Extending System Capabilities with Multimodal Control

Gregor Rozinaj, Marek Vančo, Ivan Minárik, Ivan Drozd, Renata Rybárová

Institute of Telecommunications, Slovak University of Technology, Bratislava, Ilkovičova 3, 81219 Bratislava, Slovakia, gregor.rozinaj@stuba.sk, marek_vanco@stuba.sk, ivan.minarik@stuba.sk, ivan.drozd@stuba.sk, renata.rybarova@stuba.sk

Abstract: Multimodal interface (MMI) is the first layer from a user point of view to interact with most IT systems and applications. MMI offers natural and intuitive interface for user identification and system navigation. Typical features of multimodal control contain user identification based on face recognition and speaker voice recognition, system control based on voice commands and gesture recognition. Several examples show typical applications with MMI like voting or direct shopping while watching TV.

Keywords: multimodal interface; gesture control; gesture recognition; voice navigation

1 Introduction

At the turn of the 21st Century people were thrilled when they could get connected to the Internet on monochromatic displays of their newest flagship mobile phones. Similarly, the possibility to browse the teletext was considered a high level of interaction with the classic TVs. The past 15 years lead to a stunningly extensive progress of a human – computer interaction (HCI) in all possible kinds of electronic devices.

HCI means interaction between humans and machines via all possible input and output interfaces, i.e. keyboard, mouse, pen, gesture, speech, face, iris, etc. Multimodal interface represents all input and output interfaces based on human senses and inputs from a user, and so creates a natural way of communication and control for the user. The term modality refers to a human sense or a form of user input, for example face recognition is based on vision, speech recognition is based on speech, and gesture recognition is based on movement.

This article offers a proposal of multimodal interface focused on a system navigation. Section 2 includes a short introduction to the MMI. In Section 3 some

of the most common techniques and algorithms of voice and gesture recognition are described together with implemented algorithms for the gesture recognition in our system. Results for the tested algorithms are presented in this section, too. Section 4 contains the architecture of our proposed MMI application. At the end, in Section 5, the possible use of the proposed MMI is explained in various scenarios.

2 Multimodal Interface: Natural Entry to the System

Currently, the most widely used input devices for human–computer communication are keyboard, mouse, or touch tablet. These devices are far from the idea of natural communication with a computer, and rather represent human adaptation to computer limitations. In the last few years a requirement began to pop up that humans need to communicate with machines in the same way as they do with each other: by speech, mimics or gestures, since these forms conceive much more information than traditional peripheral devices are able to acquire. Our system focuses on this need by implementing and interconnecting several modalities to achieve a more natural control. This leads us to the term Multimodal interface [1].

The first step of communication with the Multimodal Interface (MMI) starts with user identification and authorization. Devices are aware of their legitimate users continuously and either adapt to them accordingly, or deny access to unauthorized users. Multiple modalities are available to control the system, each customized to user's personal preferences and habits.

User identification is typically based on a user name and password. Within the context of HBB-Next [2], a European research project, new standard [3] was developed where face [4], [5] and voice recognition are used as main identification approaches. However, other modalities, such as fingerprint recognition, iris recognition, etc., open the possibility of multi-level identification and authentication. System control, the second main part of the MMI, includes voice command navigation, gesture recognition, eye tracking, etc. Several examples show possible applications of MMI, such as voting or direct shopping while watching TV.

3 System Navigation

Focusing on system navigation, gesture and voice command recognition are the key modalities that allow for more natural interactions between humans and computers as they are relatively well examined and easy to implement from the

practical point of view. However, there are a few considerations that have to be taken into account.

The present gesture sets are based on physical input devices used with computers. Simply said, they try to "remove" the device, but keep the same usage patterns, mostly in order to avoid the learned gesture problem. In order to come closer to a natural (touch-less) gesture-based operation, the concept has to change so that gesture sets are designed bottom-up, like if there were no other devices than gestural sensors. Our team has examined several gesture recognition approaches, each serving a different purpose. By combining them, we aim to use the most suitable method for specific situations.

The one feature that is more obvious and expected by general users of an intelligent multimedia system is the voice navigation. Just like the gesture navigation, the voice navigation represents a natural interface between computers and humans. And just like the gesture navigation, the voice navigation's first requirement is to be intuitive and comfortable. This task seems less demanding when compared to gesture recognition, especially since voice recognition is not influenced by any device or sensor used to acquire the voice.

3.1 Intuitive and Natural Gesture Navigation

One of the greatest drawbacks of wider use of natural user interfaces is their lack of usability and human-centred design. While other modalities (i.e. the voice command navigation) seem to adapt rather quickly, the gesture recognition still cannot deliver truly natural experience, especially on touch-less devices. There are several factors that determine whether the gesture recognition is a natural and intuitive process. Firstly, there are the hardware limitations that limit sensor algorithm's ability to recognize more specific details in a gesture performance. This causes gestures to be recognized incorrectly and forces users to perform gestures that are not intuitive, require plenty of effort and lack comfort. System designers tend to overcome the sensor limitations by introducing gestures that are easily recognizable but are often far from simple.

Gestures can be divided into two basic categories by user experience. Innate gestures are based on the general experience of all users such as to move an object to the right by moving hand to the right, catch an object with closed fingers, etc. Naturally, the innate gestures can be affected by habits or culture. With the innate gestures there is no need for a user to study them in order to get good gesture experience, they just need to be shown to him. The second category is learned gestures, which need to be learned.

The gestures can also be divided into three categories based on the notion of motion [6]. Static gestures represent shapes created by gesturing limbs, which carry a meaningful information. The recognition of each gesture is ambiguous due to the occlusion of the limb's shape and, on the higher level of recognition, the

actual meaning of the gesture based on local cultural properties. The second category, continuous gestures serve as a base for an application interaction where no specific pose is recognized, but a movement alone is used to recognize the meaning of a gesture. Dynamic gestures consist of a specific, pre-defined movement of the gesturing limb. Such gesture is used to either manipulate an object, or to send out a control command. There is a problem with humans' inherent inability to perform a gesture in exactly the same dynamics, distance and manner. Additionally, these three groups can be combined in different ways, for example the static posture of a hand with the dynamic movement of an arm.

The general idea behind combining gesture methods is to utilize the best method for each individual action. Where a swiping is a natural approach, trajectory tracking should not be used, and where a simple static gesture serves well, a dynamic gesture would be a waste of resources. The time spent with gesture control relates to the amount of the energy spent, so if the application control requires more energy, then users will use less gestures and more traditional forms of control.

Neural networks and genetic algorithms were mostly used in the beginnings of gesture recognition. These methods had an acceptable recognition rate, but the greatest drawback was the amount of a necessary computing power and time needed for the training of neural networks which were significant, and unacceptably high for practical applications. Nowadays, different techniques are used to recognize gestures, since algorithms which do not require neuron networks have been invented, for example the Golden Section Search, the Incremental Recognition Algorithm and probabilistic models like the Hidden Markov Model (HMM). To increase the success rate of these algorithms, machine learning can be used. There are many approaches how to implement the gesture recognition. HMM methods are one of them, the main reason being that HMM approach is well known and used in many areas. One interesting approach how to implement the HMM into gesture recognition is shown in [7] where the author describes his own method step-by-step, which consists of:

- Gesture modelling
- Gesture analysis
- Gesture recognition

The author uses a Kinect v1 sensor as the input device. There are some problems with a centre hand point because of Kinect's inaccuracy. The starting and ending point of a gesture are determined by using a "static state", where the static state is accepted when the hand is kept relatively still. When a person performs a motion, this movement will be recorded and compared against a database. HMM in this paper was used for training and recognition only. Only basic gestures were tested such as "left", "right", "up", "down" and letters "S", "E", "O". The directions had a very good success rate but the remaining three letters had an average success

rate of about 90%, which is disappointing, given by that the database consists of seven gestures only.

Many approaches which use HMM scheme are based on RGB camera sensing. But just in the last three years the researches started to solve depth images using motion sensors [8], [9] very intensively.

Our research focuses on the gesture recognition area, where we want the user to be able to control all room equipment and devices via the gesture-controlled TV application. The gesture recognition should work robustly in changing light conditions. This is achieved by using the IR-based depth camera incorporated in the Kinect sensor as it has been shown it is susceptible only to strong sources of light [10]. In our MMI application three types of gestures are implemented: static gestures, dynamic gestures and swipe gestures which are a subdomain of dynamic gestures but employ a different algorithm. Each of these methods has several unique usages. Static gestures are used as an additional symbol for dynamic gestures, or as a symbol for the start and the end of dynamic gestures: if a user shows five fingers of a hand, the system allows him to perform dynamic gestures. If the user wants to end the dynamic gesture session, the user closes his palm. The static gesture can be used as a volume controller in combination with palm rotation.

3.2 Static Gestures

We researched several static gesture algorithms [10], finding the modification of Part-based Hand Gesture Recognition (HGR) algorithm [12] as being the most reliable. In our approach a binary image of a hand area is adjusted in order to obtain a convexity hull (polygon created by connecting all extremes around the hand) and its defects. The convexity hull determines a border between two different image parts. To accurately determine a centre of the palm, the author of [12] applied an inner circle, which brings several problems, like false detection or higher computational power needed in some hand postures. To avoid this, a circumscription which is more robust against hand tilt is used. To cope with hull shapes with extreme convexity a point onto contour hull is added that belongs to the hand and has the maximal distance from the found defect.

We implemented our own method to omit the forearm area. The circle is created with centre being the centre of a palm. Then, the two intersections are found with the contour closest to the Kinect-detected elbow point, taking the shortest distance between the centre of a palm and the found points as the circle's radius as given by:

$$Min\ (pointA, pointB) \tag{1}$$

Where *pointA* and *pointB* are intersections of the circumscription and the contour.

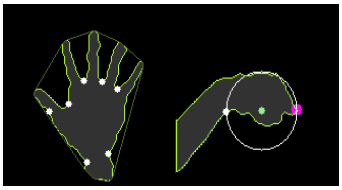

Figure 1
Finding the centre of the palm

The function of two variables is used for the representation of a hand shape. A hand contour is mapped onto X-axis, and Y-axis then describes the relative distance of each point from the centre of the palm (see **Hiba! A hivatkozási forrás nem található.**). Although this implementation is not trivial, its result is easily readable and clearly shows the hand proportions. A search for local maxima and minima is performed as a part of the contour analysis. The first and last local extremes must be local minima; otherwise local maxima at the beginning and end are removed. We modified the original implementation because it caused loss of some important higher relative distance extremes, and was ineffective for lower relative distances.

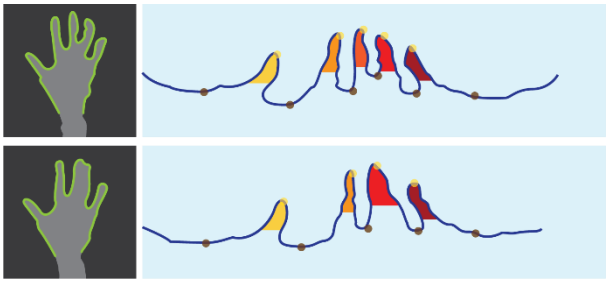

Figure 2
Static gestures are recognized by counting fingertips on a curve spread from the hand contour. Combination of static and dynamic/swipe gestures will let people use more natural gestures over traditional forms of control.

A set of tests was performed for the described method comparing with the methods [13], [14]. We attempted to estimate the complexity of each algorithm in terms of the number of the specific steps needed to obtain a recognized gesture. This information leads to a processing power and delay limitations. The maximum distance from the sensor was measured. Then, the rotation boundaries of the hand were measured in which the algorithms are able to perform reliably. The level of freedom describes the relative distance between the fingers in order to be recognized as separate fingers (0% for max. distance and 100% for joined fingers). This is supported by information about joined fingers detection based on the algorithmic properties of each method. In order to measure the success rate, the gestures were performed 110 cm from the sensor. Four subjects were tasked to

perform 100 gestures by showing the different number of fingers in various hand positions, creating a test set of 400 gestures. The results are summarized in Table 1 and Table 2.

Table 1
Algorithm Comparison [10]

	Convexity Defects	K-Curvature	Part-based HGR
Sensor Distance 80cm to	119 cm	160 cm	175 cm
Algorithm Complexity # specific steps	7	6	7
Joined Fingers Detection	NO	NO	YES
Relative Level of Freedom	40%	70%	95%
Success Rate	80%	92%	90%

Table 2
Algorithm Comparison – Hand Rotation limitations for each algorithm[1] [10]

	Hand Rotation Boundaries					
Convexity Defects	X axis		Y axis		Z axis	
	35°	75°	25°	30°	180°	65°
K-curvature	X axis		Y axis		Z axis	
	35°	75°	25°	40°	175°	170°
Part-based HGR	X axis		Y axis		Z axis	
	50°	85°	25°	40°	150°	125°

Of the three evaluated algorithms, the *Convexity Defects* approach has proved to be the least reliable. Even though the success rate could be considered acceptable, it was very susceptible to a noisy input (given by hand rotation boundaries and a level of freedom). The *K-Curvature* [13] method provided the best results in terms of overall success rate. Additionally, this method is applicable to the widest range of the possible hand rotations from the trio. However, the level of freedom is the bottleneck of the approach. The third analysed method proved to be reliable and is the most robust of the three. With its alternative approach to counting of fingers it is able to distinguish even joined finger given the input image falls within the rotation boundaries. As it was shown each of the methods can be quite reliably used for the static gesture recognition, when in compliance with each method's unique properties.

[1] All angles are relative to default hand position: open palm with index finger in line with Y axis. In X axis, the first angle describes rotation heading front, the other heading back. In Z axis, first angle is rotation counter-clockwise, second angle clockwise (as viewed by the performer). In Y axis, first angle describes rotation to the left, second to the right.

3.3 Dynamic Gestures for Better Interaction

Dynamic gestures are used to provide an authorization to a private content. They are used as a password key. A user can perform a dynamic gesture and the likeliness of the template and the performed gesture are compared via the use of an incremental recognition algorithm proposed by Kristensson and Denby [15], originally designed for digital pen strokes and touch-screen devices. For this approach, a template is defined as a set of segments describing the template gesture. Each segment describes progressively increasing parts of the template gesture so that the first segment is a subset of the second segment, which is a subset of the third segment, etc., and the last segment represents the whole gesture template. Each segment is represented as a series of time-ordered points.

With each new point of the observed gesture the system computes a Bayessian posterior probability that the gesture matches a gesture template, for each template, as given by the formula:

$$P(\omega_j|\mathrm{I}_i) = \frac{\mathrm{P}(\omega_j)P(\mathrm{I}_i|\omega_j)}{\sum_k \mathrm{P}(\omega_k)P(\mathrm{I}_i|\omega_k)} \qquad (2)$$

where $P(\omega_j)$ is the prior probability, $P(\mathbf{I}_i|\omega_j)$ is the likelihood and the denominator is the marginalization term. The prior probability can be used to influence the posterior probability when the distribution of the template probabilities is known. For example, if the probability of each gesture occurrence is known then more precise and successful recognition may be obtained.

The likelihood measure is given as a probability that the part of the observed gesture matches a gesture template:

$$P(\mathrm{I}_i|\omega_j) = P_l(\mathrm{I}_i|\omega_j)E(\mathrm{I}_i|\omega_j), \qquad (3)$$

where $P_l(\mathbf{I}_i|\omega_j)$ is the likelihood of the observed gesture and the respective part of gesture template. It is given as the max of the distance function D taking into account the Euclidean distances between the normalized points of the observed gesture and template segment, and the turning angle between the two point sequences. $E(\mathbf{I}_i|\omega_j)$ is an end-point detection term which serves to favour the complete gestures compared to the parts of the gestures in the case when one full template represents a part of a different template.

The distance function is given by formula:

$$D(I,S) = \exp\left(-\left[\lambda\left(\frac{x_e^2}{\sigma_e^2}\right) + (1-\lambda)\left(\frac{x_t^2}{\sigma_t^2}\right)\right]\right) \qquad (4)$$

The distance function depends on both Euclidean distance x_e between the corresponding points of the recorded trajectory **I** and the known template **S**, and the mean turning angle x_t between the respective line segments of the **I** and **S** sequences. The contribution of the two measures is managed with the variable λ which allows to favour one of the measures against the other.

The posterior probabilities are then filtered using a window over last five predictions to stabilize them. The interested reader may find a more detailed description of the original algorithm in [15].

For our purposes, the algorithm was altered to make use of the depth data provided by the Kinect sensor. The gesture recognition process is triggered by the user's hand movement and the movement's trail is examined in real time by comparing it with the parts of the predefined gesture templates. The set of templates is compared to the performed gesture in real time and templates that do not match the performed sample with pre-set certainty are continuously removed from the set. In this way the algorithm provides a decision on which gesture was performed with decreasing ambiguity, until only one template gesture remains having the highest probability. It is obvious that given a set of gestures which are sufficiently distinguishable from each other the recognition may be successful after only a part of the gesture was performed. An application was created to test the proposed algorithm. The application includes a set of gesture templates which can be extended with custom gestures. Gesture recording works as follows. The Kinect sensor input is used to obtain a trajectory of the performed gesture, consisting of individual points. The trajectory is then reduced in size to fit in 1000x1000 points and saved to the template group.

The default set of gestures consists of capital letters of English alphabet (26 letters). Each gesture was performed five times by four persons, creating a test set of gestures consisting of 520 gestures. It is important to note that not all of the gestures were performed with high accuracy. As opposed, some of the gestures were performed imprecisely and inconsistently with the attempt to examine the gesture variability. On this data set the overall success rate was above 91%. The average distance of users from Kinect, which is an important parameter when considering touch-less environments, was approximately 1.8 meters. Some limits of Kinect sensor by testing were determined. Kinect's accuracy decreases with the growing distance between the sensor and users. This argument is also confirmed in [16], [17]. Smoothing into individual joints was applied to eliminate low accuracy. Then we obtained slightly smooth curves on our imaginary surface. This improvement helped us to achieve more successful results and remove some problems which originated from bad accuracy.

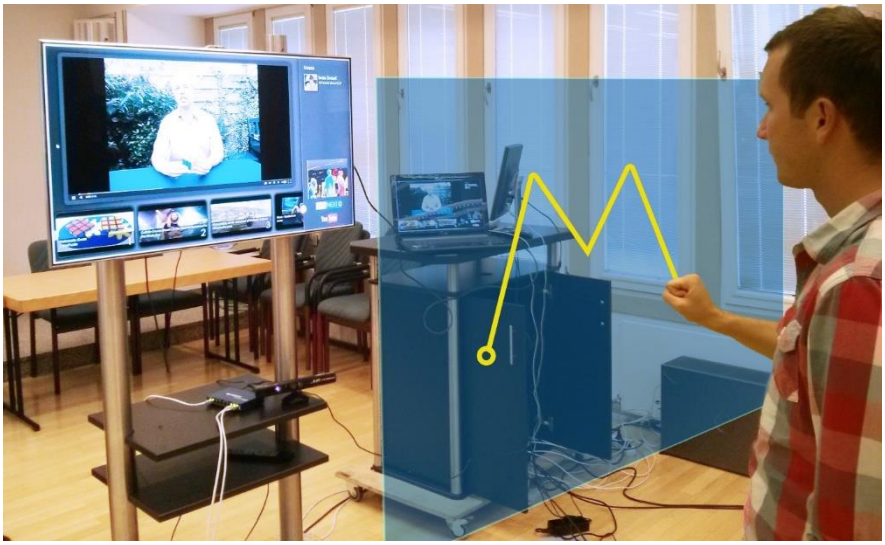

Figure 3
The incremental recognition algorithm recognizes the gesture as it is being performed on a virtual surface, simulating a touch panel of a tablet. System works with the gesture similarly to smartphone unlocking pattern.

The testing equipment consists of the hardware fulfilling the minimal hardware requirements for the Kinect v1 sensor, namely Windows 7 or Windows 8.1 operating systems (tested on both systems), dual-core 2.66 GHz CPU, 2GB of RAM and USB 2.0 connector for the sensor connection. The software used with the Kinect sensor is Kinect SDK 1.8 and EmguCV 2.4.2.

The usage of gestures is extended by swipe gestures. This gesture type brings in a very natural and comfortable approach. Swipe gestures are designed for fast and routine browsing in the menu, programs, or gallery as they consist of 4 directions of hand movement for each hand and combinations of both hands. Our method called Circle Dynamic Gesture Recognition (CDGR) is based on hand detection, speed of movement and distance (see Figure 4). While the hand is in an inner circle, the system stays inactive. After the user crosses the inner circle a short countdown is started. During the countdown the system observes if the user's hand crosses the outer circle. If the countdown reaches a limit before the hand crosses the outer circle, the method will reset and the hand will be again in the center of both circles. So, if the user moves his hand slowly, both circles will follow its joint and no gesture will be recognized. If a human hand executes a faster motion and the inner circle leaves the outer circle, the system processes this motion and determines a gesture.

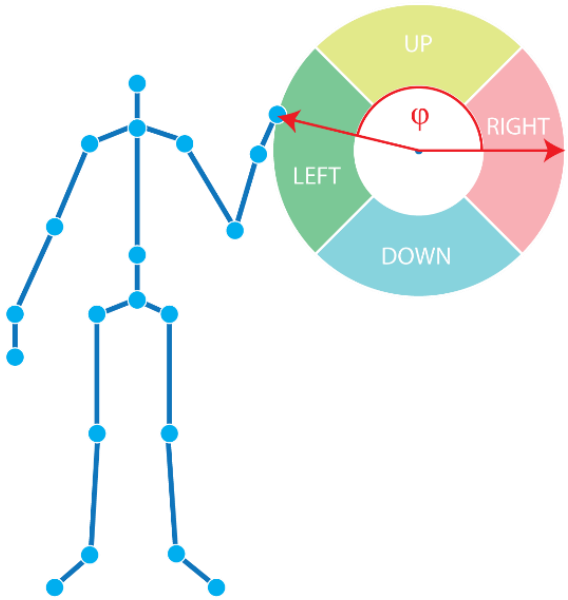

Figure 4
Swipe gestures are fast to perform and reliable to read

The gesture is given by the angle of the performed motion from the middle to the outer circle. Initially, the possible gestures are: swipe left, right, up and down. The gestures can be performed by both hands individually, or as a combination. This allows the user to perform more complex gestures, such as zoom in and zoom out.

During the testing 10 people performed a success test; each person performed 40 gestures, creating a set of 400 performed gestures in total. Our algorithm has proved reliable with 94% success rate for four defined gestures for each hand. The big advantage of the swipe method is its low computational complexity, high precision and easy implementation for many purposes.

3.4 Voice Commands Navigation

Thanks to its complexity and ability to convey deep and thoughtful message in simple form, speech recognition could be one of the most comfortable ways of natural interaction between humans and computers. In the last few years, there has been a considerable leap in development as processing power ceased to be the limiting factor for using advanced algorithms. This is mainly visible in the field of smart personal devices where the biggest players like Google, Apple and Microsoft introduced their personal assistant services. One of the key advancements in their technology, from usability point of view, is shift from command based to conversational based control. In the past, voice navigation was limited by the processing speed of the local device which lead to limited

recognizable vocabulary. This yielded either a few-command set to cover the general domain or a highly limited command domain. Extending the command set usually lead to higher ambiguity given finite set of the describing parameters. This has changed with moving the processing power to the cloud where there are virtually unlimited computational resources as well as storage.

Other restraints of voice commanding include variable and unpredictable acoustic conditions which allow reliable voice command recognition only in a controlled environment (no outside sounds etc.). Also, the success rate of recognition is highly user-dependent: apart from different accents or dictionaries of each individual, even the same person doesn't say the command in exactly the same way twice. This could be avoided partly by training the recognizer by the voice of the person, either before or during usage.

The voice command recognition, just like any other recognition, consists of two principal steps. During learning, the system has to be taught what inputs it may expect and what they mean. Secondly, during recognition, an unknown input pattern is presented and a closest match from the learned pattern set is chosen. Both steps are demanding either in terms of data quality and quantity (learning phase) or quality and speed (recognition phase). Additionally, the concept of continuous learning while recognizing is a logical enhancement of the original 2-step process.

There are currently a number of methods available that can be applied for voice command recognition, which differ greatly in approach as well as complexity. For example, Dynamic Time Warping, which is mentioned further in the text, or Hidden Markov Models that lead to good recognition success rates (up to 98.9 %, as in [18]) without being too demanding no computational power. Other, more complex approaches include neural networks, which experience a renaissance these days as computational power, storage and fast network connection are easily available. Namely, it is techniques like Deep Belief Networks, Convolutional Neural Networks or the late Hierarchical Temporal Memory which aim at modelling and learning relationships between features in the input signal in space and/or time.

With computational options increasing and becoming widely and easily available modern algorithms can look not only at the traditional properties of speech input but also on the more delicate features that had to be omitted before: emotion and context. Both features contain plenty of additional information that give humans the higher idea of the meaning of respective commands. For example, there is difference between prescriptive and calm tone of voice, just as there is difference whether the same word is heard within a fluent speech or the same word is uttered isolated. This is the area where neural networks now play an important role.

In our multimodal interface we implemented voice command recognition based on MFCC and DTW algorithm. Because range of commands used in MMI is not so wide and in different sections of MMI different groups of voice commands are

used (max 10 commands per area), it was not necessary to employ more advanced classification algorithms. We find DTW algorithm to be sufficient for our purposes.

Extraction of best parametric representation of human voice is one of the most important parts to achieve good recognition performance. In our case we decided to use MFCC coefficients. MFCC coefficients are a representation of the short-term powered spectrum on a non-linear mel-scale of frequency. Human auditory system is not linear and mel frequency scale fits it much better than linear frequency scale. The relationship between mel and linear frequency scale is given by (5):

$$F(mel) = 2595 * log(1 + f/700) \qquad (5)$$

We used 13 MFCC coefficients plus delta and delta-delta features (39 coefficients together). Since MFCC coefficients represent only power spectral envelope of the time frame, but there is also information in spectral variation, we used delta and delta-delta features. The delta coefficients can be calculated as follows (6):

$$\Delta c(m) = \frac{\sum_{i=1}^{k} i * \big(c(m+1) - c(m-1)\big)}{2 * \sum_{i=1}^{k} i^2} \qquad (6)$$

The same formula is used for delta-delta coefficients calculation, where MFCC coefficients are replaced with delta coefficients [19]. On these features DTW algorithm mentioned above was applied. DTW is a computationally inexpensive algorithm to measure the similarity between two temporal sequences which may vary in time or speed. In general, this approach calculates an optimal match between 2 given sequences with certain restrictions.

In our consideration, voice command recognition will be applied maximally on 10 commands in one section. In our testing 10 commands were tested in 200 experiments. The success rate which was achieved was 95%, which is sufficient for our purposes.

4 Multimodal Application

In our application research, we focused on natural multimodal interface and its integration into a multimedia system used on daily basis. The vision of the system is to control the TV and access multimedia content using larger number of modalities. Obviously, the usage of multimodal interface is not limited only to the TV system but has many different applications.

Figure 5
Logic behind the schematic of the multimodal interface shows applications served by the MMI controller which collects recognition information from individual modalities. All of them use input from the Kinect v1 sensor delivered by the MMI input hub.

The block diagram (see Figure 5) shows the concept of the multimodal interface divided into five layers. Physical layer represents hardware input and output devices which enable interaction with the real-world. The input device is currently represented by the Kinect sensor. Kinect is a multifunctional device which can be effectively used by each of the modalities mentioned above, for example, a microphone array for speaker identification, depth camera for gesture recognition, RGB camera for face recognition etc. Multimodal data provided by the Kinect sensor are collected by the HUB which serves as a distribution point of the input data from the Kinect sensor to multiple applications each utilizing different modalities. This is due to the technical limitation that allows the Kinect to communicate with only one application at a time. Modalities described in previous sections are represented as modules with defined APIs. Data obtained from Kinect sensor are then processed in parallel by each module separately. The modular, API-based structure allows to simplify the process of adding new modalities. The MMI controller collects output data from all modules, evaluates and combines it into one output data stream. The stream contains information about recognized users and their requested actions. Applications only depend on MMI controller

output so there is no limit in installing new applications, thus extending overall MMI functionality.

The currently implemented application consists of three micro-applications that cooperate to produce a UI on the TV screen. The first application is designed for video playback and can be easily maximized to the full screen using a simple gesture. The second application, located on the right side of the display, shows a list of users identified by speech or face recognition modules. Only users in this list are permitted to control the TV using predefined set of gestures, voice commands or other modalities, and combinations thereof, such as gestures with voice commands. When the user leaves the room, he/she is automatically removed from the list. The third application displays a list of recommended channels. Depending on user viewing preferences, system provides recommendations that best suit all users in front of the TV. Using swipe gestures a user is able to navigate this list, play or stop the video. To demonstrate the security possibilities of the system, some of the recommended channels are locked. It means that users without permission are not allowed to watch such content until they enter the secret pattern. To enter the secret pattern, we apply dynamic gestures.

In order to make the best use of multimodal interface, it is not always necessary to use touch-less gestures to perform every action. Some actions will always be better executed by using a different modality. I.e. entering text would be difficult, time consuming and by all means uncomfortable using gestures, but can be easily and faster performed with speech recognition. With this in mind, it becomes necessary to introduce an integration platform that will provide applications with requested inputs where the application does not need to know the source modality, if not required explicitly.

Within our research we have designed and implemented a multimedia system making use of several of the modalities mentioned earlier. Namely, the system uses face recognition and speaker identification for user authentication, and swipe gestures, dynamic gestures with static postures and voice command recognition for system control. In order to test the system as a whole, we have devised several use case scenarios where each of the modalities is employed. Thanks to the proposed layered model design, different applications may use different modalities. The modular structure allows for easy deployment of new applications like new ways of TV and room control, multi-device support, controlled access, etc.

5 How to Use It: Scenarios

A system that is aware of its users, knows their habits and interests, can become an intelligent concierge of the household, and can provide advanced interconnections between various services. Here we present only a few ideas of

the possible applications. Some of them are already in use with other being most certainly proposed.

5.1 Shopping while Watching

The dream of teleshopping is becoming true as connected TVs allow to make orders from the TV seat. Going a bit further, the next generation of TV shopping will happen (if not happening already) directly during watching the program. Broadcasters or 3rd party providers annotate the TV program with offerings of products and services related to the program. This additional information displays to the viewers as an optional information giving them the possibility to directly order that nice couch, brand of beer or skirt worn by their favourite actress as they see it in action on screen. Similarly, viewers can schedule for various medical procedures, or apply to subscription services. All is available at a pressing of a button or waving a gesture.

5.2 Smart Household

The integrated TV system can reach beyond the recommendation of TV channels and become the central information and operation hub of the whole household. Family members can get notified of any events that happen in the house, be it washing machine alerts, fridge notifications that the champagne is ready-chilled, or new mail in the mailbox. Additionally, the whole household can be operated from the comfort of the living room, no matter if you want to heat the room up a bit, close curtains or order groceries for delivery. Such system however requires home appliances that are interconnected with the TV system, which should not be a problem in the near future as connected appliances are already in production by several manufacturers.

5.3 Voting

Nowadays, people want to spend their time in front of the TV effectively. They watch preferred channels, programs, TV shows, that are recommended by their friends, family or colleagues. TV programs' or films' ratings are usually available in public databases such as IMDb.com, where the rating is provided by individual viewers. However, to access the rating one has to quit watching the TV and switch to the website in order to rate or obtain the rating of the programme. A more sophisticated system can collect this information immediately after the program finishes, while the viewer executes only a simple gesture. A like/dislike gesture (thumb up/thumb down) or swipe-left/swipe-right gestures may be used to obtain the rating. The main idea is to use very simple and very easily executable gesture, as users do not want to execute complicated gestures while relaxing. After watching the TV, the system automatically invites the viewer to rate the programme, and the viewer can execute the easy gesture in two seconds. The

system collects these opinions and creates global statistics and recommendations for the viewer's friends and family, as well as for the general audience, with the possibility to update the already existing rating systems.

5.4 Digital Doorman

Digital doorman is a feature that helps to keep watching TV a comfortable experience. The typical situation presents a user watching his/her favourite programme while a guest rings at the front door. Usually the user has to interrupt watching, stand up from the sofa and go to check the door phone. It is very likely that he/she will miss a short part of the favourite program. Digital doorman brings the highest comfort utilizing interconnection between multimodal interface and the doorman. In the upper corner of the screen the user can see a live camera stream from the doorbell and immediately can allow or deny the access to the building using an easy gesture or a voice command.

5.5 Phone Pickup

Another application that extends the functionality of the multimodal interface is called phone pickup. It often happens that during watching an exciting TV programme or just having fun with friends a phone suddenly starts to ring, and it is difficult or not comfortable to answer it. Multimodal interface simply enables to pick up or cancel the call in the comfort of the living room, share calls with friends etc. This functionality can be simply achieved by recognizing the phone's owner in front of the camera and offering a remote phone pickup. A phone call can be easily picked up using a voice command or a particular gesture command. The main advantage of this extension is pausing the TV channel playback during the phone call without losing comfort.

Conclusions

In this article we proposed a multimodal interface architecture with implemented voice and face recognition, gesture recognition, and voice command navigation.

Gesture recognition methods, discussed in this article, offer high reliability and can be used in a wide range of applications. The presented results show highly satisfactory recognition efficiency of the third presented method compared to the results of the other two methods for static gesture recognition, and can be applied in practical applications. A very good rate was achieved also by our own method Circle Dynamic Gesture Recognition for swipe gestures. The methods with the highest reliability were implemented into the multimodal interface for system control. We suggest that with the proper configuration of the presented methods a more intuitive gesture navigation can be achieved.

In the section devoted to multimodal applications we introduced a concept of a modular architecture for the multimodal interface. This architecture consists of

five layers with well-defined interfaces between each other. This is in accordance with the Kinect sensor being used as the core device with all the limitations implied by its APIs. Thanks to the modular architecture, the multimodal interface can be easily extended using additional modalities, input devices and micro-applications.

Despite the number of advanced features integrated in the Multimodal Control prototype application, further research is required not only in the area of more sophisticated modalities but also in the implementation of a more complex concept of the whole system. We investigate the possibilities to use the MMI in a complex intelligent room comprising multimodal control of other smart devices like light switches, sockets, air conditioning, etc. The multilevel authorization module needs to be extended for biometric methods and to consider advanced solutions such as identification via mobile devices, NFC tags or RFIDs and many others.

In near future, we plan to extend the whole system with an administration module for an easy and intuitive appearance and personalization of the application.

Acknowledgement

The authors hereby declare that the research leading to his article has been funded by the grant VEGA-1/0708/13 IMUROSA and APVV-0258-12 MUFLON.

References

[1] S. Oviatt.: Multimodal Interfaces. In The Human-Computer Interaction Handbook, Julie A. Jacko and Andrew Sears (Eds.) L. Erlbaum Associates Inc., Hillsdale, NJ, USA 286-304

[2] https://web.archive.org/web/20150711074144/http://www.hbb-next.eu/

[3] ETSI TS 102 796 v1.2.1, Hybrid Broadcast Broadband TV, European Telecommunications Standards Inst., 2012; www.etsi.org/deliver

[4] M. Oravec: Biometric Face Recognition by Machine Learning and Neural Networks, invited paper, The 5th International Conference on Innovations in Bio-Inspired Computing and Applications IBICA 2014 / International Conference on Prediction, Modeling and Analysis of Complex Systems NOSTRADAMUS 2014, June 23-25, 2014, Ostrava, Czech republic

[5] M. Oravec, J. Pavlovičová, J. Mazanec, Ľ Omelina, M. Féder, J. Ban, M. Valčo, M. Zelina: Face Recognition in Biometrics (Metódy strojového učenia na extrakciu príznakov a rozpoznávanie vzorov 2: Rozpoznávanie tvárí v biometrii), Publisher Felia, Bratislava, 2013, ISBN 978-80-971512-0-1

[6] Vanco, M.; Minarik, I.; Rozinaj, G., Dynamic Gesture Recognition for Next Generation Home Multimedia, in ELMAR, 2013 55th International Symposium, pp. 219,222, 25-27 Sept. 2013

[7] Y. Wang, C. Yang, X. Wu, S. Xu and H. Li: Kinect Based Dynamic Hand Gesture Recognition Algorithm Research, Proceedings of the 4th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2012, Vol. 1, pp. 274-279

[8] K. Lai, J. Konrad and P. Ishwar: A Gesture-driven Computer Interface using Kinect, Image Analysis and Interpretation (SSIAI), 2012 IEEE Southwest Symposium, 2012, pp. 185-188

[9] N. Villaroman, D. Rowe and B. Swan: Teaching Natural user Interaction using OpenNI and the Microsoft Kinect Sensor, Proceedings of the 2011 Conference on Information Technology Education, 2011, pp. 227-232

[10] Khoshelham K, Elberink SO: Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. Sensors (Basel, Switzerland). 2012;12(2):1437-1454. doi:10.3390/s120201437

[11] Vanco, M.; Minarik, I.; Rozinaj, G., Evaluation of static Hand Gesture algorithms, in 2014 International Conference on Systems, Signals and Image Processing (IWSSIP), 2014, Vol., No., pp.83-86

[12] Z. Ren, J. Yuan, J. Meng and Z. Zhang: Robust Part-Based Hand Gesture Recognition Using Kinect Sensor, Multimedia, IEEE Transactions on Vol. 15, No. 5, 2013, pp. 1110-1120

[13] F. Trapero Cerezo: 3D Hand and Finger Recognition using Kinect, available at http://www.scribd.com/doc/161562314/Finger-and-Hand-Tracking-With-Kinect-SDK-3

[14] M. Vančo, I. Minárik and G. Rozinaj: Gesture Identification for System Navigation in 3D Scene. In Proceedings ELMAR-2012: 54th Symposium ELMAR-2012, 12-14 September 2012 Zadar, Croatia. Zadar: Croatian Society Electronics in Marine, 2012, s.45-48. ISBN 978-953-7044-13-8

[15] P. O. Kristensson and L. C. Denby: Continuous Recognition and Visualization of Pen Strokes and Touch-Screen Gestures, Proceedings of the Eighth Eurographics Symposium on Sketch-based Interfaces and Modeling, 2011, pp. 95-102

[16] K. Khoshelham: Accuracy Analysis of Kinect Depth Data, in ISPRS Workshop Laser Scanning, 2011, Vol. 38, p.

[17] B. Molnár, C. K. Toth, and A. Detrekoi: Accuracy Test of Microsoft Kinect for Human Morphologic Measurements, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 3, 2012, pp. 543-547

[18] J. Kacur and V. Chudy: Topological Invariants as Speech Features for Automatic Speech Recognition, International Journal of Signal and Imaging Systems Engineering, Vol. 7, No. 4, 2014, pp. 235-244

[19] Sreenivasa Rao, K.; Nandi, D.: Language Identification Using Excitation Source Features, Springer International Publishing, 2015, ISBN 978-3-319-17725-0