

Virtual Fabric Paint Environment Using Touch and Gesture for Vocational Training

Nishant Gupta¹, Yogeshwara K¹., Vinay Kumar R.N.¹, Rakesh Singh Bisht¹, Yaswanth Ginnela¹, Kumar Rajamani¹, Kamal Bijlani², Krishnashree Achuthan², Bhavani Bijlani²

¹Amrita Vishwa Vidyapeetham, Amrita School of Engineering, Bangalore 35

²Amrita Vishwa Vidyapeetham, Amrita School of Engineering, Amritapuri, Kollam, Kerala - 690 525

kumar_rajamani@blr.amrita.edu, krishna@amrita.edu, bhavani@amritapuri.org

Abstract—The objective of this work is to enhance the pace of learning occupational skills by developing multimedia enriched computerized vocational training programs targeted towards the illiterate and neo-literate sections of a society. This is achieved by developing virtual computer based vocational training environments. The experience of these environments is enriched with touch and gesture based technology. This paper talks about the creation of virtual vocational training scenario of a fabric painting application. We have successfully created a working prototype which brings out the real experience of fabric painting to the end user. Emphasis is placed on human-computer interactions that will allow simplistic interface for illiterate or neo-literate individuals. This approach of ‘virtual’ computer-based vocational training environment has great potential that stimulates self-learning processes of these vocational skills to a diverse audience with varying skill sets.

Keywords— Touch and gesture, Fabric Painting Systems, Human Computer Interaction, Vocational Training, Real World Scenario of Colour Mixing, Haptics

I. INTRODUCTION

The art of painting refers to the aesthetic aspects of a painterly work. The craft of painting deals with the study of materials, including paint medium, tools, methods, i.e. the manipulation of materials to express an artist’s intent and purpose[2][4]. Nevertheless recent technological advances in computer graphics have largely centred on the art of painting. Commercial graphics editing programs like Adobe Photoshop, Corel Painter, and Project Dogwaffle [15] yield themselves very well for painting. They also support touch and gesture based technology. They have rich features and have several configurations for each of the components. One of the limitations of existing solutions is that these systems have been developed for advanced users.

The focus of this work is to build a radically different environment for fabric painting application that will allow embrace the art of painting with ease and familiarity. Compared with traditional painting methods that uses the mouse to control the brush stroke, the touch, gesture devices

or haptic devices provide users with tactile information such as the sense of touch [9]. This is achieved by capturing the input parameters such as pressure, forces etc and also by sending feedback on reaction forces [6]. This provides greater freedom in ‘feeling’ and interacting within the virtual space.

This work is unique as it integrates touch and gesture based technologies to improve HCI elements to a challenging group of inerudite individuals.

II. THE STATE OF THE ART

This section presents background of touch and gesture based technologies with examples of a few historic and contemporary applications.

A. Touch and Gesture based Technology - Overview

A touch-sensitive tablet (touch tablet for short) is a flat surface, usually mounted horizontally or nearly horizontally, that can sense the location of a stylus around its vicinity. It is a tablet that can sense touch and provide the location coordinates of the touch points. Touch tablets can vary greatly in size, from a few inches to several feet [9]. They can detect touch from a variety of devices such as styli or pucks.

A simple touch tablet senses only one point of contact. However, more sophisticated tablets allow detection of the degree or pressure of a single or multiple contact points.. Thus the impact on several points of contact can be closely monitored to enhance the virtual effect of touch.

B. Applications

Touch and gesture based devices evolved historically from Sketchpad[10] and like application and devices which uses a video screen and a light pen. A user of the sketchpad could draw lines by touching the screen with the pen and then modify them by typing instructions on a keyboard. Cutting edge technologies such as haptics also incorporates learning from these touch and gesture sensitive devices [6], [7], [8]. Recent applications of this technology are mobile phones (iPhones in particular) which recreate touch and texture through artificial stimuli. Touch based inputs also find wide

applicative uses in the field of kiosks, banks, touch screens. It is also actively used in E-learning scenarios. The class room materials of professors are captured on touch sensitive devices.

Thus tablets are commonly used as teaching aids. This work brings out the usage of tablets as practical medium to impart vocational skills such as fabric painting.

III. TOOLS INVOLVED

A. Software used

1. Microsoft Tablet PC SDK 1.7 [13]
2. Microsoft Visual Studio Express Edition

The Windows-XP tablet SDK facilitates building ink-enabled, pen-enabled, and speech-enabled applications for computers which are connected to tablets and TabletPC. The user interface has been developed using Visual C#.

B. Hardware used



Fig 1: WACOM Tablet with stylus and mouse.

WACOM Intuos3 Details: Wacom® Intuos3 has touch sensitive area, customisable express keys, and customizable touch strip controls. There is also a status LED at the top which indicates the active/idle status of the tablet. Fig.1 shows the WACOM® Tablet with stylus and mouse.

Intuos3 tools have unique IDs for each tool and are cordless and battery free. This enables usage of multiple tools and customization of each tool. The Intuos3 grip pen is a sensitive to pressure and is used as a freehand tool for image editing and creating natural looking pen and brush strokes.

IV. SYSTEM OVERVIEW

In this section, the proposed Paint application will be presented from a system point of view. The system comprises of three parts: general paint, water colour-mixing and suggestive learning module.

A. Layered view of the system

As shown in Fig 2. the Wacom tablet captures the events from the stylus used for drawing. The internal functioning of the system can be categorized as follows:

1. Packets sent by tablet to SDK when stylus is down. (one way communication)
2. Two way communication between SDK APIs and the tablet plug-in event.

3. Two way communication between the tablet event plug-in and the interface.

Prior to processing, the backend interacts with Microsoft Tablet PC SDK. The SDK acts as a library resource for all the inbuilt functions. It has built-in APIs defined, to recognize the occurrence of stylus events pertaining to changes in pressure and tilt. After processing the events, the result is shown on the interface.

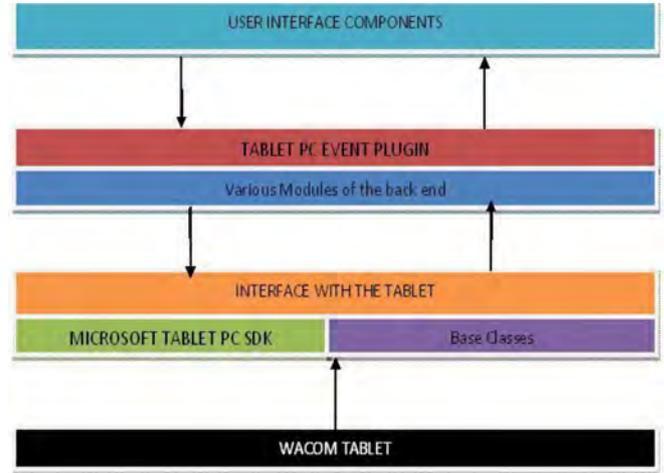


Fig 2: Application architecture.

B. System Description

In General Paint subsystem, the user can choose between pencil, two major types of brushes (round and flat) and an eraser. Water colour mixing module simulates real scenario water colour mixing. Within the suggestive learning module, the user selects design patterns from an inbuilt library. The design pattern is animated on the drawing area guiding the user to trace the pattern. Additional feedback mechanism is incorporated to guide the user to trace the pattern correctly. This module also takes the tilt and pressure into account.

C. Stylus Features

The pressure detection mechanism developed in this work is based on the pressure applied on the stylus by the user while drawing on the work area. The pressure variation on the stylus results in changes of the ink width.

Tilt detection is an additional feature of this application. The tilt of the stylus or the amount by which the stylus is bent is captured. This information is used in varying stroke properties. The tilt variation manifests in the form of line width variation. In case of brushes with increasing tilt, larger area is covered by the brush resulting in rendering more ink.

The azimuth/orientation of the stylus is important for the performance of a flat brush. Flat brush renders thin ink when the direction of the stroke is perpendicular to the azimuth of stylus. The thickness thus varies in accordance with the azimuth of the stylus and the direction in which the brush draws a stroke.

V. GENERAL PAINT MODULE

A. Components:

The components in our application include pencil, round brushes (thin, medium or thick styles), flat brush, eraser, undo and redo, colour chooser button, template chooser, next and previous buttons and a palette of colours.

The pencil feature is used to draw outlines with a constant line width. If the stylus is inverted, it acts like an eraser which erases strokes drawn by both brushes and pencil.

Round brush feature, on the other hand, have width of the stroke line proportional to the pressure and tilt applied on the brushes.

Flat brush feature takes into account all the three parameters (pressure, tilt and azimuth) of the stylus. Azimuth of the stylus determines the orientation of flat brush. Both azimuth and direction of the pen are considered while determining the thickness of painted strokes. As an example, if the pen is moved along its flat edge, it results in a thick painted line. If the pen is moved perpendicular to its flat edge, the resulting line is thin.

Eraser feature is used to erase stroke. Undo option takes a step back to the task performed by the user. Redo option repeats the previous operation again.

Colour chooser has a palette of colours which are provided to allow free selection of colours.

Next and previous buttons are used to toggle between the design pattern templates.

B. Working of Round Brush:

The pressure parameter in the packet data received from the tablet is an integer. The stroke width is computed using

$$\text{Stroke width } W = W_{\text{selected-brush}} + [(\text{Pressure} - 256) / 200]$$

where $W_{\text{selected-brush}}$ is the default width of the selected round brush.

'XTiltOrientation' is a parameter which is acquired from the stylus. It is the angle between the y, z-plane and the pen and y-axis plane.

XTiltOrientation varies 0-60 degrees and 195-255 degrees with respect to y- axis beyond which the pen loses its contact with the surface.

Based on this observation an approximate formula is as shown below:

$$XTilt = \begin{cases} -(XTiltOrientation-255), & 195 < XOrientation < 255 \\ XTiltOrientation, & \text{otherwise.} \end{cases}$$

Variation of the stroke width with respect to XTilt is:

$$W = W_{\text{selected-brush}} + N * XTilt$$

where N is constant varying according to the type of brush selected.

C. Working of Flat Brush:

Flat brushes work in similar manner as round brushes except that it takes azimuth orientation into account. Azimuth orientation is the clockwise rotation of the cursor about the z-axis through a full circular range.



Fig 3(a)



Fig 3(b)



Fig 3(c)

Fig 3(a): Rendering of ink when azimuth is 0 or 180 degrees.

Fig 3(b): Rendering of ink when azimuth is 90 or 270 degrees.

Fig 3(c): Rendering of ink when azimuth is any other angle.

When azimuth of the brush is 0 or 180 degrees the bristles of the flat brush render thick ink in the direction of azimuth. To achieve this, the target pixel's adjacent pixels to the left and right are coloured. This direction in this case colours $(x-1, y)$ and $(x+1, y)$ as shown in Fig 3(a). Similarly when azimuth of the brush is 90 or 270 degrees the pixels at location $(x, y-1)$ and $(x, y+1)$ are coloured as shown in Fig 3(b). When azimuth of the brush varies with an angle the pixels the diagonal pixels are coloured i.e., pixels $(x+1, y+1)$ and $(x-1, y-1)$ or $(x+1, y-1)$ and $(x-1, y+1)$ as shown in Fig 3(c).



Fig 4: Rendering of ink according to movement of stylus when azimuth is 180.

Consider the case when azimuth is 180. If the stylus is moved horizontally, the pixels overlap and give the effect of a single line. This approach is used to capture thin line drawing using flat brush. When the stylus is moved vertically, the pixels above and below also get coloured thus drawing thick line as shown in Fig 4.

This can be generalized for all azimuths. The general form of a line with 'm' being the slope and 'c' y-intercept is considered here. If (x,y) is a point on the drawing area and also the origin i.e. $c=0$, then pixels that must be painted are:

$$(x,y), (x+n, y+m), (x-n, y+m) \quad \text{if } |m| \leq 1$$

$$(x,y), (x+n/m, y+n), (x-n/m, y-n) \quad \text{if } |m| > 1$$

where n varies from 1 to width of flat brush.

VI. WATER COLOUR MIXING MODULE

The water colour mixing module allows the user to mix colours and dilute them by adding water virtually. Visible difference in the transition of colours from mixing and dilution are achieved in the paint module.

In general paint applications it is standard practice to select colours from a roulette wheel or grid of colours and change its opacity using a slider. This process of selection is not so user friendly to naive painters. In real life, the painter dabs his brush over various colors and then sometimes in water etc. and find the right blend of colour before painting. This phenomenon is replicated identically in the application.

Practically speaking water colour mixing could be described as subtractive in nature mathematically [5]. It is

difficult to combine colours using additive RGB colour model. The most accurate colour mixing model has been simulated using Kubelka-Munk model [3] [1]. The colour mixing feature in this work uses approximation methods of function linearity to model colours and bring in transparency.

VII. SUGGESTIVE LEARNING MODULE-PATTERN TRACING

The suggestive learning module utilizes immersive learning techniques to enhance the learner’s ability to draw. This module consists of tools such as paint templates, match indicator and toggles in addition to the general paint modules.

The user selects design patterns from in built library of patterns. The design pattern is animated on the drawing area guiding the user to trace the pattern.

An important element of this module is the integrated feedback mechanism coupled with the match indicator. The match indicator guides the user in his tracing exercise and visually indicates if the tracing has been done correctly or not. With colour coding approach, the student immediately understands the deviation from the actual pattern and corrects himself automatically without external assistance. Red indicates major deviation, while green indicates total compliance to the pattern.

VIII. RESULTS

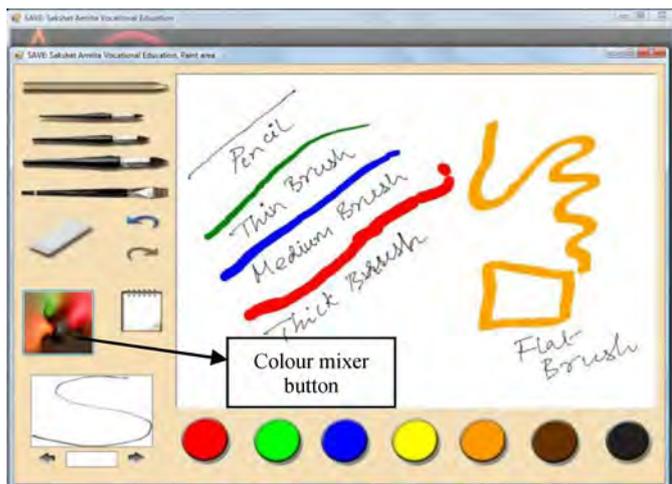


Fig 5: Screen shot of paint module depicting various types of brushes, eraser, undo or redo strokes on a canvas. The variation of width according to pressure and tilt is depicted. Also the colour mixer button is shown.

The screenshot in Figure 5 depicts the general paint module’s functionality with various sketches drawn using a pencil, three different types of round brushes and a flat brush using different colours.



Fig. 6: Colour mixer screen shot. The user can tap on the colour pots and draw on the canvas. The user can also add water and amount of water is indicated at the bottom. More the water added more is the transparency.

Fig. 6 depicts the features and functionality of the colour mixing module. In this figure it can be observed that various lines have been drawn with distinct or combination of colours. The faded colours also can be obtained as a result of adding water to the selected colour and making it transparent. Once a colour is chosen, the user can continue drawing using the selected colour from the colour mixer window.

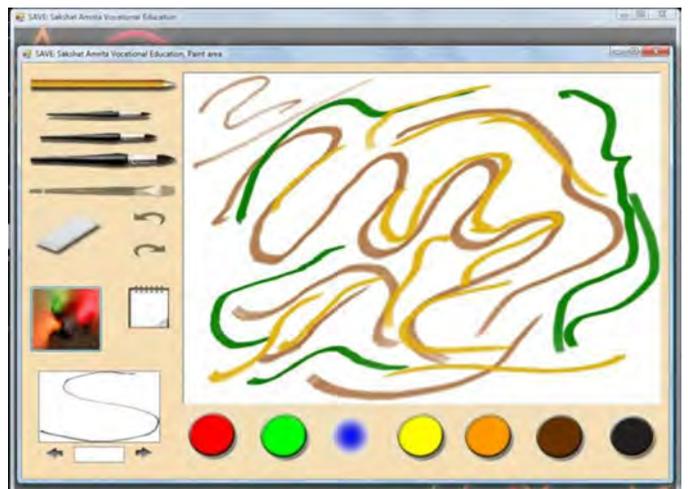


Fig. 7: Simulation of water colours. The colours overlap at point of intersections. This is near to real world scenario of painting.

Fig. 7 depicts the response to selection of colours after its mixing within the colour mixing module. It also depicts the usage of round (medium) as well as flat brush. The colours selected from the colour mixing module remain unchanged until a new colour is selected. The amount of water added in the colour mixing module accounts for the transparency of the colour selected. The overlapping of colours at the junction where two colours intersect depicts this. Figure 9a, 9b demonstrated sample graphics and patterns which have been painted using this application.

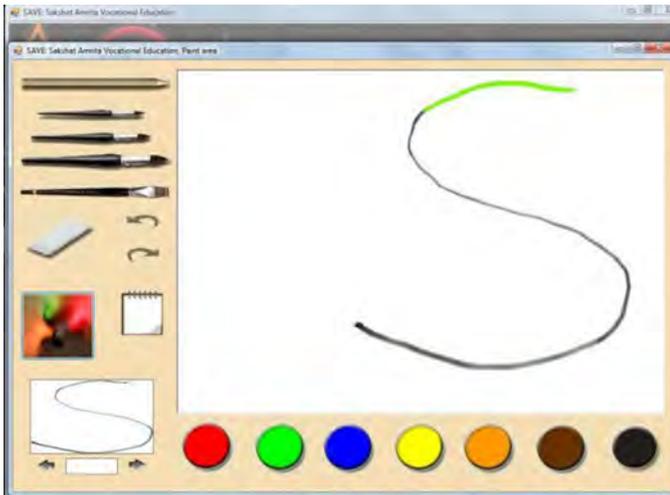


Fig. 8: Suggestive learning module with pattern matching, animating how the pattern should be drawn.

Fig 8 depicts the properties of the suggestive learning module. When user selects a template, the pattern template is loaded and animated to allow the user to explore tracing and learn the drawing characteristics.

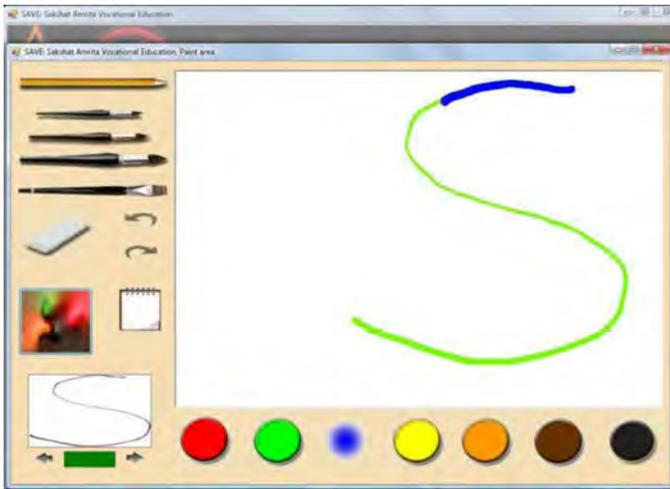


Fig. 9: Suggestive learning module with pattern matching showing the correct path traced by the user.

Fig. 9 depicts the pattern matching module in action when the user traces the path of the template correctly. The pattern match indicator glows green indicating the tracing compliance to the pattern. The user can choose any color from palette while tracing the pattern. The color selected for drawing will remain the same selected color when the tracing is proper.

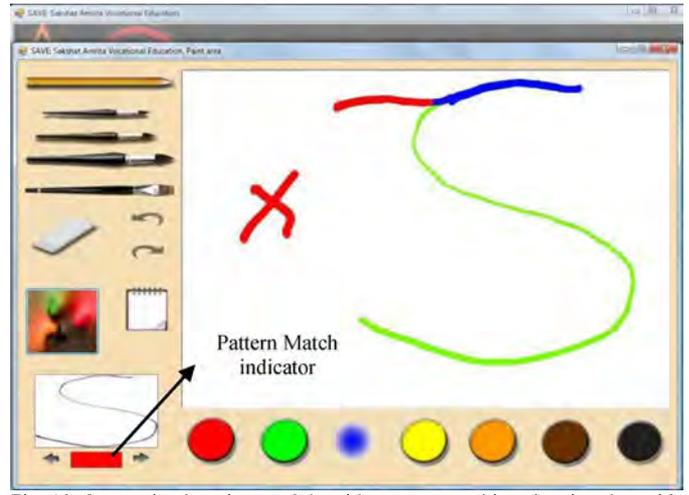


Fig. 10: Suggestive learning module with pattern matching showing the guide line being drawn for the user.

Fig. 10 depicts shows an example of the deviation of the trace from the pattern. The colour selected for drawing changes to red as well as the pattern match indicator turns red indicating the failure to trace the path correctly by the user. Apart from these visual cues, one could incorporate haptic interfaces to provides force feedback and to control/restrict user movement and simulate 3D brushes [2].

IX. USER STUDY

An extensive study of users and their perception, and grasp of this fabric painting application was done. A set of users between 15 to 35 years of age were used for this pilot study. All of them had less than 7th grade education. All of them were digitally illiterate with no introduction to computes. The response from this team has been very encouraging. 53% of the users were able to use the brush strokes effectively after an average of seven attempts at using the various brushes. Due to lack of hand and eye coordination, the match indicator tool was useful in indicating the subset of people (~ 33%) that required more training. The study was able to determine that those individuals that did some level of creative drawing in their daily lives were less prone to deviating from the trace patterns as opposed to those that had no background in drawing. About 75% of the individuals wanted to pursue actual fabric painting following this course.



Fig.11a,11b: Sample graphics and patterns which have been painted using this application.

X. CONCLUSION

This work has successfully built a prototype of fabric painting application catered to the illiterate and neo-literate population. The application mimics actual hand painting techniques unlike commercially available alternate solutions. The application takes into account all the features of painting such as tilt of a brush, the pressure applied on bristles, the azimuth of a flat brush and altitude keeping in mind simplicity and intuitiveness required for HCI. With touch and gesture sensitivity devices the user can actually practice painting and enhance the speed of learning significantly as proven by the user case study done in this work.

ACKNOWLEDGMENT

We are indebted to Dr. Kanav Kahol for his all-round committed support and active encouragement. We gratefully acknowledge Amrita Multimedia team for their staunch support and Josh Freeman for his active technical guidance and directions for this project. We would also like to acknowledge our Vice Chancellor Dr. Venkatrangan, Correspondent Br. Dhanraj, The Dean Dr. Nagaraj and The

Associate Dean Dr. Rakesh for their encouragement and giving us an opportunity for developing this Virtual Fabric Paint Environment.. Our special thanks to Rahul Jayadas and Amrita E-Learning Initiatives for their all round help and support to us.

REFERENCES

- [1] C. Curtis, S. Anderson, J. Seims, K. Fleischer, and D. Salesin. "Computer generated watercolour." *Proc. of SIGGRAPH* 1997.
- [2] W. Baxter, V. Scheib, M. Lin, and D. Manocha. "DAB: Interactive Haptic Painting with 3D Virtual Brushes". *Proceedings of ACM SIGGRAPH 01*, August 2001.
- [3] P. Kubelka, F. Munk, "An article on optics of paint layers", August, 1931.
- [4] R. Mayer. "The Artist's Handbook of Materials and Techniques." *The Viking Press*, 1970.
- [5] Jeng-sheng Yeh, Ting-yu Lien, Ming Ouhyoung, "On the Effects of Haptic Display in Brush and Ink Simulation for Chinese Painting and Calligraphy," *10th Pacific Conference on Computer Graphics and Applications (PG'02)*, pg. pp.439, , 2002
- [6] Thomas H. Massie, J.K. Salisbury, "The PHANTOM Haptic Interface: A Device for Probing Virtual Objects", *Proceedings of ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, Chicago, IL, Nov. 1994.
- [7] A.W.Beavis, J.W. Ward, R.M. Appleyard, R. Phillips, "A Virtual Environment for the Training and Development of Radiotherapy Techciques", *Proceedings of 48th Annual Meeting of the American Society for Therapeutic Radiology and Oncology*.
- [8] Kamen Kanev, Tomoyuki Sugiyama, "Design and Simulation of Interactive 3D Computer Games", *Computers & Graphics*, Volume 22, Issues 2-3, 6 March 1998, Pages 281-300.
- [9] Buxton, W., Hill, R. & Rowley, P. (1985). "Issues and techniques in touch-sensitive tablet input", *Computer Graphics, Proceedings of SIGGRAPH'85*, 19(3) 215-223.
- [10] Ivan Edward Sutherland, "Sketchpad: A man-machine graphical communication system", *Technical Report Number 574 MIT*, 2003
- [11] Lisa Yount, "Virtual Reality", *Thomson Gale Publisher*, 2005
- [12] "Microsoft Developers Network", www.msdn.microsoft.com
- [13] "WINTAB specifications" www.logicgroup.com/Wintab/
- [14] "Wikipedia", www.en.wikipedia.org
- [15] "Project Dogwaffle", www.thebest3d.com/dogwaffle/
- [16] Khaled Batiha, Safwan Al-Salaimeh, Khaldoun A.A. Besoul, "Digital Art and Design" *Electronic multi-topical Journal of International Research Publications* Volume 1, Issue Technomat & Infotel 2006