Multimodal Desktop Interaction: The Face –Object-Gesture–Voice Example

Vidakis, Nikolas; Vlasopoulos, Anastasios; Kounalakis, Tsampikos; Varchalamas, Petros; Dimitriou, Michalis; Kalliatakis, Gregory; Syntychakis, Efthimios; Christofakis, John; Triantafyllidis, Georgios

Published in:
18th International Conference on Digital Signal Processing (DSP)

DOI (link to publication from Publisher):
10.1109/ICDSP.2013.6622782

Publication date:
2013

Document Version
Accepted author manuscript, peer reviewed version

Multimodal Desktop Interaction: The Face – Object - Gesture – Voice Example

Nikolas Vidakis*, Anastasios Vlasopoulos*, Tsampikos Kounalakis§, Petros Varchalamas*, Michalis Dimitriou*, Gregory Kalliatakis*, Efthimios Syntychakis*, John Christofakis* and Georgios Triantafyllidis†
* Applied Informatics and Multimedia Dept., Technological Educational Institute of Crete, Heraklion, Greece
e-mail: {vidakis, epp1281, epp1978, epp887}@epp.teicrete.gr, demetriou.mixalis@gmail.com, gkalliatakis@yahoo.gr, pepemakis@hotmail.com
§ Electronic and Computer Engineering Dept., Brunel University, London, UK
e-mail: t.kounalakis@gmail.com
† Medialogy Section, Aalborg University, Copenhagen, Denmark
e-mail: gt@create.aau.dk

***Abstract*— This paper presents a natural user interface system based on multimodal human computer interaction, which operates as an intermediate module between the user and the operating system. The aim of this work is to demonstrate a multimodal system which gives users the ability to interact with desktop applications using face, objects, voice and gestures. These human behaviors constitute the input qualifiers to the system. Microsoft Kinect multi-sensor was utilized as input device in order to succeed the natural user interaction, mainly due to the multimodal capabilities offered by this device. We demonstrate scenarios which contain all the functions and capabilities of our system from the perspective of natural user interaction.**

Keywords-component; Multimodal Interaction; Natural Interaction; MS-Kinect; Multimodal Input

I. INTRODUCTION

Over the past decade, in the field of human computer interaction (HCI) there is a great interest in interacting via multimodal mediums.

Since Bolt's [2] "Put that there" command proposal, a new wave has been paved in HCI and more precisely in Natural User Interfaces. Multimodality came to prominence and a lot of research has been done in order to determine the right principles of a functional multimodal system from the perspective of natural user interaction. Apart from the natural way human prefer to communicate with computer [20], users tend to interact multimodaly instead of unimodal [6, 7]. So, the way we face interaction is changing in the direction of multimodality.

According to Bill Buxton's [3] vision of natural and multimodal human computer interaction, researches on this field are trying to implement this vision under the best possible conditions. One of the several difficulties experienced by researchers during multimodal natural user interface design is the absence of stable and reliable input devices in addition with their cost. In recent years a worthwhile endeavor is taking place in this field. The latest multimodal input devices, like MS-Kinect [27] or Xtion Pro [28], enabled research community to overcome adversities like cost, stability and usability. Recently, Robotics [22] and virtual reality (VR) [23] have taken advantage of this evolution and many researchers, in these fields, have developed systems based on multimodal input device evolution [15, 16, 18, 19].

Current research trends, suggest that natural-based interaction is the wave of the future [21], with considerable attention from both the research community (see recent survey articles by Mitra & Acharya [10] and by Weinland et al. [13]) and the industry (see MS-Kinect [27], Asus Xtion PRO LIVE[28]). Evidence can also be found in a wide range of potential application areas, such as medical devices [15], video gaming [15], robotics [16, 19], ambient intelligent environments [18], and physical rehabilitation [12] etc.

Scientific work on natural interaction, combined with devices like Kinect, define multimodality as: (a) every distinct section of the human body is considered as different modality and (b) depth signal and RGB signal are considered as different modalities. The next paragraphs present three different systems that use Kinect as a multimodal input device in order to offer natural interaction to their users. These systems represent different research disciplines such as medicine, virtual reality and robotics.

Gallo's et al. developed a controller-free exploration of medical image data [17]. The project allows users to interact at a distance through hand and arm gestures, giving them the opportunity to explore medical images. With the use of OpenNI as a Kinect communication library, and Open CV [26] Computer Vision library for image processing, user has the ability to navigate, click, rotate, translate, zoom, scale and erase an image from the screen. The gestures were one to one assigned to a command and could not be changed by the user.

Suma et al. developed a middleware called The Flexible Action and Articulated Skeleton Toolkit (FAAST) [15] to facilitate integration of full-body control using OpenNI-compliant depth sensors (currently the PrimeSensor and the Microsoft Kinect). FAAST incorporates a VRPN server for streaming the user's skeleton joints over a network, which provides a convenient interface for custom virtual reality applications and games. Although its flexibility at assigning

different commands to the users movement, it uses only one type of input.

Stowers et al. on their project use the Kinect device, built on a quad rotor, to control its altitude by using the information received from both the depth and image sensors [16]. Another similar approach is developed by Michael Van den Bergh et al. They use the Kinect device to direct a robot with human 3D hand gestures real time [19] by using Depth and RGB sensor information.

In this paper, we demonstrate our work in progress with regards to multimodal natural user interface system which is based on real-time audio, video and depth signal processing. The intermediate input device between user and system is MS-Kinect and the signals processing is performed by device's API. For supporting multi-applications [29], a generic container has been designed which runs at the background and serves as an intermediate between multimodal input and active applications running on a computer. To illustrate the concept we present a use case scenario consisting of 4 steps, namely: 1) Login via face detection-recognition, 2) Application selection via object detection-recognition, 3) Authorization control according to login data and 4) application operation, based on steps 1-3, via gesture and speech commands.

The rest of the paper is structured as follows. Then we present our system which uses face, object voice and gesture as interaction means to interact with (a) the operating system and (b) the currently active applications of a computer. Finally, we draw some conclusions and discuss on-going and future work.

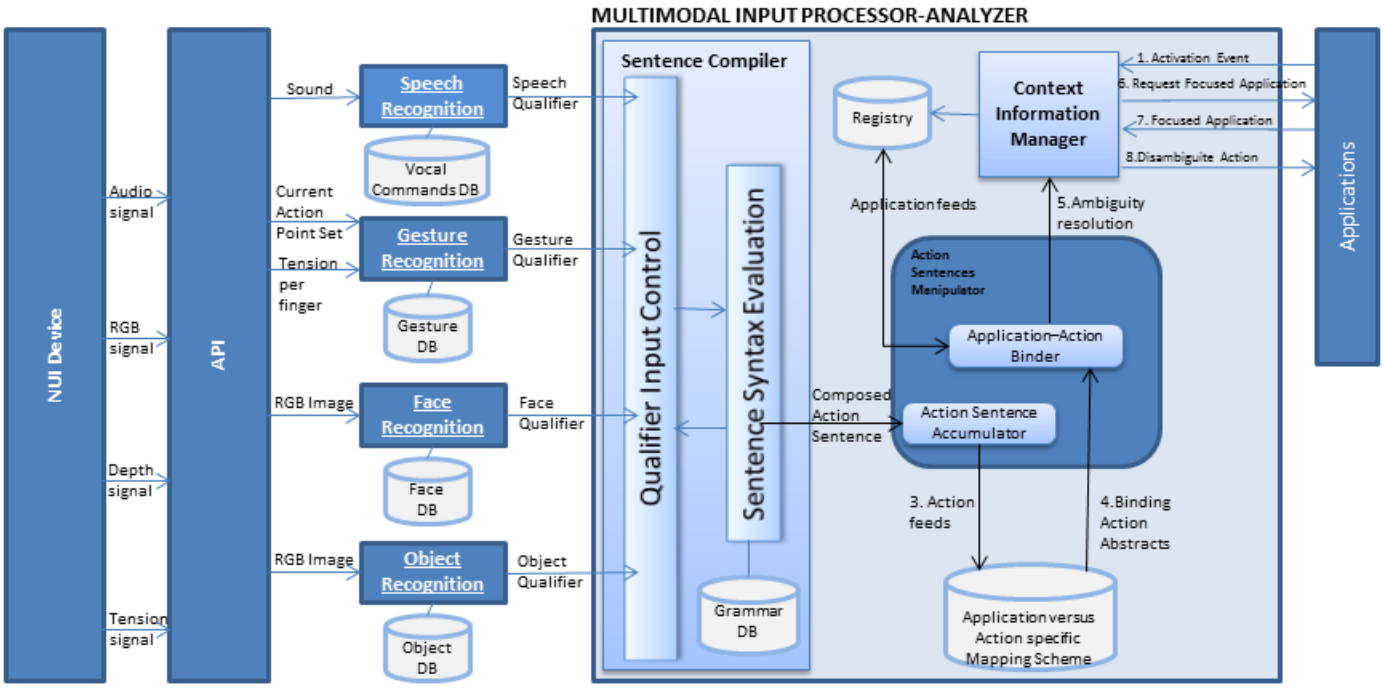

Figure 1: The System Architecture

II. THE SYSTEM ARCHITECTURE

The system we demonstrate on this paper is a multimodal system based on natural user interaction. It integrates the principles of multimodality, as it supports different input devices like RGB Cameras, depth and audio sensors. Each device represents a separate module of our system architecture which can be integrated independently without affecting the rest of the system. This system acts like an intermediate between input devices and operating system.

A. Multimodal Processes

We have integrated four basic processes in order to support multimodal functionality, Face Detection-Recognition, Object Detection-Recognition, Speech Recognition and Gesture Recognition.

One of the main multimodality requirements, in regard to Face Detection-Recognition, is to procure multiple applications for multiple users. A logistical problem occurs in order to provide all user accounts with different rights assigned to each one. This security issue can be easily fixed by assigning passwords for each user account. Although multimodality architecture permits the use of multiple sensors that can be used to upgrade this process in favor both system security and user account management.

Face recognition is the way of optimally using all the provided multimodality advantages mentioned above. The method used in this system is basic but very efficient. The first step towards face recognition is always face detection, making both concepts highly associated.

Face detection is completed in two steps. The first process is to track the joint of the head, provided from Microsoft's sensor. An area is then cropped around it, always by taking

under consideration the depth in which the joint was found. The cropped image does not provide the data for user recognition. This is due to the fact that joints are not always stable, and this leads to step two of the recognition procedure. Now that is certain that the search area is kept to an absolute minimum. A face detection algorithm from [8, 9] is used to extract the recognition data from the cropped image. The combined methods provide absolute accuracy to face data that will be later used for training when adding a new user, or for cross-reference when checking the user identity. The final output is a face image barely resized, at 100x100 pixels for better face recognition performance, and we also apply a histogram equalization to deal with lighting condition variations. The drawback of this method is the vast computational cost when used in real time. We tackle this problem by simply using the face detection method only for a short period of the initial skeleton tracking, which means that a new user was detected and must be identified.

Completing user login procedure demands successful face recognition. Our face recognition methodology is also simplistic and very efficient, demonstrated in experimental results section. It is based in Eigenfaces, firstly introduced in [3, 4], and a classification system based on prediction of similar face images. For successful recognition rates each user can not just be described with one facial image. By using multiple images to each user guarantees its successful identification. The system is trained with five images per user, which include the user looking from different angles to the screen. These images describe the user's face when looking in the center, right, left, bottom and up towards the screen as seen in Figure 2. Although computationally efficient by itself face recognition is executed ten times, because of the former connection with face detection technique. The ten times samples collection is evaluated by the system in order to prevent miss classification of the users, taking a more integrative decision. Downsides of this method are the sensitivity on lighting variations and facial changes. Histogram equalization is not efficacious for every lighting condition, these results to faulty identification under severe lighting changes. Alterations to facial characteristics, is a well-known problem among recognition methods which also affects the results of our current face recognition method and is among our future research agenda to explore new methods that will enable us to tackle the specific problem.

Regarding Object Detection-Recognition, based upon multimodality features we can enrich user interaction with the system. Following this philosophy we introduce a user-object based interaction. Our method implies simple tasks conducted from users by holding different objects in order to control applications.

To achieve this method we have to implement object recognition and classification techniques based on [14, 24, 25]. First of we have to find the location of an object into an image cluttered mostly by unwanted regions. We set user's hands as the main area of investigation for the required objects. Methodology is so similar to the first part of face recognition that was introduced earlier. By using skeleton tracking, we fixate on both user hands and try to identify the objects from the cropped images around them. In this part, it also should be mentioned that depth does not affect the cropping procedure because cropping region is interdependent with depth variations of the user. Objects are not essential to be cropped as a whole because our method is able also to recognize fractions of them.

Ensuring that the produced images contain objects or just a piece of them we carry on with object recognition method. From each cropped image SURF features [11] are extracted which ensure robust object descriptions against rotation and scale variations. For training, the object recognition system uses a database from objects that were corresponded with applications. When classifying an object, the extracted features are categorized with nearest neighbor algorithm [29] providing a number of matches between the examined image and every object class in the database. The final classification result is corresponded to the highest matching object class, providing efficient results as stated in experimental result section.

As a result of Speech Recognition, users are able to communicate with the system via voice commands. A set of vocal commands are pre-defined and stored in the system's vocal command database. The speech recognition algorithms which are running in the background are part of the API of the Microsoft SDK for Kinect. Every spoken word is recorded by the system and compared against the stored words in the vocal command database. Subsequently, if the comparison has positive results, a speech qualifier input is produced and passed to the Sentence Compiler.

Microsoft SDK for Kinect [30] offers speech recognition algorithms which support English, French, Spanish, Italian, and Japanese language. Additionally, Kinect's API offers the ability to develop applications that recognize language spoken in different regions: English/Great Britain, English/Ireland, English/Australia, English/New Zealand, English/Canada, French/France, French/Canada, Italian/Italy, Japanese/Japan, Spanish/Spain, and Spanish/Mexico [30]. Kinect is attempting to recognize speech from a distance through four microphones together with beam forming in order to focus and listen on specific areas of the scene. This way, the device is able to recognize the source of speech and decide which user is talking in a multi-user scene. Additionally, an audio processor is responsible for the multichannel echo cancelation performing, so that confusion of echo phenomenon is avoided.

In the course of the Gesture Recognition process, user gestures are analyzed and managed. A specific set of gesture commands, like vocal commands, are predefined and stored in the system's gesture database. The set of pre-defined gestures that the system recognizes and supports are non-static.

Simple gestures are produced by using right or left arm's wrist joint through Microsoft's skeleton tracking and by calculating the distance between two points. The first point is the current wrist coordinates in x, y, z axis and the second point is where the user's hand will be at the end of the gesture. Thus, the sentence is produced depending on, which axis value is increased during the motion.

In other words, the system focuses on user's hand direction changes. In this manner, when for instance the user's right hand changes direction from left to right our system considers

this movement as a gesture-command. Subsequently, a gesture qualifier is produced and passed to the Sentence Compiler.

B. Multimodal Input Processor-Analyzer

As Figure 1 shows, the Processor-Analyzer module of our system consists of three distinct components that collaborate together, (a) the Sentence Compiler, (b) the Action Sentences Manipulator and (c) the Context Information Manager

The Sentence Compiler component receives the commands that derive from the different input modalities. Every command is a potential part of a sentence. The purpose of the Sentence Compiler is to compile sentences that are composed of these commands. It contains two parts, the Qualifier Input Control and the Sentence Syntax Evaluation. The Qualifier Input Control is responsible for collecting all types of input qualifiers (speech, gesture, face and object). The input reception is parallel and consecutive. The Sentence Syntax Evaluation assumes the burden of checking the words-commands which arrive as input in the Sentence Compiler. If the sentence that is checked is part of at least one of the structured predefined sentences, stored at the grammar database, then this sentence is stored as an incomplete sentence. The system's flow returns back to the Qualifier Input Control in order to wait for the next word-command. When the next command comes as input to the Sentence Compiler, the above process is repeated until a complete sentence is found by the Sentence Syntax Evaluation. If a word-command is not part of a predefined sentence, stored at the grammar database, alone or combined with previous kept word-command then this is rejected by the system.

TABLE I. STEP1: DESKTOP LOGIN VIA FACE DETECTION-RECOGNITION

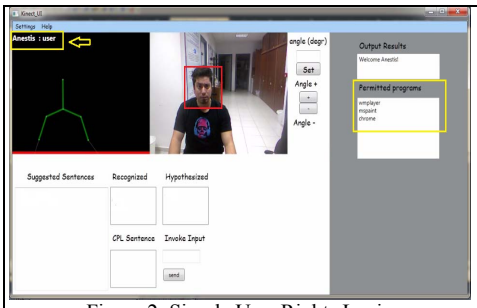

Figure 2: Simple User Rights Login

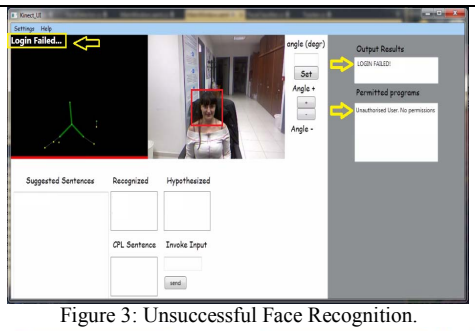

Figure 3: Unsuccessful Face Recognition.

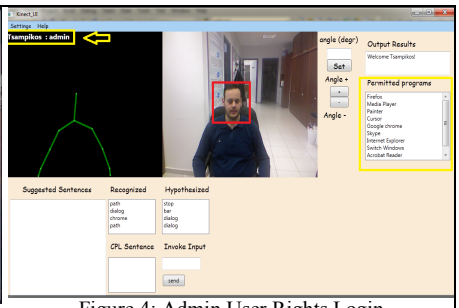

Figure 4: Admin User Rights Login

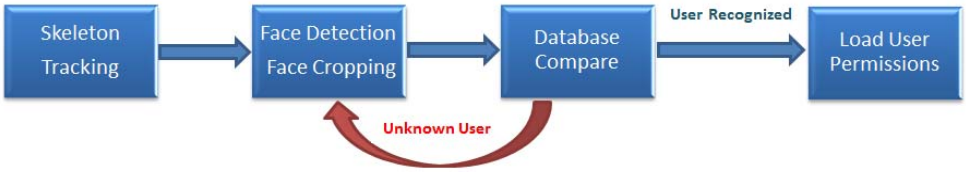

Figure 5: Desktop Login Flow Diagram

The Action Sentence Manipulator collects all the composed sentences that come out as an outcome form the Sentence Compiler described above. The purpose of this module is to handle the sentences as actions that should be executed on an application. It contains two parts, the Action Sentence Accumulator and the Application-Action Binder. The Action Sentence Accumulator compares the sentence received from the sentence compiler, with the information stored in an Action Sentence-Application database, concerning the ability of applications to respond to sentence actions. The outcome of this comparison is passed on to the Application-Action Binder component. The Application-Action Binder component, having on one hand the information about which application/s are capable of handling the action sentence and on the other hand which application/s are up and running, informs the "Context Information Manager" which in turn passes the action sentence to the corresponding application/s.

The Context Information Manager (CIM) is responsible to communicate with the operating system. It retrieves information about the applications running on the system. More specifically, CIM interest lies on the state of the application (focused or unfocused). This is necessary as applications not in focus, are not capable of executing an action. For this reason CIM updates continuously a list with all running applications and there focus state.

III. EXPERIMENTAL RESULTS

Our example application uses Microsoft Kinect as a multimodal input device that provides Audio, RGB and Depth Image. We use Microsoft's official SDK for Kinect to process the incoming information in order to get the user's skeleton joints as our action points and the user's voice as input sound. The action points' position is refreshed at the rate of 30 times per second, which is Kinect's frame rate for skeleton tracking. By action points we mean the user's detected body joints, by the Microsoft's skeleton tracking mechanism, which are then used for face, object and gesture recognition. Our gesture recognition algorithms, apart from tracking gestures such as push-in, pull-out, swing left, right, up and down, also detect stable poses of the user and checks possible changes of action points position every 10 frames. Sound is processed in real-time with the use of Microsoft Speech Recognition SDK, which searches for a match within our Vocal Command XML

database. The database consists of commands and corresponding speech qualifiers.

In the next paragraphs we present a use case scenario for multimodal desktop application interaction consisting of four distinct steps: step1) Desktop login using user's face detection-recognition, step2) application selection via object detection-recognition, step3) User application authorization control and application launch and step4) application operation via gesture and speech commands according to steps 1-3.

The first step of our use case scenario deals with desktop login through user's face detection-recognition. To detect the user has the choice either of standing (default position) or sitting (seated position) in front of the Kinect device, so that the skeleton tracking procedure is activated. After the user's skeleton is detected (see upper left window of Figure 2, Figure 3, and Figure 4), our system continues with cropping the RGB image at head's joint coordinates, creating a rectangle around the face (see middle window of Figure 2, Figure 3 and Figure 4). The cropped image is then compared with a set of users' images which are stored at the face database of the system. Depending on the result of the comparison the system either returns to face detection-recognition (see

Figure 5) if user is unknown with the appropriate failure message (see upper right info area called Output Result as well as message on the skeleton tracking window of Figure 2, Figure 3 and Figure 4) or proceeds and loads the user permission i.e.: user type and name (see message on the skeleton tracking window of Figure 2 and Figure 4), which can be "admin" having full permissions or "user" with limited permissions and list of applications that can be used (see upper right info area called Permitted Programs of Figure 2, Figure 3 and Figure 4). Furthermore the user name is passed to the Sentence Compiler through the Qualifier Input Control for participating at the creation of the command sentence.

TABLE II. STEP2: APPLICATION SELECTION VIA OBJECT DETECTION-RECOGNITION

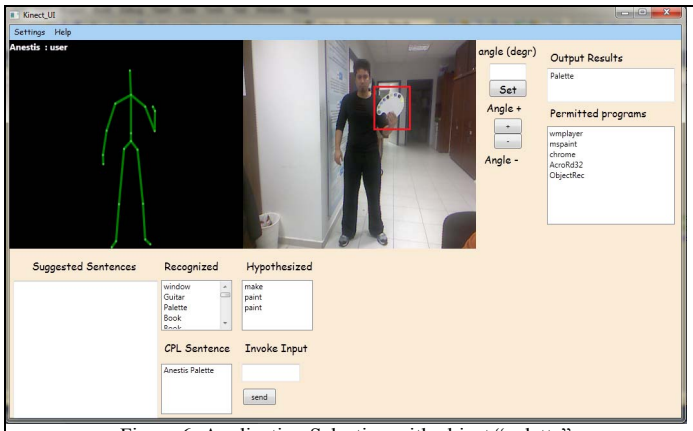

Figure 6: Application Selection with object "palette"

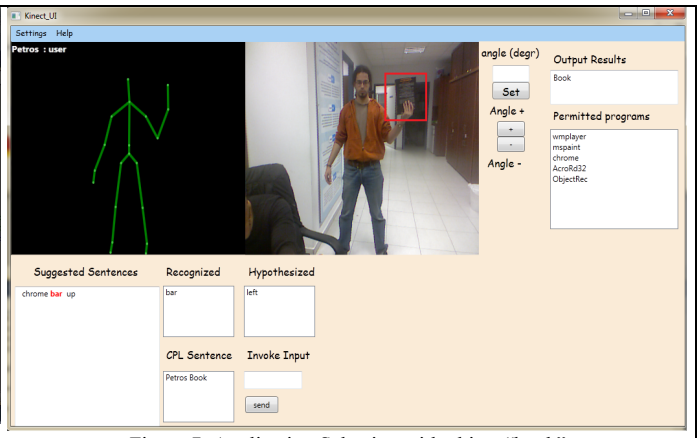

Figure 7: Application Selection with object "book"

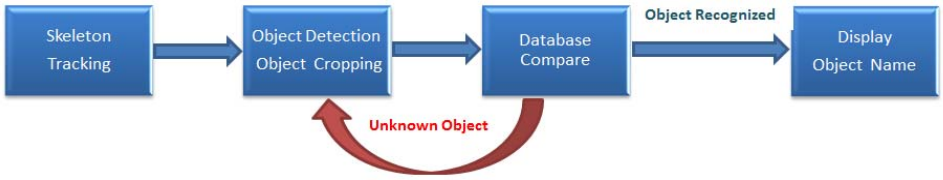

Figure 8: Application Selection Flow Diagram

The second step (see Table II) deals with the application selection through object detection-recognition. To accomplish that, once again the user must be in front of the Kinect device, in order for the skeleton tracking to start and as soon as they are tracked, the cropping of the RGB image at the right hand's joint coordinates is ready to begin, creating a red rectangle around the object (see RGB image in Figure 6 and Figure 7). The cropped image is then compared with a set of objects' images which are already stored in our database. The result of successful object recognition is the name of the recognized object (see Output Results field in Figure 6 and Figure 7). Successful object recognition results are sent to the Sentence Compiler through the Qualifier Input Control which in turn compiles a correct sentence, if any, using the results from step1 and step2 (e.g. "Anestis Palette" at Figure 6 or "Petros Book" at Figure 7). Upon successful completion of steps 1&2 the system launches the appropriate application (e.g. if recognized object is "palette" start MS-Paint or if recognized object is "book" start Google Chrome). If there is no result from the object recognition process the procedure is repeated until the object is recognized by the system, as we can see at the flow diagram in Figure 8.

At step 3 the system compares user permissions, which is data acquired from step1 and application information, which is data acquired from step2. More specifically, the system loads a list of applications that the user is permitted to operate (step1) and the application information connected with the object hold by the user (step2). If there is a match between permitted applications and application defied by object then the application is launched. The above described process can be

codified in the following steps: step3.a: Read User Permissions, step3.b: Read Applications connected with Object, step3.c: Compare data from 3a and 3b and step3.d: Launch application and proceed to step 4 or display appropriate message.

Once the user is successfully logged in the system and the application is launched he/she can proceed to step 4 (see Table III) i.e. operate the application via gesture and voice commands. In our use case scenario, one application per login, namely MS-Paint and Google Chrome, is initially launched depending on the user ("Anestis Palette" for the MS-Paint and "Petros Book" for the Google Chrome) that has logged in. The MS-Paint operation scenario is demonstrated in Figure 9 while the Google Chrome operation scenario is demonstrated in Figure 10.

TABLE III. STEP4: APPLICATION OPERATION VIA GESTURE AND SPEECH COMMANDS

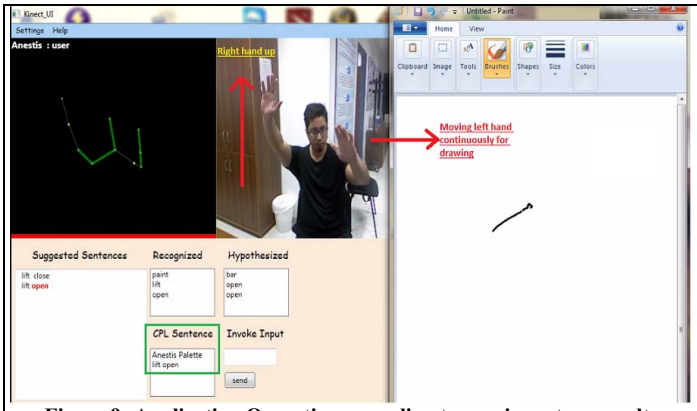

Figure 9: Application Operation according to previous steps results "Anestis Palette"

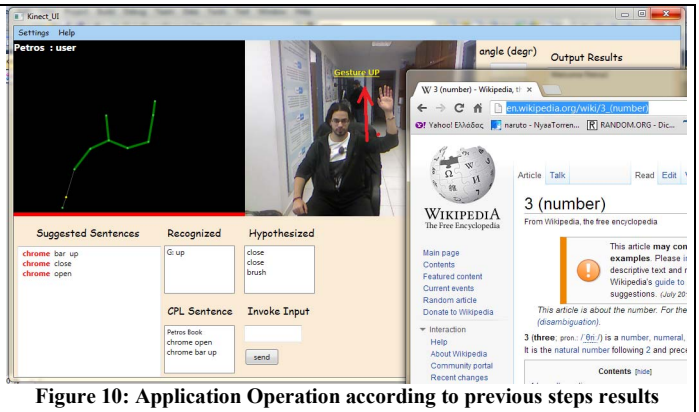

Figure 10: Application Operation according to previous steps results "Petros Book"

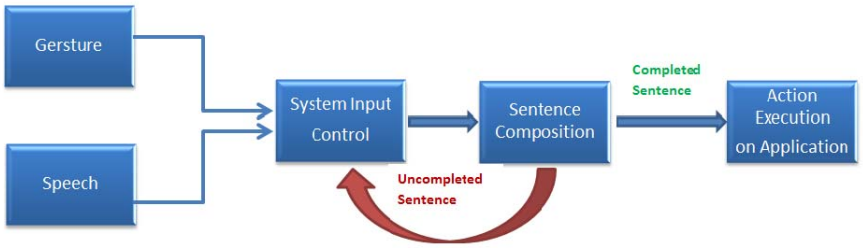

Figure 11: Application Operation according to Gestures and Voice commands Flow Diagram

Regarding the MS-Paint operation scenario, it is implemented with two complex action sentences. The first one consists of two voice commands as shown in the left lower part of the image in Figure 9. The firsts command of this sentence is a vocal one (Lift) which is sent to Sentence Compiler as Speech Qualifier. The Compiler checks the sentence syntax, recognize the qualifier in its library but needs more input to compose a complete sentence, so the qualifier input control waits for another input. After that the system detects another vocal command (open, which is translated into "Brush manipulation "), and sends a Gesture Qualifier to the Sentence Compiler. The Compiler checks the sentence syntax, recognizes the qualifier in its library and sends the two word sentence to the Sentence Syntax Evaluation component which compares the two commands with the data stored in the Grammar database and finds a match. The composed sentence is sent to the ASM. The ASM in his turn, checks that the launched and focused application (in our scenario the MS-Paint) can use this "Lift Open" action sentence. If the focused application can use the action sentence it disambiguates the sentence and sends the command to be executed to the application. Since MS-Paint application can respond to the specific command "Lift Open", the action sentence is executed and user takes the control of the default tool of the application.

The second sentence action sentence of this use case scenario consists of two gesture command. This sentence is about the drawing part of this example. The first gesture is "up" expressed by the user's right hand as shown in Figure 9. This gesture is translated to the start of the drawing and it can be compared to the action of mouse touching. When user lowers his hand below his head then the drawing procedure stops as it would be if user leaves his hand from mouse device. The second part of this sentence is another gesture command which is expressed by user's left hand. Essentially, this is a continuous gesture in order to manipulate the brush of MS-Paint application according to the movement of the user's left hand as shown in Figure 9. As explained above, if user's right hand is lowered below the head position ("down" gesture) then the manipulation of MS-Paint ends.

Regarding the Google Chrome operation scenario, in the left lower part of the image of Figure 10 we show that our system detects a vocal command (Google Chrome) and sends the Speech Qualifier to the Sentence Compiler. The Compiler checks the sentence syntax, recognize the qualifier in its library

but needs more input to compose a complete sentence, so the qualifier input control waits for another input. Subsequently, another vocal command (BAR) is detected by the system. This command is sent as Speech Qualifier to the Sentence Compiler. The compiler recognizes this second vocal command, which combined with the previous one contributes to the completion of a sentence. After that the system detects a gesture command (UP, see middle RGB image of Figure 10), and sends a Gesture Qualifier to the Sentence Compiler. The Compiler checks the sentence syntax, recognizes the qualifier in its library and sends the third word sentence to the Sentence Syntax Evaluation component which compares the three commands with the data stored in the Grammar database and finds a match. The composed sentence is sent to the ASM. The ASM in his turn, checks that the launched and focused application (in our scenario the Google Chrome) can use this "Chrome-Bar-Up" action sentence. If the focused application can use the action sentence it disambiguates the sentence and sends the command to be executed to the application. Since Google Chrome application can respond to the specific command "Chrome-Bar-Up", the action sentence is executed and the url bar of the browser is focused. (see right image of Figure 10)

IV. CONCLUSION & FUTURE WORK

This paper has described a system for multimodal natural interaction of users with computers using a multimodal input device, the MS-Kinect, as the main input source of RGB, depth and audio signals which are considered as different modals. System's aim is to provide users the means to become acquainted with natural user interfaces and multimodality.

From the perspective of multimodal desktop interaction, we presented our system that allows users (a) to login to the desktop via the natural means of face through face detection and recognition, (b) activate a desktop application via a non-natural modality namely an object, through object detection and recognition and (c) operate a desktop application with the natural means of gestures and speech. In order to illustrate the concept of our approach, we presented a use case scenario which consist of 4 distinct steps according to the above presented procedure

In conclusion, regarding future evolution could exploit (a) different types of input modalities such as pressure or proximity sensors, (b) more sophisticated techniques and algorithms for facial and object detection-recognition, (c) grammar databases to enrich the ability of the system to respond to complex sentences and (d) exploit it's usage in the field of robotics in order to accomplish robot navigation by natural means.

REFERENCES

[1] ^ Cover TM, Hart PE (1967). "Nearest neighbor pattern classification". IEEE Transactions on Information Theory 13 (1): 21–27.

[2] Bolt, R.A. Put that there: Voice and gesture at the graphics interface. *ACM Computer Graphics 14*, 3 1980.

[3] William Buxton, There's more to interaction than meets the eye:some issues in manual input. In: User Centered System Design, D. A.Norman and S. W. Draper (Eds.), Lawrence Erlbaum, 1986, 319-337. 1986.

[4] M. Turk and A. Pentland (1991). "Face recognition using eigenfaces". Proc. IEEE Conference on Computer Vision and Pattern Recognition. pp. 586–591.

[5] M. Turk and A. Pentland (1991). "Eigenfaces for recognition". Journal of Cognitive Neuroscience 3 (1): 71–86

[6] Oviatt, S. Multimodal Interactive Maps: Designing for Human Performance. *Human-Computer Interaction*. Volume 12, pp. 93-129, 1997.

[7] Oviatt, S.L. Ten Myths of Multimodal Interaction,Nov. 1999.

[8] Paul Viola and Michael J. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. IEEE CVPR, 2001.

[9] Rainer Lienhart and Jochen Maydt. An Extended Set of Haar-like Features for Rapid Object Detection. IEEE ICIP 2002, Vol. 1, pp. 900-903, Sep. 2002

[10] Mitra, S. and Acharya, T. 2007. Gesture recognition: A survey. IEEE Trans. Syst. Man, Cybernet. C: Appl. Rev. 37, 3, 311–324.

[11] Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346--359, 2008

[12] Chang,Y.-J,etal.AKinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. Research in Developmental Disabilities (2011)

[13] Weinland, D., Ronfard, R., and Boyer, E. 2011. A survey of vision-based methods for action representation, segmentation and recognition. Comput. Vis. Image Understand. 115, 2, 224–241.

[14] T. Kounalakis and G. Triantafyllidis, "Object detection and recognition using depth layers and SIFT-based machine learning" , 3D Research Journal Spinger 2011

[15] Suma, E.A.; Lange, B.; Rizzo, A.; Krum, D.M.; Bolas, M.; , "FAAST: The Flexible Action and Articulated Skeleton Toolkit," Virtual Reality Conference (VR), 2011 IEEE , vol., no., pp.247-248, 19-23 March 2011

[16] Stowers, J.; Hayes, M.; Bainbridge-Smith, A.; , "Altitude control of a quadrotor helicopter using depth map from Microsoft Kinect sensor," Mechatronics (ICM), 2011 IEEE International Conference on , vol., no., pp.358-362, 13-15 April 2011

[17] Gallo, L.; Placitelli, A.P.; Ciampi, M.; , "Controller-free exploration of medical image data: Experiencing the Kinect," Computer-Based Medical Systems (CBMS), 2011 24th International Symposium on , vol., no., pp.1-6, 27-30 June 2011

[18] Carrino, Stefano; Mugellini, Elena; Khaled, Omar Abou; Ingold, Rolf; , "Gesture-based hybrid approach for HCI in ambient intelligent environmments," Fuzzy Systems (FUZZ), 2011 IEEE International Conference on , vol., no., pp.86-93, 27-30 June 2011

[19] Van den Bergh, M.; Carton, D.; De Nijs, R.; Mitsou, N.; Landsiedel, C.; Kuehnlenz, K.; Wollherr, D.; Van Gool, L.; Buss, M.; , "Real-time 3D hand gesture interaction with a robot for understanding directions from humans," RO-MAN, 2011 IEEE , vol., no., pp.357-362, July 31 2011-Aug. 3 2011

[20] Marti A. Hearst; 'Natural' Search User Interfaces, Nov. 2011.

[21] Yale Song, et al., Continuous body and hand gesture recognition for natural human-computer interaction, ACM, Transactions on Interactive Intelligent Systems(TiiS),vol 2,issue 1, March 2012

[22] Ayrton, O.; Steven K.; Burkhard C. W.; Bruce M.; "Using the Kinect as a Navigation Sensor for Mobile Robotics", Robotics, IVCNZ '12 Proceedings of the 27th Conference on Image and Vision Computing New Zealand, Nov. 26 2012

[23] Hubert P. H.; Edmond S. L.; "Real-time Physical Modelling of Character Movements with Microsoft Kinect", VRST '12 Proceedings of the 18th ACM symposium on Virtual reality software and technology, Dec. 2012.

[24] M. Demetriou, T. Kounalakis, N. Vidakis and G. Triantafyllidis, "Fast 3d scene object detection and real size estimation using microsoft kinect sensor",13th IASTED International Conference on Computer Graphics and Imaging, Heraklion, Greece, 2012.

[25] G.Kalliatakis,T.Kounalakis, G. Papadourakis and G. Triantafyllidis, "Image-based touristic monument classification using graph based visual saliency and scale-invariant feature transform",13th IASTED

International Conference on Computer Graphics and Imaging, Heraklion, Greece, 2012.

[26] OpenCV Computer Vision, http://opencv.willowgarage.com/wiki/

[27] Microsoft Kinect, http://www.microsoft.com/en-us/kinectforwindows/

[28] Asus Xtion PRO LIVE, http://www.asus.com/Multimedia/Motion_Sensor/Xtion_PRO_LIVE/

[29] Vidakis, N.; Syntychakis, M.; Triantafyllidis, G.; Akoumianakis, D., "Multimodal natural user interaction for multiple applications: The gesture — Voice example," Telecommunications and Multimedia (TEMU), 2012 International Conference on , vol., no., pp.208,213, July 30 2012-Aug. 1 2012

[30] http://research.microsoft.com/en-us/collaboration/focus/nui/kinect-windows.aspx