TeachMe, a Programming by Example Customizable Gesture Recognition System

Hazem Kathem Qattous

hqattous@asu.edu.jo

Department of Computer Information Systems
Applied Science Private University
Amman, Jordan.

Summary

Much research is conducted to provide lecturers and presenters the naturalness and intuitiveness in using presentation tools. Presentation tools are controlled traditionally using mouse and keyboard. These required features can be provided to presenters through allowing them to control presentation tools using gesture recognition systems. Most introduced such gesture recognition systems require the user to learn their gesture vocabulary as the set of gestures that control the presentation are hardcoded in the system at implementation time. This problem prevents users from customizing gestures according to their preference and limits the users with the gestures and actions available and hardcoded in the system without being able to add new gestures. A third problem is that most gesture recognition systems require the user to use their both hands which leaves no chance for special needs or amputated users to use such systems. In this paper, TeachMe, a gesture recognition system that uses Microsoft Kinect as an interface to control MS PowerPoint presentation and mouse pointer is introduced. The paper presents a TeachMe implementation that adopts the notion of programming by example for the purpose of gesture-action customization. The idea of gesture customization feature implemented in TeachMe is available in only few documented gesture recognition systems, which indicates the very little research conducted in this area. Additionally, it is the first time that programming by example notion is adopted in the field of gesture recognition systems. The idea of gesture customization and its application using programming by example to solve the problems addressed above is considered the contribution of this research.

Key words

Programming by Example, Microsoft Kinect®. Gesture recognition system. Gesture customization.

1 Introduction

Presenters such as lecturers at classrooms conduct their presentations using presentation tools such as MS PowerPoint and Prezi. These tools help the presenters to control the presentation flow and help the audience to focus on specific points of the presentation [1]. During presentations, users of presentation tools are required to be beside the computer machine all the time to use the mouse and keyboard for controlling the presentation. This limits their movement during the presentation and leads to reduce the eye contact between the presenter and the audience [2] and [3]. Using traditional techniques, mouse and keyboard, for controlling presentation also reduces the interactive intuitiveness and naturalness in communication with audiences and with machine [4]. Research that lists the problems above such as [5], introduces gesture-capturing devices as a solution where the communication between the human and the presentation tool will be mediated by a gesture recognition system instead of the traditional techniques, mouse and keyboard. Gesture recognition systems presentation as a solution is supported by their intuitiveness, naturalness and efficiency they provide in communication between human and machine [6] and [4].

In gesture recognition systems, the user provides a gesture such as swinging left hand from left to right, the system recognizes it and maps it to an action (command) such as moving a slideshow one slide from left to right. However, such solution adds more complications on the burden of the user, that is, learning the gesture vocabulary of the gesture recognition system [7]. This vocabulary is required to control the presentation and it is hardcoded in the system [8]. Meaning that in the previous example, the user should know that swinging left hand from left to right would lead to the action of moving a presentation slide from left to right. Although the example looks intuitive, [9] introduced that gestures vocabulary has no standards or global agreement on specific gestures to mean specific actions or commands. Accordingly, this gesture-action mapping (vocabulary) should be taught to the user before being able to use a specific gesture recognition system.

The problem of hardcoded vocabulary that needs to be learned is divided into three problems that this research is addressing. The first is that the user, in many cases, may like to customize the implemented gesture-action mapping.

Manuscript received December 5, 2016
Manuscript revised December 20, 2016

The user will be motivated with his/her preference for a specific gesture to perform specific action. Secondly, the user may need to add some gestures and actions that are not implemented already in the system. This problem was discussed by [9] as they documented the importance of customizing an existing gesture and the necessity of creating new user own gestures. A question may arise of the feasibility of allowing users to invent new gestures. There will be always a question about their ability for creating new gestures. The question was answered by [10] who asked participants to create new gestures for car-driving commands and concluded that they found it easy to accomplish the job. This encourages the work on customizing gestures problem. The third problem is that the implemented system with predefined gestures and mapped actions will be directed to specific users, not disabled users as an example. Let us take the above mentioned example, moving a slide from left to right using swinging the left hand from left to right. A left hand amputated person would not be able to do this gesture, and accordingly, would not be able to use the system.

Little research focused on such limitations and problems exist in the gesture recognition systems introduced as replacement for the traditional techniques in controlling presentations and mouse pointer. This research introduces TeachMe, a gesture recognition prototype that is able to control MS PowerPoint presentation and the mouse pointer. An initial TeachMe implementation with a customization feature has been introduced by [5] to find a solution for the customization problem according to user preference. That is the first problem introduced in this research. However, the introduced customization feature has a significant limitation because it provides the user with the ability to customize limited number of gestures that start a presentation, end a presentation, move slideshow forward and move slideshow backward. It does not provide the ability to add new gestures and does not take into consideration the special needs users.

This research introduces Programming by Example (PBE) (also called Programming by Demonstration (PBD)) as a solution for customizing gestures, which will be also a solution for the above listed problems. PBE is a programming technique that depends on introducing examples to the system. The system generalizes these examples and generates a program [11]. It has been introduced as helper for non-programmers to make programming easier for them. According to [12], people can express themselves and express what they want through examples better than using programming language script. This paper describes TeachMe adoption of PBE notion to implement an advanced customization feature. The customization feature implementation is able to take an example from the user in form of a gesture the user needs, prefers or would like to add as new gesture. User also provides the system with the required action to be mapped to the provided gesture.

Next section describes TeachMe gesture recognition system with the focus on its new implemented PBE-based customization feature. A simple example is introduced to clarify the work of the new feature. Related work in the field of programming by example and gesture recognition systems with comparisons to TeachMe features and implementation is included as the following section. Limitations and future work thought of, are listed at the end of the paper. Finally, a conclusion that summarizes the paper is presented.

2 TeachMe Gesture Recognition System

TeachMe is a prototype that was developed by graduate students as their graduation project at Faculty of Information Technology / Applied Science Private University. The project development continued by the researcher and the help of two more students to produce the current implementation. TeachMe works with Microsoft Kinect, which is designed to capture human body gestures for running games. TeachMe uses Kinect to capture gestures for the purpose of controlling MS PowerPoint presentation and mouse pointer. Its implementation depends on SDK 1.8 library. Analysis of captured gestures takes the form of comparing a gesture using its extracted features with existing gestures' features. The features of a gesture depend on three points of human body, the two hands and a head, in addition to time and distance factors.

When controlling a PowerPoint presentation, the system recognizes a gesture by capturing the movement of the three points, mainly the hands points, in reference to each other and in reference to the head point. As an example, if the user moves his/her left hand from left side to right side the system captures this movement and extracts that the left hand point passed the head reference point from left to right. The system also captures the time required to do this gesture and the distance that the left hand point moved from left to right. Time required to perform the gesture in addition to the direction and reference point features specify if the system recognizes the gesture or not. As an example, to move a slide forward, the user needs to move right hand from right to left. However, if the user moves his/her right hand from right to left and returned it back very quickly, the system my not recognize the gesture even if it captures the movement. This is because the gesture does not satisfy the full set of features required for the system to recognize the gesture. For the same example, if the user moves his/her right hand from right to left, but with a short distance gesture, the system will not recognize the required gesture. There are time and distance thresholds set as features to recognize gesture. In the last

case, the hand point should have minimum distance from the head point in the x-axis. For more details about TeachMe initial implementation, please refer to [5].
With this special combination and thresholds of gesture extracted features, the system compares the gesture with gestures maintained in the system, and finally, performs the action mapped to the gesture, if found. In this case, the action will be moving the running slideshow a slide backward. The user can perform four PowerPoint controlling actions using gestures; these are, starting a slideshow, moving a slide forward, moving a slide backward, and ending a slideshow. TeachMe was also implemented to control mouse pointer, which is a feature exists in a very little number of systems documented in the reviewed research papers. Controlling mouse pointer was implemented to follow the right hand movement. Actions that the user can perform using mouse pointer controlling feature include open a folder using double click, right click anywhere in the desktop and "drag and drop" items on desktop. The user can use gesture to drag a folder and drop it in the Recycle Bin as an example. Controlling mouse pointer movement depends on capturing the z-axis in addition to the x and y-axis.
In the previous TeachMe implementation, a simple gesture customization feature has been implemented. The customization depends on that the user can select between the four actions available for controlling a PowerPoint slideshow, start a presentation, move slide forward, move slide backward and end a presentation. The user then can select the required gesture to perform the selected action through a dropdown menu [5]. However, customization feature with its previous implementation is not flexible as it provides the user with fixed number of choices. The user cannot customize a gesture according to his/her preference if this "preference" is not between the choices available for customization. Hence, this research tries to get a step forward in solving the problem of gesture customization problem. Programming by example technique, which depends on introducing an example to the system and the system learns from it, has been adopted as a solution in this research. Programming by example has been applied in different domains but not yet in the domain of controlling presentation through Kinect or the domain of gesture customization. This is considered as the main contribution of this research. [13] introduced the idea of using programming by example as they state that Kinect can be implemented using programming by example technique but this implementation is the most difficult between others for gesture recognition.
The best way of describing the programming by example technique adopted in TeachMe is through an example. If the user would like to move a slideshow forward, that is moving a slide from right to left, the user should move their right hand side from right to left. This is the default implementation. Suppose that the user wants to customize the gesture of this action. S/he wants to move a slideshow one slide forward using left hand where the user moves their left hand to the right first then get it back to the left. The process starts with capturing the gesture by Kinect. The second step is analyzing the gesture and checking the combination of features that represent a gesture. This is a feature extraction step. Third step is to pass the information to a dictionary data-structure that maps the features, if exist, to an action. Once the set of features found as a key in the dictionary, the required action is identified as the value and passed for execution. In this case, the system will not find any matching gesture in the dictionary. However, the system does not know that it should learn this gesture instead of trying to find a mapped action to execute. This was the first problem faced the implementation of this research. That is, how can the system know that the user is teaching it instead of giving it a gesture to recognize?

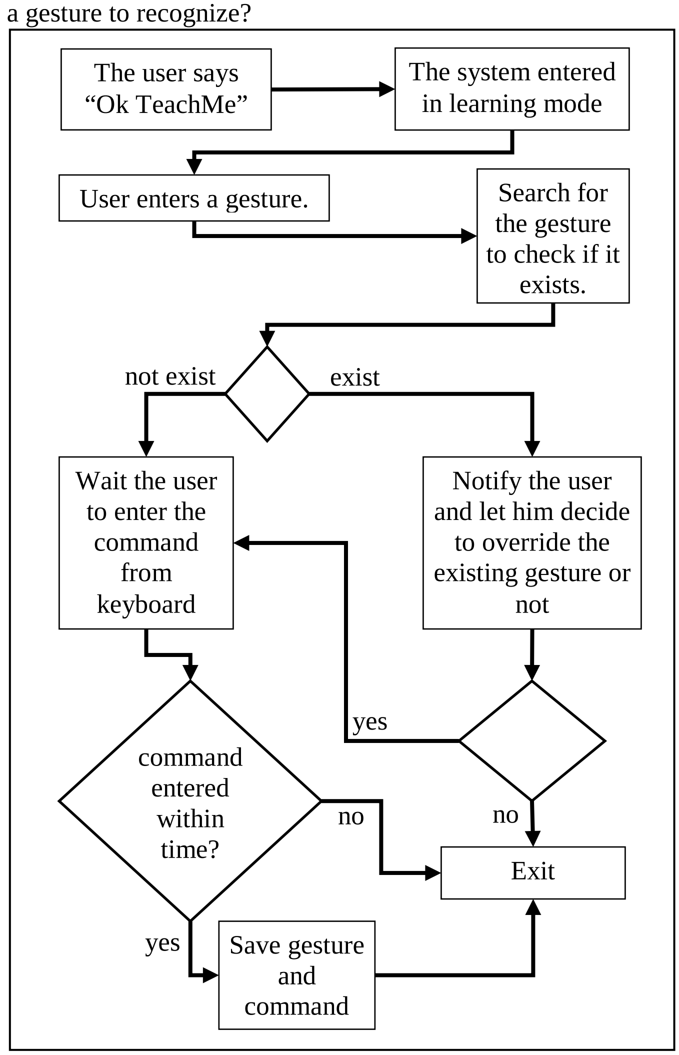

Fig. 1 Gesture Customization Process

To solve this implementation problem, there was a need to trigger the system to enter into a learning mode. For this purpose, a sound recognition feature has been implemented into TeachMe. The implementation used Kinect sound recognition library available in the SDK 1.8. However, there was a need for keyword that switch the system from recognition mode to learning mode. Researcher invented “Ok TeachMe” keyword inspired by “Ok Google” available in Google search engine. Once the user says “Ok TeachMe”, the system enters a learning mode and starts waiting the user to provide a gesture. At this point, the user performs the required (customized or new) gesture. As the system is in learning mode, it captures the gesture, analyzes it by recording and extracting all of its features that are required to recognize it later on. These features are saved. The system now knows the required gesture but does not know the required action to perform when the gesture is recognized. Hence, the user needs to provide the required action. This is done through the keyboard.

The user presses on the keyboard to provide the required action to be associated or mapped to the provided gesture. In this case, the user needs to know the keyboard shortcut for the required action. As an example, if the user provided a gesture and wants that gesture to be mapped to an action of ending a slideshow. In this case, the user should know that “Esc” key in the keyboard ends the slideshow. To continue the example introduced above, the user wants to make the slideshow moves one slide forward. The correct key in this case is the left arrow key. Once the user presses the left arrow key, the system saves the captured gesture in the dictionary data structure as a key and the left arrow key as a value. When the user close the system, the dictionary data structure is saved in a simple text file to be retrieved when started again. Basically, TeachMe implementation moved the hardcoded gestures and mapped action to a dynamic runtime dictionary data structure which facilitates gesture customization. When the user performs the new gesture again, the system will recognize it and performs the mapped action. Fig 1 shows the process of customizing a gesture.

In programming by example notion, TeachMe implementation can be encapsulated as following. The user first provides an example gesture, of course after ordering the system to enter the learning mode by voice. Then the user provides the required action. The system generated the required code, which is the gesture features and its associated action in this case. It can be noticed that TeachMe learns by generating the required code from only one example. This is because of the domain nature of gestures recognition. In many programming by example concept applications, either the system should generalizes the example(s) or the user needs to provide more than one example. Providing more examples allows the system to generalize from them instead of specifying. However, in TeachMe, generalization is not needed as one example is enough to generate the required code. A problem arisen during implementation of learning mode is, what if the user entered the learning mode through saying the magic words “Ok TeachMe” then changed his/her mind regarding adding or customizing a gesture? In this case, TeachMe waits for a time period, 5 seconds, and then the system exits learning mode assuming that the user is not going to continue in customizing a gesture process. Another issue is that when the user provides the example gesture and then changed his/her mind before providing the required associated action through the keyboard. A time period is also set in this case for 5 seconds. After that, the system deletes the provided gesture and exits learning mode.

From the above description, it is clear that using PBE customization feature, it is possible to either customize a gesture or add a new gesture that has not been defined before. In other words, the customization feature allows creating new gestures that have not existed before. When the user provides a gesture during the learning process, this gesture may exist or not exist at all in the dictionary. The two cases are handled as following:

- If the gesture does not exist, there will be no problem as the gesture will be inserted as a new one and mapped later to the required action. In a situation like this, if the action exists already, then it will be mapped to another gesture. The problem in this case is that the same action is mapped to two gestures, the old one and the customized (new) one. The old gesture with the same action will not be deleted or removed from the dictionary as the system works by searching the features of the gesture instead of the action. Meaning that there will be two gestures perform the same action. This case never caused a problem during subjective testing of the system throughout implementation.
- The second case is that the gesture exists already in the list of gestures that can be recognized by the system. In other words, the gesture exists in the dictionary data structure and mapped already to an action. This case creates problems because if the system behaves normally, the gesture will be saved and the current action will be overridden by the new entered action. Meaning that the old action will be deleted and there is a possibility that the old action is not mapped to any other gesture. This leads to losing the old action. To reduce the effect of such problem, when the user is going to override an action with a gesture that already exists, the system provides the user back with a dialog box to let him/her decide what to do. The system tells the user that the provided gesture exists and shows the mapped action as the ASCII code of the keyboard key. It also gives the

user the control to continue customizing the gesture or to cancel the request.

The above two described cases, also appear in Fig 1, summarize the difference between creating a new gesture and customizing one. When the user requires to add a new gesture, then this is the first case where the gesture does not exist before in the system. On the other hand, when the user wants to customized an existing gesture to respond with another action as a preference, then this is the second case described above. The system in this case, tells the user that the old action will be overridden.

3 Literature Review and Related Work

3.1 Programming by example

Programming by Example (PBE) or Programming by Demonstration (PBD) is a technique of introducing examples and values from which the system generalizes and generates a program [11]. A motivation for applying programming by example is that people can express themselves by providing examples of what they want to perform better than expressing themselves using textual programs [12]. PBE has different application domains with different abilities and requirements to generalize examples depending on the domain. One of the famous application of PBE is the macros in Microsoft office. In MS Excel implementation, as an example, PBE technique does not generalize at all. It is the responsibility of the user to complete the generalization through direct code implementation.

[11] introduced Peridot, a system that adopts programming by example technique for GUI constraints definition. In the same domain of GUI constraints definition, “Chimera” [14] and Druid [15] were also introduced. aCAPpella system [16], is another adoption of programming by example technique in the field of context-aware applications. This includes smart home systems with their required inputs in form of sensors. aCAPpella requires the user to intervene for modifying the inference generated from its embedded machine learning technique. The intervention here can be considered as a help provided by the user to tune and refine the system inference and generalization.

Programming by example was also introduced in the fields of game code generation as in case of Gamut [17] and Kidsim [18]. The technique is also applied in the field of documentation and text editing as in DocWizard [19], SMARTedit [20] and CHINLE [21] and also in the field of robotics such as in [22] and [23]. PBE was also applied in the field of spreadsheets as in case of Gencel that provided with spreadsheets examples and infers templates [24]. Another domain of PBE application is the web sites as [25] and [26] introduced using PBE technique in user interface enhancing and creating versions of websites that are suitable to be displaying on mobile screens.

This shows that PBE is applied into wide range and diverse domains. For the domain of this research, the only found documentation of PBE notion application with Kinect is introduced by [13] as one of different methods of Kinect programming techniques.

3.2 Gesture recognition and gesture customization

Several research have introduced gesture recognition systems in the field of education and used Kinect as an interface for capturing gestures such as [27], [3] and [2] who introduced Ki-Prez gesture recognition system. Much research also lists the importance of gesture recognition systems in controlling presentation tools such as [28] that evaluated a gesture recognition system and concluded that such a system is a supportive tool in delivering simple information. The gesture recognition system “Presenter” that uses Kinect for controlling MS PowerPoint is introduced by [1]. The difference between “Presenter” and TeachMe is that the last can control the mouse pointer and has the required vocabulary (gestures) to start and end a presentation. On the contrast, [29] introduced a gesture system that controls the mouse pointer but not a presentation because its intended application field was not the education. A gesture prototype that is similar to TeachMe in the purpose of controlling the mouse pointer is introduced by [30]. The system, through gestures, can control the mouse pointer to deal with graphical user interfaces on the desktop such as windows, icons and menus. Controlling a presentation is out of the scope of their prototype.

[9] present uWave that is an algorithm for gesture recognition. Similar to TeachMe, uWave can customize gesture according to the user preference. They call customization as “personalization”. Similarity between the two systems extends to include the ability to create new gestures. uWave needs only one training sample to personalize a gesture or create new gesture. This is exactly similar to TeachMe. However, they do not put their algorithm in the notion of programming by example. Unlike TeachMe, uWave starts by creating new gestures. Meaning that the system has no predefined or hardcoded gestures before starting the system. uWave also targets the systems that have limited resources such as mobile devices. This problem is not part of TeachMe research prototype. Apparently, uWave was designed to work in different domains; however, the researchers do not include any mention of the possibility of using it in the domain of controlling a presentation or a mouse pointer. Instead, they include an experiment with results of using uWave for user authentication when is using 3D mobile interface.

[27] introduced a system that is very close to TeachMe. They use Kinect for gestures capturing and provided a

machine-learning algorithm for recognition. Their algorithm allows the user to define new gestures and mapping them to commands. They applied their algorithm in the e-learning field through connecting Kinect with smartphones to control discussion or presentation interactively. However, they document that their algorithm "requires 3-5 times of trainings" to be able to recognize a new gesture. As explained before in this paper, TeachMe does not require more than one example for training in its adoption of PBE technique. One more similarity between TeachMe and [27] system is that both can recognize voice. However, their system recognizes voice as commands (actions) to be performed, while TeachMe recognizes voice to enter the learning mode that allows the user to teach the system for the purpose of adding or customizing gestures. One more reviewed gesture recognition system for controlling presentation that can also recognize voices is introduced by [31]. Unlike TeachMe, the gesture and voice are used together in their system. Voice commands are used for refinement of gesture commands to increase the accuracy of recognizing the needed action.

Another gesture recognition system was presented by [32]. The system uses different capturing technique than TeachMe uses as it depends on 3D motion capturing. In the same paper, they were trying to provide the user with the ability to customize 3D gestures through teaching the system from different examples. Unlike TeachMe, their system requires several examples to learn and they work on 3D capturing for different purpose than TeachMe. One more documentation in the field of training a gesture system was introduced by [8] who reported that traditional and current techniques of increasing the efficiency of a gesture classifier require very large of training cases. They eliminate the possibility of adding new gesture for a gesture recognizer (classifier) at runtime or with very short time. This is because such addition of new gesture to the database of the classifier requires intensive training with dozens of validation sets. TeachMe overcame the problem of ability to teach the system at runtime using PBE technique.

From the above reviewed research papers, it can be noticed that none of them has introduced a system that uses Kinect as an interface for controlling MS PowerPoint presentation and mouse pointer and uses PBE to customize gestures. Accordingly, this research is unique in different manners. Firstly, it includes the notion of PBE in the field of gesture recognition systems and gesture customization. Secondly, TeachMe can control MS PowerPoint presentation and mouse pointer movement using gestures through Kinect interface with very simple features. Finally, it is unique with its ability to customize and add new gestures and its associated commands using gesture recognition and voice recognition features. These TeachMe features are considered as the main contributions of this research.

4 TeachMe Limitations and Future Work

Currently, TeachMe has two limitations that will be considered to be handled and developed in future work. The first is that TeachMe has no technique to customize gestures that control mouse pointer. The problem is that mouse pointer controlling comes as pre-compiled code chunks which prevents manipulating it. The mission in the next implementation is to reprogram these chunks from scratch to be able to maintain them into the data structure, dictionary (or hash-map) in this case, with the gestures that control PowerPoint presentation.

The second limitation is that when a user customizes or adds new gestures, this will be saved in the system at runtime and saved in text file to be retrieved in the next run. When another user tries to use the same customized version of TeachMe, s/he may change this customization or return it back to default. The next implementation of TeachMe should include a user account that maintains and saves their customization in different text files. Similar note and work were introduced by [27]. They introduced a system through which the user is able to customize and invent new gestures with some training for the system. They also introduced the idea of using an account for each user of the system that reserves their customized and newly added gestures.

5 Conclusion

This paper introduced the application of programming by example (PBE) notion in the field of gesture recognition systems through a system called TeachMe. TeachMe is a gesture recognition system that allows the user to control MS PowerPoint presentation and mouse pointer using Kinect as an interface. In most reviewed gesture recognition systems that control presentation tools, several problems were noticed and can be summarized in three points. Firstly, some users may need to add new actions because one or more required actions are not included. Accordingly, this requires to enrich the gesture vocabulary by adding new gestures to be mapped to the new required actions. Secondly, some users may need to customize implemented gestures to suit the preference of the user. Thirdly, users with special needs may require to customize gesture to suit their disability.

The paper introduced an implementation of TeachMe that enables the user to customize or add new gestures as a solution for the introduced research problems. Customization or adding new gestures is done through entering TeachMe into learning mode using voice recognition library available in Kinect. The system enters learning mode when the user says “Ok TeachMe”. The user then can provide the required gesture and the required action. The system maps the gesture to the action and reserves them for later use. The system notifies the user if they are going to override an existing gesture and shows the action that will be replaced with the new action. The implemented customization feature using programming by example opens the ability for special needs people such as those are amputated to use gesture system for controlling their presentation.

Acknowledgement

We would like to thank the students who participated in building TeachMe prototype and its new feature, Mohammad Zedan Ismael, Rslan Zedan, Saif Netham, Mohammad Khaleel. The author is grateful to the Applied Science Private University, Amman, Jordan, for the full financial support granted to this research.

References

[1] S. Cuccurullo, R. Francese, S. Murad, I. Passero and M. Tucci, "A gestural approach to presentation exploiting motion capture metaphors," in Proceedings of the International Working Conference on Advanced Visual Interfaces, Capri Island, Italy, 2012.
[2] D. Martinovikj and N. Ackovska, "gesture recognition solution for presentation control," in The 10th Conference for Informatics and Information Technology (CIIT 2013), 2013.
[3] S. Butnariu and F. Girbacia, "Development of a Natural User Interface for Intuitive Presentations in Educational Process," in The 8th International Scientific Conference eLearning and software for Education, Bucharest, Romania, 2012.
[4] S. Change, "Using Gesture Recognition to Control PowerPoint Using the Microsoft Kinect," MIT, Massachusetts, 2013.
[5] H. Qattous, B. Sowan and O. AlsheikSalem, "TeachMe, a Gesture Recognition System with Customization Feature," International Journal of Advanced Computer Science and Applications, 2016. (accpted on 19 November 2016)
[6] T. Osunkoya and J.-C. Chern, "Gesture-Based Human-Computer-Interaction Using Kinect for Windows Mouse Control and PowerPoint Presentation," in Proc. For the 46th. Midwest instruction and computing symposium (MICS2013), Wisconsin, 2013.
[7] A. Butalia, D. Shah and R. Dharaskar, "Gesture recognition System," International Journal of Computer Applications, vol. 1, no. 5, p. 48–53, 2010.
[8] T. Hachaj and M. Ogiela , "Rule-based approach to recognizing human body poses and gestures in real time," Multimedia Systems, vol. 20, no. 1, pp. 81-99, 2014.
[9] J. Liu, L. Zhonga, J. Wickramasuriya and V. Vasudevan, "uWave: Accelerometer-based personalized gesture recognition and its applications," Pervasive and Mobile Computing, vol. 5, no. 6, p. 657–675, 2009.
[10] T. Doring, K. Dagmar, P. Marshall, M. Pfeiffer, J. Schoning, V. Gruhn and A. Schmidt, "Gestural interaction on the Steering Wheel Reducing the Visual Demand," Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 483-492, 2011.
[11] B. Myers, "Peridot: Creating User Interfaces by Demonstration," in Watch What I Do: Programming by Demonstration, Cambridge, Mass., MIT Press, 1993.
[12] B. Myers, "Visual Programming, Programming by Example, and Program Visualization: A Taxonomy," CHI Proceedings, pp. 59-66, 1986.
[13] J. Webb and J. Ashley, Beginning Kinect Programming with the Microsoft Kinect SDK, California: Apress, 2012.
[14] D. Kurlander and S. Feiner, "Inferring constraints from multiple snapshots," ACM Transactions on Graphics, pp. 277-304, 1993
[15] G. Singh, C. Kok and T. Ngan, "Druid: A System for Demonstrational Rapid User Interface Development," in UIST '90 Proceedings of the 3rd annual ACM SIGGRAPH symposium on User interface software and technology, SNOWBIRD, UTAH, 1990.
[16] A. Dey, R. Hamid, C. Beckmann, I. Li and D. Hsu, "a CAPpella: programming by demonstration of context-aware applications," in Conference on Human Factors in Computing Systems, Proceedings of the SIGCHI conference on Human factors in computing systems, Vienna, Austria, 2004.
[17] R. McDaniel and B. Myers, "Getting More Out Of Programming-By-Demonstration," in Conference on Human Factors in Computing Systems, Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit, Pennsylvania, 1999.
[18] D. Smith, A. Cypher and J. Spohrer, "KidSim: programming agents without a programming language," Communications of the ACM, vol. 37, pp. 54-67, 1994.
[19] M. Prabaker, L. Bergman and V. Castelli, "An evaluation of using programming by demonstration and guided walkthrough techniques for authoring and utilizing documentation," in Proceedings of the SIGCHI conference on Human Factors in computing systems, Montreal, QC Canada, 2006.
[20] T. Lau, S. Wolfman, P. Domingos and D. Weld, "Learning Repetitive Text-editing Procedures with SMARTedit," in Your wish is my command: giving users the power to instruct their software, Morgan Kaufmann, 2001.
[21] J.-H. Chen and D. Weld, "Recovering from Errors during Programming by Demonstration," in IUI '08 Proceedings of the 13th international conference on Intelligent user interfaces, Gran Canaria, Spain, 2008.
[22] B. Argall, S. Chernova, M. Veloso and B. Browning, "A survey of robot learning from demonstration," Robotics and Autonomous Systems, vol. 57, no. 5, pp. 469-483, 2009.
[23] B. Lewis and G. Sukthankar, "Know thy user: Designing human-robot interaction paradigms for multi-robot manipulation," in IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, 2014.
[24] R. Abraham and M. Erwing, "Inferring Templates from Spreadsheets," in International Conference on Software Engineering, Shanghai, 2006.
[25] M. Toomim, S. Drucker, M. Dontcheva, A. Rahimi, B. Thomson and J. Landay, "Attaching UI enhancements to

websites with end users," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Boston, USA, 2009.
[26] J. Nichols and T. Lau, "Mobilization by demonstration: using traces to re-author existing web sites," in Proceedings of the 13th international conference on Intelligent user interfaces, Gran Canaria, Spain, 2008.
[27] V. Tam and L.-S. Li, "Integrating the Kinect Camera, Gesture Recognition and Mobile Devices for Interactive Discussion," in IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE), Hong Kong, 2012.
[28] S. Ha, S. Park, H. Hong and N. Kim, "Study on Gesture and Voice-based Interaction in Perspective of a Presentation Support Tool," Journal of the Ergonomics Society of Korea, vol. 31, no. 4, pp. 593-599, 2012.
[29] A. Argyros and M. Lourakis, "Vision-based interpretation of hand gestures for remote control of a computer mouse," in Computer Vision in Human-Computer Interaction, Berlin Heidelberg, Springer, 2006, pp. 40-51.
[30] F. Farhadi-Niaki, R. GhasemAghaei and A. Arya, "Empirical Study of a Vision-based Depth-Sensitive Human-Computer Interaction System," in 10th Asia Pacific Conference on Computer Human Interaction, Matsue, Japan, 2012.
[31] J. Kim, S. Kim, K. Hong, D. Jean and K. Jung, "Presentation Interface Based on Gesture and Voice Recognition," in Multimedia and Ubiquitous Engineering, Berlin Heidelberg, Springer, 2014, pp. 75-81.
[32] A. Bigdelou, T. Benz, L. Schwarz and N. Navab, "Simultaneous categorical and spatio-temporal 3D gestures using Kinect," in 2013 IEEE Symposium on 3D User Interfaces (3DUI), Costa Mesa, CA USA, 2012.
[33] R. Fiebrink, P. Cook and D. Trueman, "Human Model Evaluation in INteractive Supervised Learning," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, New York, 2011.

Dr. Hazem Qattous is currently an assistant professor at Applied Science Private University in Amman, Jordan. Dr. Qattous holds a Ph.D. degree in Computing from Glasgow University, UK. His research interests are in Human-Computer Interaction.