation. Top right: binarization result. Bottom left: resulting image of connected component analysis. Bottom right: result image showing binarization on the original image

4.2 Motion Analysis

Users control games by waving hands in front of a camera. Despite motion alone is not enough to detect the position of hands -they can be stopped for a period of time–, motion still is a great source of information required to control games when combined with appearance cues such as color. In the case of lacking depth information, motion is a good alternative to locate the position of the user.

We attempted a series of algorithms for detecting motion in images. These algorithms took advantage of the fact that the camera is static and there are no dramatic changes either in the background or in the illumination. These algorithms first detect pixels in the image that suffered a change and use them to build a historic record of the areas in the image that presented motion. The history allows the segmentation of the regions in the image that for a certain period of time showed a moving part. The historic record can be kept intact or forgotten after a while, so areas with no motion are incorporated to the background. By varying the persistence we can select between long and short range movements.

Motion Detection. Hereafter we briefly review motion detection and history creation procedures:

Frame Differentiation (FD): The most basic algorithm to detect changes in a video stream consists in computing the difference. The nature of the video sequences allows obtaining good motion detection since the background and the camera are static. Another advantage is that it is no computational burden. Main drawback is the detection of spurious motion due to light flickering.

Background Subtraction (BGS): Based on the creation of a model representation of the background updated through time to incorporate small variations provoked by illumination and appearance/disappearance of objects in the background. We tested the family of algorithms called *Mixture of Gaussians* (MOG), which build a Gaussian Mixture Model (GMM) to represent the background. MOG is a very fast method that also performs shadow detection and incorporates a number of Gaussian components adapted pixel-wise. It recursively uses unsupervised learning for computing a finite mixture of models.

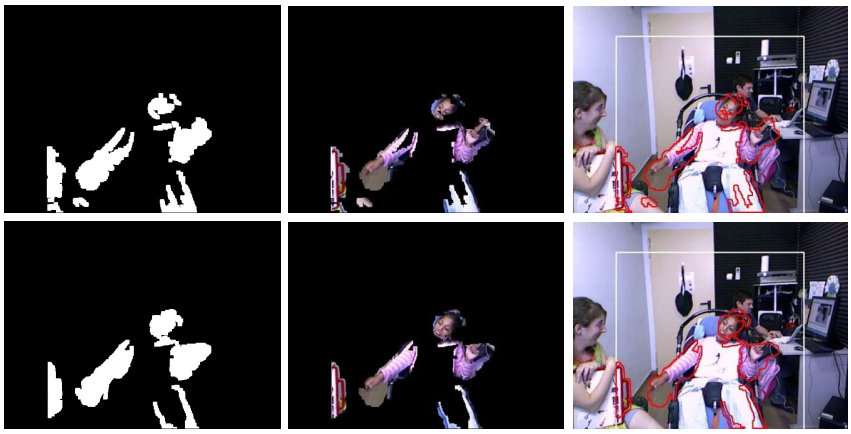

Fig. 15. Motion detection: frame differentiation (top row). Mixture of Gaussians MOG (bottom row).

Combining Cues. Previous methods generate segmented images using color and movement as information cues. Each method provides an independent segmentation that has some potentially useful information about the behavior of the user to control games. The question is how these different cues can combined into making the whole process more reliable and robust. We proceed by using motion segmentation as a way to anchor the region where the user is located. Color and movement segmentations are then combined in order to be more discriminative about the localization and tracking of body parts (head and hands).

Some precautions must be taken since due to the nature of the patients it is possible that the movement of these parts is not as fluid as in other game users. Their hands sometime move quite slowly according to their level of motor impairment. As a consequence, hard discarding of image regions might end up losing some important regions, like a hand that lags too much in a position. Therefore, we have combined color and movement taking into account the *verisimilitude* of each cue, i.e., how much we can trust color or motion per pixel based on a likelihood function.

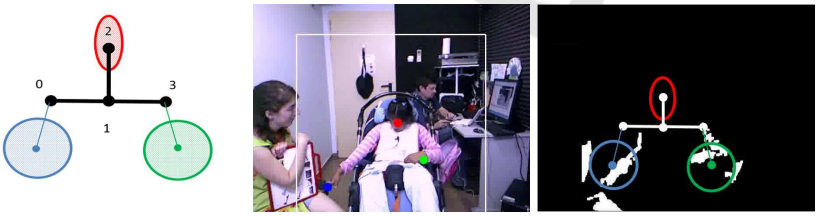

Fig. 16. Hand location and tracking. These images represent a sequence where detected regions are corresponded to head (red), left (blue) and right (red) hands.

Hand Discrimination. The final part of the process is to determine the identity of each region detected in the video sequences. The goal is to detect and to track hands to control the games, so these regions must be sorted out from the list of potential candidates obtained in the previous segmentation and connected component analysis. We

employed a heuristic method to speed up computations that imposes a given structure of the body to fit the set of candidate regions found in the image.

First approach is based on imposing a given simplified skeleton structure of the body where a set of candidate regions are fitted and anchored. Since the head has already been found by a face detection algorithm, this structure is composed of four points -0: left shoulder, 1: center point between left and right shoulders, 2: face midpoint, and 3: right shoulder-, and is updated at each step (see Fig.16). The skeleton separates candidate regions in groups (upper, left, right) according to geometrical restrictions. These regions are later identified into right and left hands. The heuristics implemented basically associate candidate regions inside circular influence areas near shoulder points and selects the most representative to be the final result.

The algorithm to select a body part region from a candidate list of works as follows. Each region is approximated both by an ellipse and a convex hull, in addition to extract its contour. This data is used to compute the extreme points for each region. The list of such extreme points are divided into left/right groups. For each pair of extreme points, a bounding box is computed and tested to be a body part. From the winning pair of points, depending on the body part (head, right/left hands) one representative point is selected. The movement of this point is later smoothed using a Kalman filter.

Moreover, the heuristic was improved employing tracking results from previous frames. By limiting the likely position using filtering techniques, the amount of false positives has been greatly reduced and the candidates that finally are selected are likelier to be correct one.

Additionally, since the capacity of movement of the patients is limited and it is pretty frequent they stay still for some period of time, we have locked the position of the body parts to the last known detected position, which is only updated when a sufficient important amount of movement is measured in the image.

As a result of the tests carried out at the APPC with patients and some suggestions addressed by the caregivers there, the final version of the hand tracking system is a version of the previous algorithms that only follows one single region. The main reason is that it is easier for the players only focus their attention in moving a single body part. Consequently, it is not necessary to follow several parts of the body at the same time.

The heuristic to obtain the movement of a single body part consists in defining a region in the calibration stage and extracting the motion of the most important group of connected components that define a moving region. This reduces substantially the computational burden and increases the robustness of the algorithms, which facilitates the use of the games by the patients, benefiting the playing experience.

Hand Tracking. For tracking the hands movements, that is, following their motion through time filtering out the errors in the localization, a *Kalman* filter with a *constant acceleration* dynamical model is employed to predict the movement of the tracked coordinates. In this model the state transition matrix controlling the dynamics of the moving element is defined as an identity matrix with another block formed by another identity matrix half the size of the first one and located in the upper-right side of the matrix.

Kalman filters are used to stabilize and track all the candidate regions found in the image, described as rectangles containing a body part (hands) or points (head). Different levels of filtering were defined corresponding to three types of velocities for the

Fig. 17. Hand location and tracking. These images represent a sequence where detected regions are corresponded to head (red), left (blue) and right (red) hands.

moving elements (slow, medium and high). In the case of tracking points, the number of dimensions of the filter state and measurements are 4 and 2, respectively. New measurements (x, y) are fed into the filter to update the filter state and the state is composed of potion and velocity coordinates (x, y, v_x, v_y). For tracking rectangles, the measurement is performed on the coordinates that define the rectangle, (x, y, w, h), where the point (x, y) corresponds to the upper left corner of the rectangle and (w, h) are width and height dimensions. The state vector is defined with this coordinates and the corresponding velocities, that is, $(x, y, w, h, v_x, v_y, v_w, v_h)$.

5 Conclusions

In this article we introduced the FP7 European project GAME-ABLING aiming to develop a platform for creating games by non-expert users for people with Cerebral Palsy (CP) disorder. The ultimate goal of the project is to increase the quality of life and motor movements of CP patients through various games. The games are played by different devices such Nintendo Wii balance board, Wiimote, keyboard, mouse, joystick and, more importantly, cameras. The idea behind using cameras is to control the games using hand and head movements and, consequently, to increasingly and unawares engage the patients in physical activities. CP patients have some specific characteristics that make difficult to use straightforwardly available game controls. Due to motor and cognitive restrictions, CP patients have few control on their body movements and tend to cover their face during the games. Taking this conditions into account, we developed two different approaches for analyzing hand movements and head posture. The overall outcome of the project has been positive, with all the goals achieved. Specifically, CP users can effectively control the games that are created by their caregivers. Also the acceptation of the users that have played with this platform has been positive. Nevertheless, the creation of tools for the inclusion of groups with severe motion and cognitive restrictions is still a work in progress that requires more attention by the research community.

Fig. 18. GAME-ABLING logo and members of the consortium.

Acknowledgements. This project was funded by the EU FP7 Program within the gran agreement for *Research for the Benefit of Specific Groups* with grant number 315032. We would also like to thank the work and help provided during the execution of this project by the staff at the *Associació Provincial de Paràlisi Cerebral*(APPC) in Tarragona, Spain.

References

1. Erik Murphy-Chutorian, Mohan Manubhai Trivedi, ”Head Pose Estimation in Computer Vision: A Survey”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, No.4, 2009
2. Kevin Bailly, Maurice Milgram, ”Head Pose Determination Using Synthetic Images”, 10th International Conference on Advanced Concepts for Intelligent Vision Systems, Springer, 2008
3. Vineeth Nallure Balasubramanian, Jieping Ye, ”Biased Manifold Embedding: A Framework for Person-Independent Head Pose Estimation”, Conference on Computer Vision and Pattern Recognition, IEEE, 2007
4. Bisser Raytchev, Ikushi Yoda, Katsuhiko Sakaue, ”Head Pose Estimation by Nonlinear Manifold Learning ”, 17th International Conference on Pattern Recognition, IEEE, 2004
5. Mikhail Belkin, Partha Niyogi, ”Laplacian Eigenmaps for Dimensionality Reduction and Data Representation”, Journal of Neural Computation, vol.15, 2003
6. Joshua B. Tenenbaum, Vin de Silva, John C. Langford, ”A Global Geometric Framework for Nonlinear Dimensionality Reduction”, Journal of Science, vol.290, No.5500, 2000
7. Matthias Straka, Martin Urschler, Markus Storer, Horst Bischof, Josef A. Birchbauer, ”Person Independent Head Pose Estimation by Non-Linear Regression and Manifold Embedding”, 34th Workshop of the Austrian Association for Pattern Recognition, 2010
8. Erik Murphy-Chutorian, Mohan Manubhai Trivedi, ”Head Pose Estimation and Augmented Reality Tracking: An Integrated System and Evaluation for Monitoring Driver Awareness”, IEEE Transactions on Intelligent Transportation Systems, vol.11, No.2, 2010
9. Kevin Bailly, Maurice Milgram, Philippe Phothisane, ”Head Pose Estimation by a Stepwise Nonlinear Regression”, 13th International Conference on Computer Analysis of Images and Patterns, Springer, 2009
10. Teodora Vatahska , Maren Bennewitz , Sven Behnke, ”Feature- based Head Pose Estimation from Images”, 7th IEEE-RAS International Conference on Humanoid Robots, IEEE, 2007
11. Pedro Martins, Jorge Batista, ”Accurate Single View Model-Based Head Pose Estimation”, 8th IEEE International Conference on Automatic Face and Gesture Recognition, IEEE, 2008

12. Pedro Martins, Jorge Batista, ”Monocular Head Pose Estimation”, 5th international conference on Image Analysis and Recognition, Springer, 2008
13. Jrgen Ahlberg, ”Candide3 An Updated Parameterized Face”, Deptartment of Electrical Engineering, Linkping University, 2001
14. Viola P., Jones M., ”Rapid Object Detection Using a Boosted Cascade of Simple Features”, Conference on Computer Vision and Pattern Recognition, IEEE, 2001
15. Danijela Vukadinovic, Maja Pantic, ”Fully Automatic Facial Feature Point Detection Using Gabor Feature Based Boosted Classifiers”, International Conference on Systems, Man and Cybernetics, IEEE, 2005
16. Guoying Zhao, Matti Pietikainen, ”Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007
17. T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham, ”Active shape modelstheir training and application”, Journal on Computer Vision and Image Understanding, Elsevier Science, 1995
18. Marc CHAUMONT, Brice BEAUMESNIL, ”Robust and Real-Time 3D-Face Model Extraction”, International Conference on Image Processing, IEEE, 2005
19. Daniel F DeMenthon, Larry S Davis, ”Model Based Object Pose in 25 Lines of Code”, International Journal of Computer Vision, vol.15, 1995
20. La Cascia, M., and Sclaroff, S., ”Fast, Reliable Head Tracking under Varying Illumination: An Approach Based on Robust Registration of Texture-Mapped 3D Models”, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 22, No. 4, 2000,
21. M.F. Valstar, M. Pantic, Induced Disgust, Happiness and Surprise: an Addition to the MMI Facial Expression Database, Proceedings of the International Language Resources and Evaluation Conference, Malta, May 2010
22. M. Pantic, M.F. Valstar, R. Rademaker and L. Maat, Web-based database for facial expression analysis, Proc. IEEE Int’l Conf. on Multimedia and Expo (ICME’05), Amsterdam, The Netherlands, July 2005
23. Chang Chih-Chung, Lin Chih-Jen, ”A library for support vector machines”, ACM Transactions on Intelligent Systems and Technology, vol.2, issue 3, 2011
24. Sukwon Choi, Daijin Kim, ”Robust head tracking using 3D ellipsoidal head model in particle filter”, Pattern Recognition, vo.41, issue 9, Elsevier, 2008