Developing E-learning lectures using interactive whiteboard

Howida A. Shedeed[1], Karim A. Emara[2], E. K. Maklouf[3], M. G. Barakat[4], M. R. Elhoufy[5]

Faculty of Computer and Information Sciences, Ain shams university

Key words: *Elearning developing tools, Wireless technology in elearning, Interactive whiteboards in elearning*

Abstract:

This paper presents the design and implementation of a domain-specific tool used to develop elearning content. This tool uses a low cost interactive whiteboard which created based on using a wii Remote video game controller. Our tool helps the instructor in explaining her lessons with less effort and high facilities. These facilities include: 1-Recognize/store hand-written text in real time 2-Recognize and store drawing objects that used in drawing basic charts, UML diagrams, flowcharts and so on. 3- Enable the instructor to do modification on the drawing object such as resizing / moving / rotating / copying or deleting. 4- Illustrate and trace the execution of a hand written pseudo code by converting it after recognition to C# code and then compile and trace the execution of it on the whiteboard. The benefit derived from the recognition of the instructor's lecture is to give him the facility to store it in digital format which can be easily shared with students. Another benefit is the ability to convert such saved file to SCORM format to be reused in constructing the E-Course for the considered subject.

1. Introduction

This paper presents the design and implementation of a domain-specific elearning tool. This tool uses a low cost interactive whiteboard created based on the Wii remote video game controller (Wiimote). A main feature of the Wii Remote is tracking the infrared light [2]. This feature can be used to allow user to interact with items on a projected screen via an infrared pen [1].

An interactive whiteboard is a large interactive display. A projector projects the computer's desktop onto the board's surface, where users control the computer using a pen, finger or other device. The board is typically mounted to a wall or on a floor stand. Interactive whiteboards may use one of several types of sensing technology to track interaction on the screen surface: resistive, electromagnetic, infrared, laser, ultra-sonic, and camera-based (optical). We are interested in Infrared technology. When we point to the whiteboard surface by a special marker with infrared led, the infrared camera in the Wiimote senses the infrared light emitted from marker led. The Wiimote manipulates the captured scene to track the location of the marker led and then sends this location information to the computer via Bluetooth connection. A calibration and mouse cursor emulation software developed by [1] is used to map the marker movement on the whiteboard to the mouse movement on the computer screen. This technology allows whiteboards to be made of any material.

2. Tool Idea

There are many problems face the instructors during the teaching of their lectures like drawing charts, shapes, 2D images and tracing codes on whiteboards in traditional face to face teaching. They may take long time to draw/modify with poor quality of drawing and it is so difficult to save/retrieve this drawing for later use. Moreover, while explaining coding problems, it may be hard to trace written code on the board. The idea of our tool is to develop an interactive whiteboard based on the idea of "Johnny Lee" [1] for the low cost interactive whiteboard and then use it on developing a domain-specific **E-Learning** tool. This tool exploits the **interactive whiteboard** to provide the instructor with all features help him in explaining his lessons on the traditional whiteboards and then benefit from his lectures in developing the ecourse for the considered subject.

3. Tool features

Our tool supports the following features:

1. Recognize hand-written text in real time
2. Recognize drawn objects which used in drawing basic charts, UML diagrams, flowcharts, and so on.
3. Enable the instructor to do modification on the drawn object such as resizing it automatically on the board, moving, copying and deleting the selected object.
4. Illustrate and trace a hand written pseudo code by converting it after recognition to C# code and then compile and trace its execution on the whiteboard.
5. The benefit derived from recognition of the instructor's lecture is to give him the facility to store his lecture after recognition in digital data format. This file can be shared with students and can be converted to SCORM format to be reused in constructing the Ecourse for the considered subject.

Thus, our tool provides the instructor with all features that help him in explaining his lessons and preserving his lectures to be reused with less effort and high facilities.

3.1 Handwriting and Shape recognition

The first track in our tool is the recognition of handwriting text and shapes. *Handwriting text recognition* is the computational analysis of handwritten digital ink to return character-based interpretation in a given language. In handwriting recognition, the image of the written text sensed "on line" using intelligent word recognition which performs correct segmentation into characters and finds the most plausible word. Handwriting recognition algorithm may be Online or Offline. Online recognition is applied just after a new stroke is drawn while the Offline recognition is applied after finishing drawing the whole text. Both types are used in different cases in the tool.

There are many researches in handwriting and shape recognition such as [7], [8], and [9]. we used Microsoft InkAnalysis APIs. They are built-in .Net Framework 3.5 to implement our model of handwriting and shape recognition, because we find that it is an advanced tool that contains most of the advanced features we need for handwriting and shape recognition. The Tablet PC handwriting recognition engines have been primarily designed to recognize a single horizontal line of ink[10]. With the InkAnalysis APIs, the analyzed ink is transformed to horizontal before being recognized which improving the recognition results.

We apply a recognition algorithm on strokes formed from points sent by Wiimote as shown in Figure1. Our recognition algorithm consists of the following four steps:

1. Setting up the recognition constraints by: Selecting the recognizer, selecting input mode (geometric constraints), and then setting the language constraints. For example, it is possible to restrict input to alphanumeric characters or pass schemas to assist the recognizer.
2. Sending Ink to the recognizer.
3. Processing the input (recognizing).
4. Returning the results of the recognition.

Steps 2 and 3 may be repeated in a loop for online recognition. Otherwise (offline recognition), all of the Ink Strokes may be added to the Ink before recognition.

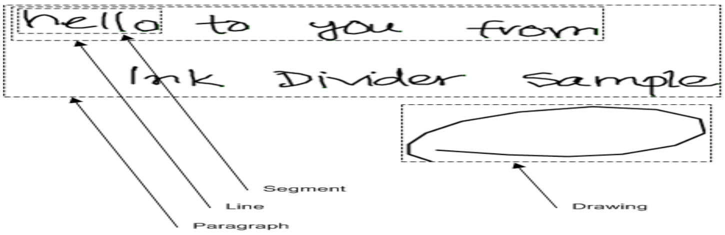


Figure 1. Ink sample

After recognition we apply the technique, known as the *k*-gram index, for processing wildcard queries to correct the spelling of the recognized word[10]. In a *k-gram index*, the dictionary contains all *k*-grams that occur in any term in the vocabulary. Each postings list points from a *k*-gram to all vocabulary terms containing that *K*-gram. We use the 3-grams index in our implementation.

3.2 Pseudo Code Tracing

The second track in our tool is to enable the instructor to write a pseudo code and trace its execution on the whiteboard. Since the pseudo-code has no standard or compiler [5], we converted it to a known programming language (C#, for example) to support compilation and tracing. The tool recognizes the hand-written pseudo code, converts it into C# code then compile and run it according the following steps:

1. Parse the given program code and extract variables
2. Insert the parsed code into a template code which supports inspecting variable values after each code statement
3. Update the value of changed variables after executing each statement
4. Show and highlight updated values.

As, there is no standard rule for pseudo code syntax exists [5], so we put our pseudo code syntax rules. Some samples of our syntax rules are shown in Table 1.

Operation	Allowed phrases	Operation	Allowed phrases
Input	read, get, obtain	Output	write, show, display, print
Increment	increment, bump	Decrement	decrement
Addition	Add	Multiplication	multiply
Subtraction	subtract	Division	Divide
Initialization	init, init =,set	Assignment	assign,equal,=
Arithmetic operations	=, ≠, <, >, <=, >=, equal, not equal, less than, greater than, less than or equal, greater than or equal.	Condition	if-then-endif, if-then-else-endif, if then else if then else ….endif)
Looping	While-endwhile, do-while, repeat-until, for-endfor loop-while		

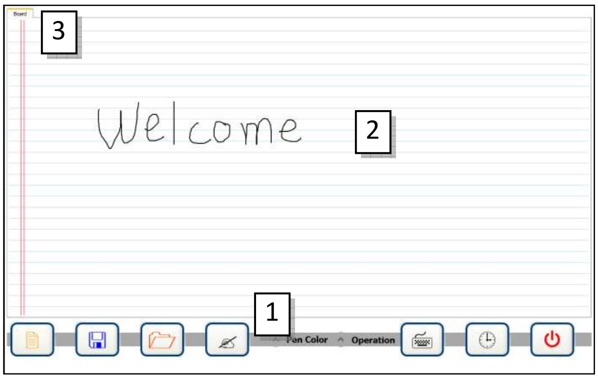
Table 1. Pseudo Code Syntax

4. System Implementation

Our tool is implemented using Microsoft Visual C# 2008 to use the new technology of WPF which enables using the InkAnalysis APIs for handwriting and shapes recognition.

5. System Functional Description

Figure 2.A shows the main user interface screen and (Figure 2.B) Explain the system user interaction using a projector, an InfraRed pen and Wiimote



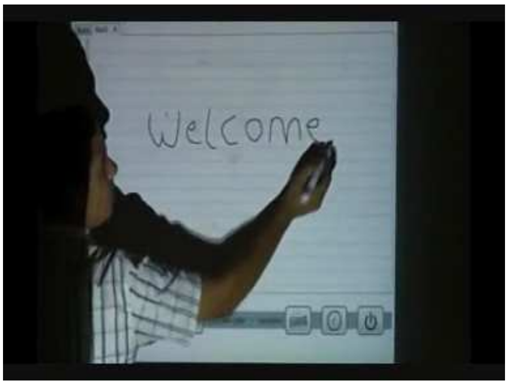
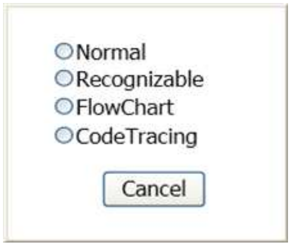


Figure 2. A: A screenshot for the user interface. **Figure2.B: A snapshot for system usage.**

The main interface consists of three parts: 1- a Toolbar 2- the Board area 3- Tabs for open boards. The toolbar at the bottom contains buttons to begin new screen, save the current screen or append it to the previous in the lecture's file, open previously saved file to add on it, delete a selected object. Also it contains at the right a keyboard tool, a clock and Exit buttons.



○ Normal
○ Recognizable
○ FlowChart
○ CodeTracing

Cancel

Figure 3: Boards Types

The user interface allows the instructor to use multiple whiteboards. There are four types of boards as shown in figure 3. The first type is a normal board which allows the instructor to write freely on it without any recognition. The second type is a recognizable board which recognizes the user hand-written text and shapes. The third type is a “FlowChart” board which recognizes hand-drawn shapes and approximates them to standard flowchart shapes. The last board type is the “CodeTracing” board which allows the instructor to hand-write pseudo-code and recognize it for running and tracing. The instructor can select the board type when he opens a new board using the “New” button. He can open multiple boards and navigate between them from the upper boards’ tabs. Based on the selected board, the operations in the “Operations” list box on the toolbar are changed as illustrated next. In case of the normal type of board the content of the board can be saved as image file but for the “Recognizable” and “FlowChart” types the content before recognition can be saved as image file and after recognition can be saved as “xml” format or text format. For the “CodeTracing” type of board the recognized code can be saved as text format.

5.1 Recognizable Board

In “Recognizable” board, the instructor can hand-write text and shapes which are recognized immediately. For the “Recognizable” board the allowed operations include : ***copy, cut, paste***, ***delete, Draw***: to switch to drawing shapes mode, ***Edit***: to switch to hand-write mode, ***Select***: to select an object, ***minimize*** and ***maximize*** the selected object. Figure 4 shows how recognizable board is used and the Operations list allowed for this type of board

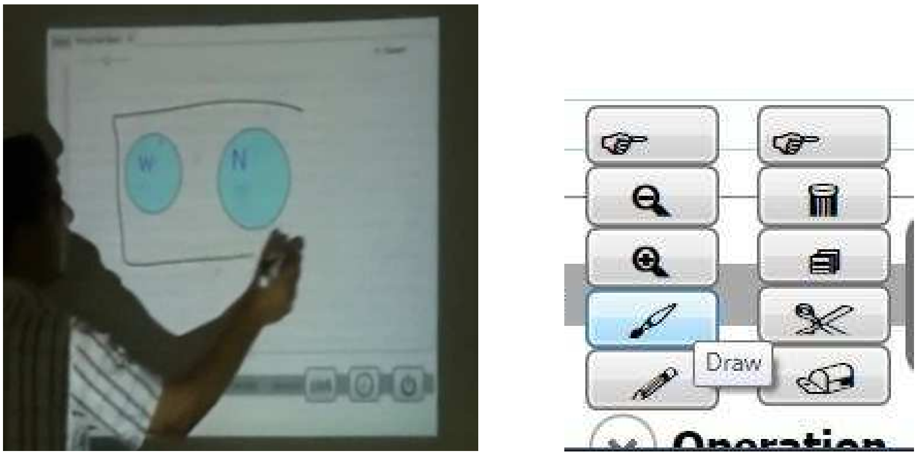


Figure 4. Recognizable board

5.2 Flowchart Board

This is a specialized type of board that used to draw flowcharts only. In “FlowChart” board, the instructor can hand draw flowcharts which are recognized immediately to the most similar flowchart standard shape. Recognized shapes can be connected together using flow arrows. They also can be resized, copied and pasted, moved and deleted. Instructor can add text to the flowchart shapes by hand-writing or by on-screen keyboard. Figure 5 shows features of flowchart board and the allowed operations for this type of board.

5.3 Code Tracing Board

In code tracing board, the instructor can hand-write pseudo-code and run it. For this board type the allowed operations include: Recognize, “Start tracing”, “Erase by point” and “Erase by stroke”. The pseudo-code is recognized and converted into C# code to be compiled and run as shown in Figure 6.A. Instead of writing code, instructor can open ready-written code from a text file and trace it. In both cases, the instructor can run the code step by step and input the values of code’s variables and/or arrays’ elements using the keyboard tool as shown in Figure 6.B

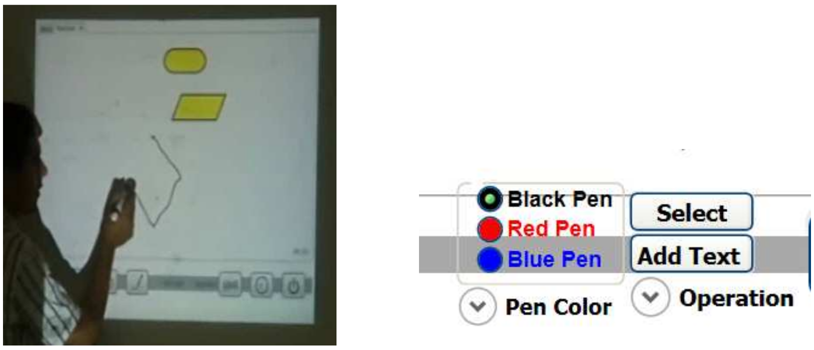
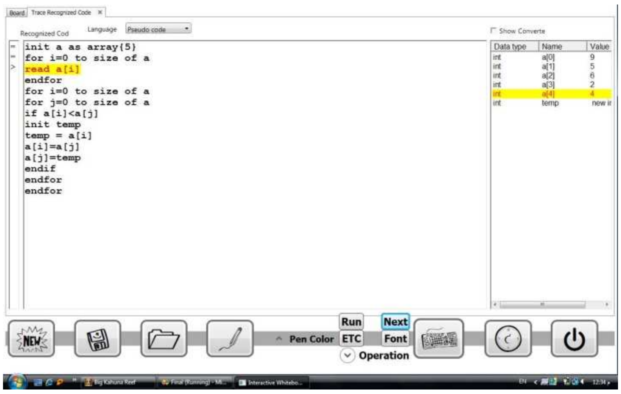
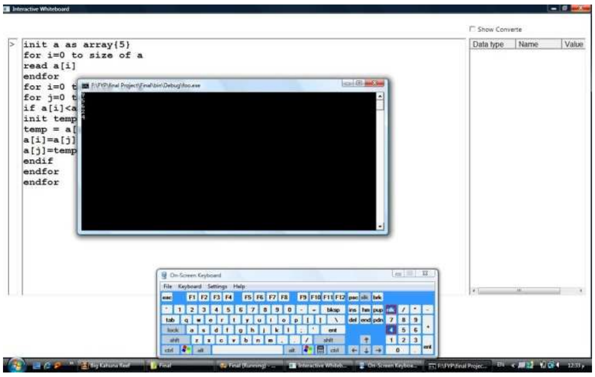


Figure 5. Flowchart board



```
init a as array{5}
for i=0 to size of a
read a[i]
endfor
for i=0 to size of a
for j=0 to size of a
if a[i]<a[j]
init temp
temp = a[i]
a[i]=a[j]
a[j]=temp
endif
endfor
endfor
```

Data type	Name	Value
int	a[0]	9
int	a[1]	5
int	a[2]	6
int	a[3]	2
int	a[4]	4
int	temp	new ir



```
init a as array{5}
for i=0 to size of a
read a[i]
endfor
for i=0 t
for j=0 t
if a[i]<a
init temp
temp = a[
a[i]=a[j]
a[j]=temp
endif
endfor
endfor
```

Data type	Name	Value

Figure 6.A CodeTracing board **Figure 6.B using the keyboard in Code Tracing**

6. Steps of using the tool

To conclude, the steps of using our tool are as follows:

1. The first step is to connect the Wiimote to the PC via Bluetooth. There are a number of online tutorials explaining how to do so such as [4], even for your specific software/hardware configuration. The Wiimote works with many (but not all) Bluetooth drivers. You can report/read about compatibility issues at [3].
2. Make sure your Wiimote is connected via Bluetooth. Note that the good placement of the Wiimote is a key for good pen tracking.
3. Buying/Building the IR pen. There is a simple schematic showing how to hand-made an InfraRed pen at [1].
4. Run the executable file for our tool which begins by running the calibration and mouse cursor emulation software that can be downloaded from [1].

5. Mark the four points shown on the calibration window that constructthe boundaries for InfraRed Camera.
6. Select a new board specifying its type and begin operating normally like a traditional board but enjoy with much more features and facilities.

7. Conclusion and Future Work

This paper presents an innovative idea to build a domain specific tool that recognizes the handwriting text and drawing objects written on the traditional whiteboard. It also helps the instructor to trace the handwritten pseudo code. The tool stores the lecture after recognition in digital data format which can be shared with students. Another benefit is the ability to convert the saved file to SCORM format to be reused by the instructor in constructing the ecourse for the considered subject.

Future work will include adding the recognizing of the handwritten arithmetic equations and calculating it on line. Allowing the use of multiple pens to allow multi-touch capabilities. We can include 3D shapes recognition to support other domains such as Computer graphics and visualization. In code tracing we can try to support more conventions and expressions. Also the calibration and mouse cursor emulation software should be enhanced to provide more accurate sensitivity.

8. References

1. http://johnnylee.net. Retrieved at May 2009
2. Wii:TechnicalDetails".Nintendo of Europe. http://www.nintendo.co.uk/NOE/en_GB/systems/technical_details_1072.html Retrieved at May 2009
3. http://www.WiimoteProject.com. Retrieved at April 2009
4. http://www.instructables.com. Retrieved at May 2009
5. http://en.wikipedia.org/wiki/Pseudocode Retrieved at May 2009
6. Bertolami R.; Bunke H.: Including Language Model Information in the Combination of HandwrittenText Line Recognizers, 11th International Conference in frontiers in handwriting recognition, Aug 2008.
7. Christopher, D.; Prabhakar, R.; Hinrich, S.,"An Introduction to Information Retrieval" TextBook-online Edition © 2009 Cambridge UP – England.
8. Gregory R. B.;Sargur N. S.:Prototype Integration in Off-line Handwriting Recognition Adaptation, 2008.
9. Likforman-Sulem L.; Vinciarelli A.: HMM-based Offline Recognition of Handwritten Words Crossed out with Different Kinds of Strokes, 11th International Conference in frontiers in handwriting recognition, Aug 2008.
10. West,J.V. Windows XP Expert Zone Community Columnist: Using Tablet PC Handwriting Recognition 101, May 28,2003. Retrieved May 2009

Authors

1. Howida A. Shedeed is a lecturer in Scientific Computing Department, Faculty of Computer and Information Sciences (FCIS), Ain Shams university, Cairo, Egypt. Email: Dr_howida@yahoo.com
2. Karim A. Emara is a teaching assistant in Computer Systems Department, Faculty of Computer and Information Sciences (FCIS), Ain Shams University, Cairo, Egypt. Email: kemara@eun.eg

3. Ehab K. Maklouf, graduated student 2009, Computer Systems Department, FCIS.
4. Mohamed G. Barakat, graduated student 2009 Scientific Computing Department, FCIS
5. Mohamed R. Elhoufy, graduated Student 2009, Information Systems Department, FCIS.