Handicapped People Virtual Keyboard Controlled by Head Motion Detection

Ondrej Krejcar
Department of Information Technologies,
Faculty of Informatics and Management,
University of Hradec Kralove,
Hradec Kralove, Czech Republic
ondrej.krejcar@remoteworld.net

Abstract – Nowadays, computers and modern technologies are available to almost all people. However, we should not forget about disabled or handicapped customers. Those people should have access to a PC as well as to the Internet. This project implements the boundary for people with movement disability. Although, the basic assumption is that they can at least move their heads and eyelids. These parts of the body are monitored by web cameras and with their help the user moves around the virtual keyboard shown on the screen. In this way the user is able to utilize their own PC. Moreover, through winking an eye he/she can virtually press a button and write the required symbol. The whole software for detection system was established in a development environment of the Visual Studio 2010 from Microsoft company. The main advantage of our solution is a cheaper price in compare to any existing solutions. By this fact the solution is possible to use even in third countries like in Africa continent.

Keywords – HCI; face detection; users‘ interface; handicapped people; Open CV libraries

I. Introduction

As mentioned in the abstract, the developed software is aimed at operating a PC using movements of the head and eyelids. In order to detect head motion using web camera, it is necessary to implement face detection of a particular user and to determine the direction of its movement. The face detection is basic and important technology from the range of scanning systems which are used as a boundary between a PC and a person [1]. A large number of authors have already dealt with such problems. They have published many books and developed considerable numbers of algorithms and methods for both object detection in production processes and face detection. In recent years, face detection algorithms gave rise to extremely precise methods for face localization in images [2].

The main issue of each face detection system is a camera which contains a CCD (Charge-Coupled Device) sensor. With ample speed this sensor generates images of space in front of the camera. These images are then saved by the camera onto a hard disk in video file format. It is mostly the format of AVI (Audio Video Interleave) [3]. The detection system then searches according to the chosen algorithm for a face in the images of the video, eventually in its segments and identifies their position based on the number of pixels from the left margin and the number of pixels from the upper margin. The programmed software's objective is to contrast the offset of detected parts in comparison with former images. According to the offset of parts, it enables the movement on a virtual keyboard. The motion of eyelids is then analyzed as the pressing of a certain button on the keyboard. A person usually blinks with both eyes at once, which is the reason why the program cannot respond to this movement.

Therefore it is clear that two components have the most significant impact on the system's quality. These are the apparatuses for face scanning of a user and the algorithm for face detection from a scanned image. It is very important to know the value of resolution in a web camera. If using a camera with small resolution, the user's motion does not have to be detected with required certainty. On the other hand, if using a high-quality camera with high resolution and speed of image scanning, it might take longer for older computers to implement detection algorithm and face location.

Perhaps the most complicated issue of detection system concerns choosing the correct algorithm for face detection. First of all, it is necessary to consider the uncountable number of variations in human appearance. People differ from one another not only through the shapes and sizes of individual body parts, but also through skin pigmentation, eye color and hair and eyebrow lengths. A woman has an entirely differently shaped face than an unshaved man etc. Another disturbing impact might be caused by a movement behind a user, i.e. a sudden change of light intensity in a monitored scene, or the entry of another person to the scanned zone. All of these and many other obstacles are supposed to be eliminated by this system in order to accomplish the highest levels of quality and reliability. On the other hand, there are also some advantages such as the homogeneity of faces in comparison with other categories of detected objects. All of these share a certain canonical structure, including two eyes, a nose and a mouth. These are symmetrically placed with regard to longitudinal axis of oval contour (although not all of the components are traceable from all visual angles) [4].

It is suitable to use face static components for detection of head movement. These components represent the motion of a whole face (i.e. nose, birthmarks).

II. Problem Definition

In recent times, modern technologies and computers became essential parts of our lives. They are present in our homes, schools, offices etc. Access to the internet and working on computers became part of a routine for ordinary people. However, how do you access the internet or send emails to friends living on different continents, when you cannot move your hands and fingers? Fortunately, a large number of experts started to be concerned about this issue and developed many alternative boundaries for the operating of computers even without a mouse and a keyboard.

The first very interesting system was proposed and designed in the Czech Republic. It is called the i4control. The system i4control® is the new type of computer periphery which enables disabled immobile users to operate personal computers by eye motion (eventually by head movement). This system substitutes an ordinary mouse and offers to its user a simple way to communicate with installed applications by eye movement [Fig. 1] [7].

Figure 1. i4control User Interface.

Another product which uses a similar system is called LifeTool IntegraMouse. Basically it is a mouse controlled by lips' motion and the pressing of a button is accomplished by sipping or blowing. This system is ideal for users with various physical problems such as whole body paralysis, after amputation, muscular dystrophy or disseminated sclerosis [Fig. 2] [8].

Figure 2. IntegraMouse User Interface.

The system My Voice is another very interesting option. This program enables users to operate a computer and install programs exclusively by vocal commands. Using these commands, users are able to carry out the same actions as would be otherwise realized by mouse and keyboard. The program is capable of immediately recognizing vocal commands from random person and therefore, it is not necessary for the user to narrate anything before usage.

Figure 3. SmartNav User Interface.

Finally, there is a product SmartNav from NaturalPoint company. SmartNav uses infrared (IR) camera for monitoring user's head motions [Fig. 3] [10]. The system is based on scanning the user's head using infrared LED. Special reflex facets are placed on the user's head (glasses) and then the radiation is reflected back to a camera which analyses its motion.

All described solutions are very expensive [Table 1]. This fact was one of motivation to develop new solution as a low cost.

TABLE I
Existing User Interface Solutions for Handicapped users

Solution name	Price [USD]	Principle
i4control	2200	Eyes movement
IntegraMouse	3400	Lips motion
My Voice	650	Voice recognition
SmartNav	400-500	Head motion
Our Solution	free	Head and eyes motion

As mentioned in the introduction, in this project a virtual camera that scans user's head motion was used in order to enable the user to move over a virtual keyboard. However, this solution method gives rise to certain problems mainly during head detection and its individual parts. Another person might approach the visual field of the camera which might lead to a fault in a detection algorithm. Moreover, it is necessary to consider the distance in which the user's face will be scanned. In order for the system to be cheap and available to everyone, it is also necessary that the system is not composed of parts whose purchase price is equal to hundreds of USD. This is an example of an HD camera. Therefore it is necessary to use ordinary web cameras that are found in most of the notebooks.

These cameras usually have a resolution of around 1.3 megapixels. This means that the scanned image has proportions of 1280x1024 pixels. The image is often programmatically adjusted into a resolution of 640x480 pixels in the case of video scanning in order to fasten the transfer of information on the internet. In this case it is necessary to consider another restricting factor – the distance of a user from camera. An ordinary person is capable to move his/her head in the range of 20 cm. If the person is in distance of 60 cm from the camera, the head movements are going to be scanned in a segment of 150x150 pixels. These values are just approximate, because there are no two web cameras from various producers that would be the same. Furthermore, every one of them has a different defined angle in which its lens is able to scan an image and carry it on CCD sensor. The recognition of head movements in a range of 140x140 pixels is very impractical for mouse movement on a working surface with a resolution of 1280x1024 pixels. A mouse pointer would be likely to move by jumps over the screen which would in turn be caused by slight unwanted head movements of a user.

The mouse movement would be then possible to realize using virtual touchpad with the function which is offered by every notebook. An especially interesting solution might be designed by using accelerometer to scan head motion. However, that is not the objective of this project.

The alternative ways of computer operating which are described in the first part of this chapter are very sophisticating. However, every one of these solutions brings certain disadvantages. Firstly, for using the i4control system a user needs to wear glasses with a camera that is placed a couple of centimeters away from the user's eyes. This might be a very disturbing factor for some people. Moreover, a data cable is needed for connection of the glasses to the computer. This cable might also get attached to some object causing the glasses to fall. A similar situation may happen with IntegraMouse. Another system, My Voice, can be difficult for usage outside of the home because of disturbing factors such as television or radio in an office or other public place. The last system, SmartNav, which is similar to the solution proposed in this project, might be slow for the user because of the necessity of special head movement in order to press a button. Therefore a great emphasis in this project was put on the utilization of all systems' described advantages and their interconnection. This leads to highest possible level of comfort for the user.

III. NEW SOLUTION

This chapter might be imaginarily divided into two parts. In the first part a necessary theory about programmed detection of face and other parts is given, including information about basic functions of developed programs. The latter part of the chapter introduces the structure of the program itself with description of its individual parts and linkage.

A. Face Detection

A detection of a face and both eyes is used for keyboard operating. The face movement represents the movement across the keyboard. Moreover, a reasonably long wink using both eyes gives the system order for pressing a button. Open CV library from Intel company is used in this project in order to launch the camera and detect individual parts. Open CV is a free and open multiplatform library designed for manipulation with image. It is aimed mainly at computer vision and real-time image processing [11]. The CV is an abbreviation which stands for Computer Vision. However, it only works in C/C++ setting which constitutes a disadvantage. Therefore, this project also uses Emgu CV system [12] which is a cross platform for .NET that works as a cover for Open CV libraries indented for image processing.

Moreover, it enables CV libraries to use a function for languages compatible with .NET such as C#, VB, VC++, IronPython etc. [12]. Emgu CV then allows in just few commands image scan by web camera and detection of individual objects on created image. In order to obtain the highest possible speed of the whole application the scanned image is converted into black and white color. The detection itself is then undertaken by DetectHaarCascade command which is based on a high-speed detection algorithm called Viola & Jones (it is the first boundary for detection of objects that carries out real-time detection; it was created in 2001 by Paul Viola and Michael Jones [13]) and another algorithm called Adaboost. Adaboost constitutes an abbreviation for Adaptive Boosting. It is an algorithm of machine learning established by Yoav Freundem a Robert Schapirem. Moreover, it is a meta-algorithm which might be used in conjunction with other learning algorithms in order to improve their performance [14].

The very first argument of a command DetectHaarCascade is so-called Haarcascade (cascade haar style) which is a file of .XML type saved in the same folder as the used application. The detected object is parameterized in this file.

The detection algorithm returns a face variable of VAR data type. Then it is necessary to search only for the wanted face using a command foreach and transfer it to a variable of rectangle type – in this way a rectangle which is represented by coordinates of upper right corner and the size of its sides is acquired. The testing of detection algorithm uncovered that any insignificant zooming in or out of the user's face from the camera would cause unstable changes of the rectangle's size (or rather square, because both width and height are consistent). For this reason, the upper left corner coordinates of the square are inapplicable and the middle of this square was used for the movement across a keyboard. The middle coordinates do not considerably change regarding to different sizes of human faces. These coordinates might be calculated using the following formula:

$$X_{Middle} = X_{Corner} + \frac{W}{2}$$; where W is the width of a square

According to a similar formula it is even possible to calculate coordinates Y. In this way those coordinates which are necessary for movement across the virtual keyboard are obtained. It was discovered during another testing of the detection algorithm that the best ratio between distance and sensitivity of a control is within the 211x211 or 295x295 pixels of detected square. (The size of square changes by jumps.) The distance of user's face from the camera in these circumstances was maximally 65cm. If this distance becomes greater, it would not be possible to operate the keyboard just by slight turn of a head, because the face detection itself would fail due to the face being turned too much. Moreover, it would lead to a failure of detection of eyes. In the case when the detected face had size of 211x211 pixels, user's head could be turned by the range of 140x140 pixels.

B. Application Structure

The proposed virtual keyboard [Fig. 4] has 7 rows and maximally 14 buttons in one single line. Therefore, the boundary was invented in a way to move to the next button if there is a change by 10 pixels upward or in width. In this way slight head motions and failures of detection algorithm are eliminated.

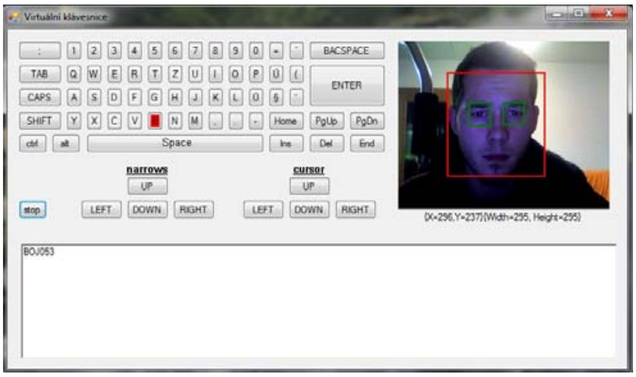

Figure 4. User interface of developed application.

The pressing of a button might be represented in two ways - eye winking or special mouth movement (i.e. opening of a mouth). The first way was chosen for this project, because mouth movement was discovered to cause frequent failures of the system. The program analyzed a beard as a part of a mouth of an experimental subject. In order to remove this problem it would be necessary to modify or create absolutely new cascade haar style.

Moreover, the detection would have to be done in a colorful image which would significantly increase the time necessary for accomplishment of detected algorithm and reduce the speed of developed application. It was much more convenient to use the scanning of eye motion and, in order to press a button, to close both eyes for a reasonable period of time.

This solution is also well-founded because during the turning of the head, the eyes, especially in limit value, might be covered by the nose. The system is then capable of detecting only one of the eyes. However, at least one of the eyes should be almost always recognizable by algorithm on the acquired image from a web camera. If the algorithm is unable to find either of the eyes for 0.5 – 1.5 s and then detects at least one of them, it will evaluate it as the pressing of a button. In this way it is possible to eliminate shorter autonomous eye winking and also the cases in which eyes would not be detected for longer periods of time (i.e. if a user moves too far from the camera).

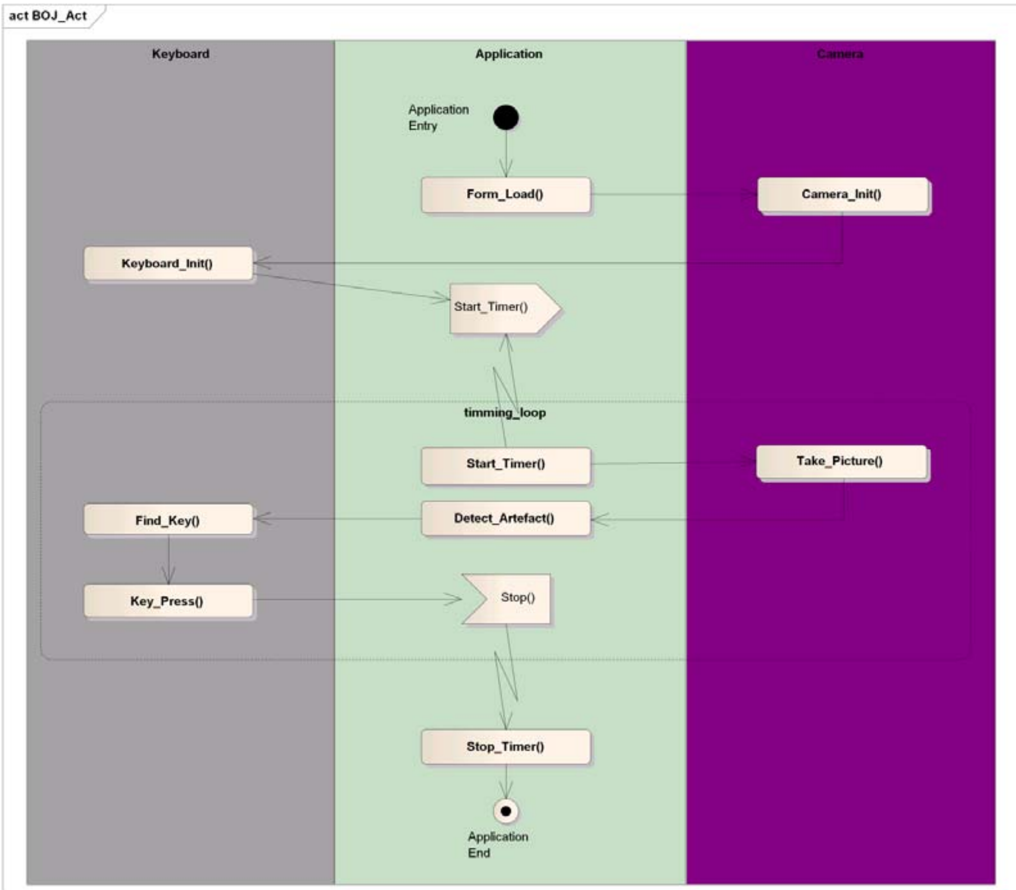

Figure 5. UML activity diagram of developed user interface for handicapped users.

The fundamental part of the application is formed by a main class – a keyboard - and by a supplementary class – a camera (next viz. Image 1). The class camera was rather used during the studies of parameter transfer by methods among different classes. Two functions for working with virtual keyboard are placed within the class keyboard. These are the methods ‘detect’ in which the detecting algorithm is implemented and ‘timer1_Tick’. The latter method is excessively important, because it calls other methods and assures that the whole application is running smoothly in a certain loop. Firstly, it calls the method ‘take_picture’, placed in the camera class, which returns images taken by a web camera. The method forms a mirror image of the previous one alongside an imaginary vertical axis. The turnover is necessary, because the image on the screen would not otherwise move in the same direction as the user.

Furthermore, the method timer1_Tick calls the method ‘detect’ whose entry parameter is an image taken by web camera. This method is able to find user’s face and eyes by using already mentioned libraries Emgu CV. The outcome is again the given image, which is then modified through having colorful rectangles drawn into it that border the already found objects (the face and eyes). Then the image is simply adjusted by timer1_Tick method to the size that corresponds with the actual size of the picture_box into which the image is subsequently drawn as a bit map. However, as a result the ‘detect’ method carries out more tasks. At the beginning it is necessary to calibrate the head position and the middle of the keyboard. Therefore the ‘detect’ method is used right at the start to draw a black square in the middle of the image. Consequently, the user has to place his face into this square. If this is done correctly, the ‘detect’ function overwrites the variable value ini_pom to 1 and the condition in the ‘detect’ function which draws this black square stops working. Conversely, the condition which initiates the ‘find_button’ method takes place. Moreover, it changes the particular button on a keyboard to a red color and draws a red square around the user’s head. The initiated method ‘find_button’ has the middle of a user’s face as its input parameter and a button which corresponds to the particular face positioning is its output. An algorithm for searching in a two dimensional field for keys called Button was implemented into this method. Those keys are situated due to the position of the middle of a face in accordance to the middle of a scanned image and also to the number of keys in a particular keyboard row. Furthermore, certain insensitivity for position changes of less than 10 pixels was implemented in order to eliminate minor head motions and detection algorithm’s faults (as mentioned before). The function of algorithm and of the whole method is very hard to describe. However, it is perfectly understandable from a written code.

The matrix of Button keys which was mentioned before was implemented at the beginning using the ‘initialization_key’ method that is called in ‘key_load’ method. Only two keys from the matrix, Enter and Space, occupy more spaces. This is due to the fact that they take in more columns or rows and the transfer to them must be possible even from those locations.

If the timer interval of 1 ms is used, the speed of the whole application should be dependent only on the speed of a camera and Viola & Jones detection algorithm.

IV. Implementation

This chapter is concerned about the practical implementation of the new solution.

The application was developed using the C# language of .NET Framework in Visual Studio 2010 environment from Microsoft Company. The whole graphic page of the application was invented in GUI which is a library of graphical user’s boundary. Apart from the usual elements as *Button*, *Picture box* or *Label*, another two components were used to run in the background. Those were *Timer* and *FolderBrowserDialog*.

FolderBrowserDialog is a graphic component which serves as an instrument for browsing folders in a computer and after choosing one of the folders it reveals the route towards it. This function was used in the initialization part of the program. The whole initialization runs in *Form1_Load* method. It is necessary within the program because the application works with two external XML files, in which the two above mentioned cascade styles for detection of face and eyes are saved. Those files are called *haarcascade_eye.XML* and *haarcascade_frontalface_alt2.XML*.

After the first initiation of the application, the program firstly determines whether the *ini_data.txt* file is located in the application’s directory. If this file is missing, the application creates it. Consequently, it initiates a system message for a user in which it passes on the information about necessary initialization of the program and the user is asked to insert the path towards files *.XML with cascade styles. The component *FolderBrowserDialog* is used for inserting this path.

After the next running of the application the program investigates again the presence of initializing file ini_data.txt. If this file is found its content is uploaded and if there are any files *.XML* with cascade styles detected in the content, a form with virtual keyboard is produced. If the initializing file *ini_data.txt* is detected, but it does not contain valid data, it is deleted and the user is informed using *MessageBox* that the application will be closed and it is necessary to reopen it (after a new start-up the whole situation will be running in the same way as during the first initiation of the program).

Figure 6. Part of the application user interface. User face did not recognized – the black rectangle is showed.

When the main window of the whole application is showed, user need to sit down in front of the web camera and position his face into the black square [Fig. 6]. When positioning user face into the middle of the image, the black square should become red and begin to move together with the detected face.

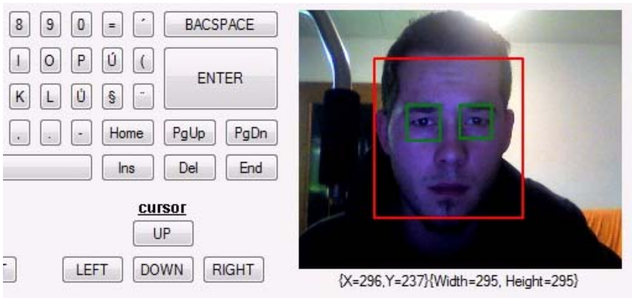

Figure 7. Part of the application user interface. User face is detected and recognized – the red rectangle is showed around green rectangles at eyes positions.

After successful detection of user face, the position and size of user face and also user eyes is being scanned. A red colored active key moves around the keyboard together with the face movement [Fig. 4].

The pressing of this key is done by closing of user eyes for the period of 0.5 – 1.5s. This time depends mainly on the speed of a used computer (processing of an image from the web camera, implementation of detecting algorithm) and also on the speed of the used web camera. The required letter is written into the text field under the keyboard.

The buttons ENTER, BACKSPACE and SPACE have special functions that were programmed in order to adjust the text as it is standard with any other keyboard. The button stop ends the application.

V. TESTING OF DEVELOPED APPLICATION

This chapter is concerned with testing of the developed application. The results from this testing should be compared to the results gained from the testing of other authors. However, these other authors' results were not founded and their solutions are built upon completely different principles. Therefore these projects could not be compared.

In order to somehow compare gained results there is an application which was developed in order to operate a personal computer using movement of face or eyes. This application was run and tested on two individual computers.

The first computer works with the processor AMD TURION X2 2GHz (Dual Core), 3GB RAM and it scans the user by internal web camera with 1.3 MP resolution. The second computer operates with processor AMD TURION X2 1.6Hz (Dual Core), the main board contains 2GB RAM and an external camera (of older manufacture date) with 0.3 MP resolution which is used for image scanning.

A. *Testing Methodology*

The testing was undertaken in variety of ways. The first is based on writing a predetermined text. The sentence which is necessary to be written is: 'APPLICATION TESTING'. While writing this sentence the time taken for the task to be completed, and the number of faults committed by the user, are recorded. In order to limit the influences of incidental faults (processor's working capacity, user's fault, etc.) this sentence is written down and its results are consequently 5x mapped. The arithmetic average is then calculated from the resultant values. The same procedure is applied even for the second computer. Finally, this type of testing is undertaken in two differently lighted environments.

The second style of testing is based on the number of writing of the same symbol in the beforehand set time. At the present time the number of faults which were made by the user at the given time, and the number of correct key presses, are recorded into tables. The whole testing is again consequently undertaken 5x and then the average of these values is generated in order to lower the influence of random faults and the user's faults during the testing. Similarly to the previous case the testing is undertaken in two environments with different light settings.

B. *Testing Results*

The results of testing using 1st technique are stated in the table [Table 2].

TABLE II
TEST RESULTS OF WRITING OF DEFINED TEXT

	Try	Better light		Worse light	
		Good	Errors	Good	Errors
HP	1	33	12	38	14
	2	40	6	41	12
	3	42	8	35	18
	4	38	5	30	15
	5	39	9	39	8
Average		38,40	8,00	36,60	13,40
FS	1	19	15	21	11
	2	18	16	28	13
	3	23	14	27	19
	4	26	14	24	12
	5	17	21	31	18
Average		20,60	16,00	26,20	14,60

The abbreviation HP identifies a notebook from Hewlett Packard Company. The abbreviation FS describes a notebook from Fujitsu-Siemens company. The results from a testing by 2nd technique are stated in the table [Table 3].

TABLE III
TEST RESULTS OF WRITING PER DECLARED TIME

	Try	Better light		Worse light	
		Time	Errors	Time	Errors
HP	1	3:16	7	3:26	10
	2	3:07	8	3:04	8
	3	3:25	5	3:12	7
	4	2:51	6	2:36	12
	5	2:31	8	2:53	7
Average		3:02	6,80	3:02	8,80
FS	1	4:13	17	4:28	21
	2	4:25	16	5:03	27
	3	4:12	13	4:45	22
	4	3:55	11	4:11	18
	5	4:06	12	3:58	14
Average		4:10	13,80	4:29	20,40

C. *Results Discussion*

As it is visible from the table [Table 2], the best results were achieved by using the application of HP notebook when there were good light conditions. This result is not surprising. When there is a good light, the camera scans the image significantly faster, because it does not have to adjust to worse light conditions. The images given by the camera are also sharper than when taking a picture with limited lighting. During these conditions the error rate was raised while writing a text. When considering the tables for notebook FS it is obvious that the precise text writing was very difficult. The main problem was that eyes were scanned even when the eyelids were closed. It was necessary to keep closing the eyes in order to slightly shrivel the face and during this the head position also always changed. Therefore, the error rate in this case was much higher and the text writing took longer. One solution would be supplementing the program by adjusting the sensitivity with which the eyes are detected on the image. (This sensitivity is adjustable in an argument of method called *DetectHaarCascade*).

If we focus on the results recorded in the table [Table 3], it is obvious that the best results were again achieved by using notebook HP which is situated in a well lighted setting. During the testing it was also discovered that the sensitivity of the application is still too high. On the other hand it has to be noticed that attempts numbers 4 and 5 in tables are always slightly better than the ones before. This means that practice plays a considerable role when using the application and it is necessary for the user to become accustomed to the operation of the keyboard through face motion.

VI. Conclusions

Within this project we were able to successfully practise working with a web camera through the detection of different parts of the human body on a scanned image. After certain changes the developed application could be largely beneficial to handicapped people and it could simplify the way in which they use their personal computer. It would be necessary to solve two essential problems. The first problem is the communication of the virtual keyboard with other computer processes (Word, Explorer, etc.). The second problem is getting all the function keys to work (CAPS, SHIFT, etc.) and also the keys for operating the cursor. A virtual keyboard, which is possible to be operated through face motion, was established. The main advantage of this project is that the user does not have to buy an expensive hardware (if he has a camera integrated in a notebook). The only thing necessary is to install Emgu.cv software (which is free) and initiate the invented application.

Acknowledgements

This work was supported by „SMEW – Smart Environments at Workplaces“, Grant Agency of the Czech Republic, GACR P403/10/1310.

References

[1] Zhu, HJ., Koga, T.: Face detection based on AdaBoost algorithm with Local AutoCorrelation image, In IEICE ELECTRONICS EXPRESS, Volume 7, Issue 15, pp. 1125-1131, 2010, DOI: 10.1587/elex.7.1125

[2] Sznitman, R., Jedynak, B.: Active Testing for Face Detection and Localization, In IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, Volume 32, Issue 10, pp. 1914-1920, 2010, DOI: 10.1109/TPAMI.2010.106

[3] < http://cs.wikipedia.org > *http://cs.wikipedia.org/wiki/AVI* [Accessed 8.10.2010]

[4] Hershler, O., Golan, T., Bentin, S., Hochstein, S.: The wide window of face detection, In JOURNAL OF VISION, Volume 10, Issue 10, Article number 21, 2010, DOI: 10.1167/10.10.21

[5] < http://www.inb.uni-luebeck.de > *http://www.inb.uni-luebeck.de/publikationen/pdfs/BoHaRiMaBa09.pdf* [Accessed 8.10.2010]

[6] Peng, ZY., Ai, HZ., Hong, W., Liang, LH., Xu, GY.: Multi-cue-based face and facial feature detection on video segments, In JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY, Volume 18, Issue 2, pp. 241-246, 2003, ISSN: 1000-9000

[7] < http://www.i4control.cz/ > *http://www.i4control.cz/* [Accessed 19.10.2010]

[8] < http://www.lifetool.at/ > *http://www.lifetool.at/show_content.php?sid=218* [Accessed 19.10.2010]

[9] < http://www.fugasoft.cz > http://www.fugasoft.cz/index.php?cont=myvoice#o-programu [Accessed 19.10.2010]

[10] < http://www.naturalpoint.com > http://www.naturalpoint.com/smartnav/products/about.html [Accessed 19.10.2010]

[11] < http://cs.wikipedia.org > *http://cs.wikipedia.org/wiki/OpenCV* [Accessed 5.11.2010]

[12] < http://www.emgu.com > *http://www.emgu.com/wiki/index.php/Main_Page* [Accessed 5.11.2010]

[13] < http://en.wikipedia.org > *http://en.wikipedia.org/wiki/Viola-Jones_object_detection_framework* [Accessed 5.11.2010]

[14] < http://en.wikipedia.org > *http://en.wikipedia.org/wiki/AdaBoost* [Accessed 5.11.2010]