TangiPaint: A Tangible Digital Painting System

Anthony M. Blatner, James A. Ferwerda, Benjamin A. Darling, Reynold J. Bailey; Rochester Institute of Technology, Rochester, NY 14623, USA

Abstract

TangiPaint is a digital painting application that provides the experience of working with real materials such as canvas and oil paint. Using fingers on the touchscreen of an iPad or iPhone, users can lay down strokes of thick, three-dimensional paint on a simulated canvas. Then using the Tangible Display technology introduced by Darling and Ferwerda [1], users can tilt the display screen to see the gloss and relief or "impasto" of the simulated surface, and modify it until they get the appearance they desire. Scene lighting can also be controlled through direct gesture-based interaction. A variety of "paints" with different color and gloss properties and substrates with different textures are available and new ones can be created or imported. The tangiPaint system represents a first step toward developing digital art media that look and behave like real materials.

Introduction

The development of computer-based digital art tools has had a huge impact on a wide range of creative fields. In commercial art, advertisements incorporating images, text, and graphic elements can be laid out and easily modified using digital illustration applications. In cinema, background matte elements can be digitally drawn, painted, and seamlessly integrated with live footage. In fine art, painters, printers, and engravers have also been embracing the new creative possibilities of computer-based art tools. The recent introduction of mobile, tablet-based computers with high-resolution displays, graphics processing units (GPUs) and multi-touch capabilities is also creating new possibilities for direct interaction in digital painting.

However a significant limitation of most digital painting tools is that the final product is just a digital image (typically an array of RGB color values). All the colors, textures and lighting effects that we see when we look at the digital painting are "baked in" to the image by the painter. In contrast, when a painter works with real tools and media, the color, gloss, and textural properties of the work are a natural byproduct of the creative process, and lighting effects such as highlights and shadows are produced directly through interactions of the surface with light in the environment.

In this paper we introduce tangiPaint, a new tablet-based digital painting system that attempts to bridge the gap between the real and digital worlds. tangiPaint is a tangible painting application that allows artists to work with digital media that look and behave like real materials. Figure 1 shows screenshots from the tangiPaint application implemented on an Apple iPad2.

In Figure 1a an artist has painted a number of brushstrokes on a blank canvas. Note that in addition to color, the strokes vary in gloss, thickness, and texture, and run out just as if they were real paint laid down with a real brush. The paints also layer and mix realistically as they would on a real canvas.
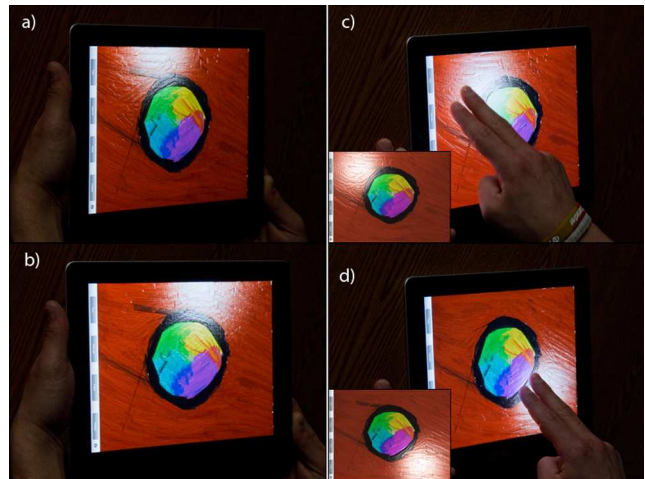

Figure 1. Screenshots of paintings created using the tangiPaint system. Note the gloss and relief of the brushstrokes and the texture of the underlying canvas. The system allows direct interaction with the "painted" surface both in terms of paint application and manipulation of surface orientation and lighting.

Figure 1b shows that the system also incorporates the capabilities of the Tangible Display System introduced by Darling and Ferwerda [1]. Thus by tilting the device the artist can change the relationship between the digital painting and the virtual light source that illuminates it. This action reveals both the texture of the strokes and canvas and the gloss properties of the paints just as manipulating a real painting would. Figures 1c and 1d show that the artist can also position the light source with a simple hand gesture akin to moving a real light source The tangiPaint system represents a significant first step toward developing digital painting tools that allow artists and others to work with digital art media that look and behave like real materials.

In the subsequent sections of the paper we describe the design, functions, and use of the tangiPaint system. We conclude by discussing the contributions and limitations of the work and directions for future research and development.

Background

Arguably, the earliest example of a digital painting system is Sutherland's Sketchpad graphics system [2]. Among its many innovations was the ability to draw with a light pen on the face of a large computer-driven oscilloscope.

In the 1970's work at Xerox PARC on graphical user interfaces included the development of a GUI-based painting tool that used the raster display on the Xerox Alto computer [3]. Many of the ideas developed in the Alto painting system found their way into the MacPaint application bundled with the first Apple

Macintosh computers [4]. These included standard methods for selecting different substrates or backgrounds, methods for selecting tools such as brushes, pencils, and erasers and modifying their properties, and methods for sampling from existing paintings to specify tool properties. Continual innovation has led to current state-of-the-art applications such as Adobe Illustrator, Corel Painter, AutoDesk SketchBook and ArtRage.

While the features of mainstream digital painting systems continue to improve, the tools and media in these systems do not necessarily behave the way real materials would. A number of researchers have been trying to improve the physical fidelity of digital painting systems.

Curtis et al. [5] developed an interactive painting system for watercolor paints that used diffusion equations to simulate the interactions of liquid and paper. Rudolf et al. [6] developed a system to simulate the behavior of wax crayons taking into account the topography of the drawing surface. Chu and Tai [7] developed a deformable brush model that is a component in a brush-and-ink painting system.

Most closely related to our work is the work of Baxter et al. [8] who developed a sophisticated digital painting system called IMPaSTo that simulates the behavior of thick oil paints applied with a brush on canvas. Baxter's system models the layering properties of paint using an advection model to create height fields and the Kubelka-Munk model to simulate the mixing of colored pigments. A physical brush model allowed the buildup of complex surface textures.

Interactive graphical input tools

Paralleling the developments in digital painting software have been innovations in graphical input devices to support direct interaction. Sutherland's light pen notwithstanding, early digital painting systems commonly used a mouse for input control. The desire for absolute positioning led to the development of graphical input tablets [9], both with puck shaped controllers and more natural stylus shaped devices. Multiple buttons and pressure sensitive tips provided for more expressive control [10], but the artist's "pen/brush" (tablet) and "canvas" (display) were still physically separate.

Touchscreens overcame this limitation, providing a direct visual connection between the input controller and the display device [11]. Pen based touch screens initially provided better resolution and performance but recent improvements in the technology [12] and the emergence of mobile devices and tablet PCs have led to the dominance of finger-driven touchscreens. The development of multi-touch devices [13] has also provided a rich new vocabulary for user interaction.

Tangible display systems

Darling et al. [2009] have been developing tangible display systems that support natural interaction with virtual surfaces. The first generation "tangiBook" shown in Figure 2 was based on an off-the-shelf laptop computer that incorporated an accelerometer and a webcam as standard equipment. Custom software allowed the orientation of the laptop screen and the position of the observer to be tracked in real-time. Using this information, realistic images of surfaces with complex texture and material properties, illuminated by environment-mapped lighting, were rendered to the

Figure 2. Image sequence showing a model of an oil painting being displayed on the tangiBook laptop. Custom software allows the orientation of the laptop screen and the position of the screen observer to be tracked in real-time. Tilting the laptop or moving in front of the screen produces realistic changes in surface lighting and material appearance.

screen at interactive rates. Tilting the laptop or moving in front of the screen produced realistic changes in surface lighting and material appearance. Thus the tangiBook allowed virtual surfaces to be observed and manipulated as naturally as real ones. More recently these researchers have been developing second generation tangible displays based on the iPod and iPad systems that provide even more natural form factors for direct interaction.

In this project we leverage the capabilities of tangible displays to provide the experience of direct manipulation of the paintings created by our system.

System Design

The tangiPaint system is comprised of several components that work together to allow the creation, modification and viewing of tangible digital paintings. In the following section we describe the design and functionality of each of the components.

Substrate

Substrates are the bases to which paints are applied. Each substrate is composed of a color map and a height map, as shown in Figure 3, and it is the combination of these two components through interaction with the lighting model that describes the rendered appearance of the surface. As the substrate is modified, such as by adding paint or digging grooves, the height map is changed appropriately. Height values are then converted into normal vectors for rendering by taking differences of neighboring locations. Standard substrates such as canvas and paper are included in the system, and new substrates can be imported by the user.
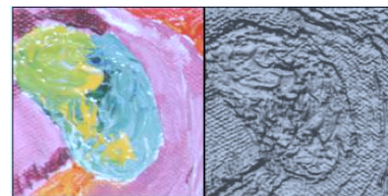

Figure 3. The left image shows a section of the color map of a painting and the right image shows the associated height map.

The substrate is separated into a dry and wet layer. The dry layer contains the color and orientation of the base material (e.g. canvas), which is not modified directly. Instead the user interacts with the wet layer to apply new paint to the substrate. The wet and dry color layers and height maps are blended to produce the current painting as shown in Figure 4.
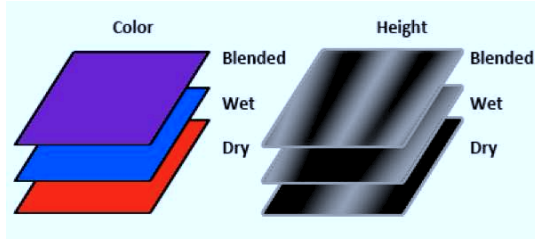

Figure 4. The color and texture (height) layers that make up a painting.

Paint

In real paintings, the color, gloss, and thickness of the paint determine how it interacts with light, and blends with other colors. In the tanigPaint system these properties can be varied to produce a range of effects. Currently the tangiPaint system is geared toward producing oil-like paintings and to model the behavior of these paints, the system implements a simplified, opaque Kubelka-Munk model to perform real-time subtractive color blending. The equations for the model are shown in Figure 5. Colors specified in RGB are converted into absorption/scattering (K/S) ratios for each channel, and then weighted averages are taken based on the volume of paint being mixed. These values are then converted back to RGB for rendering and display.

R	Reflectance
K	Absorbtion coefficient
S	Scattering coefficient
A	Absorbance
h	Height

Table 1: Kubelka-Munk Parameters

$$\left(\frac{K}{S}\right)_R = \frac{(1-R_R)^2}{2R_R} = A_R$$

$$\left(\frac{K}{S}\right)_G = \frac{(1-R_G)^2}{2R_G} = A_G$$

$$\left(\frac{K}{S}\right)_B = \frac{(1-R_B)^2}{2R_B} = A_B$$

$$A_{R,mix} = h_{wet} * A_{R_{wet}} + h_{dry} * A_{R_{dry}}$$
$$A_{G,mix} = h_{wet} * A_{G_{wet}} + h_{dry} * A_{G_{dry}}$$
$$A_{B,mix} = h_{wet} * A_{B_{wet}} + h_{dry} * A_{B_{dry}}$$

$$R_R = 1 + A_R - \sqrt{A_R^2 + 2A_R}$$

$$R_G = 1 + A_G - \sqrt{A_G^2 + 2A_G}$$

$$R_B = 1 + A_B - \sqrt{A_B^2 + 2A_B}$$

Figure 5. Simplified Kubelka-Munk model used for color blending.

Brushes

Like the substrate, brushes are also composed of a color map and a height map and each location in the combined maps represents a bristle of the brush. Brushes can be flat or rounded and of varying size. As a brush is stroked along the substrate, the heights and colors of the wet layer of the substrate interact with the colors and heights of the brush. Each bristle interacts with the substrate independently, which produce the textured "stroke" effect seen in real paintings, as shown in Figure 6.
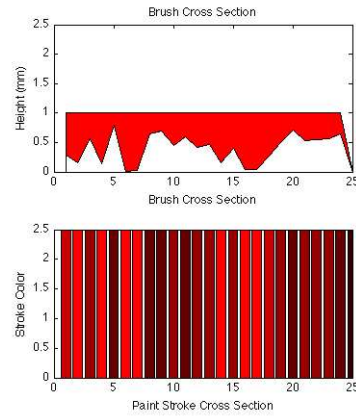

Figure 6. Relationship between the heights and colors of the brush that produce stroke textures.

Brush stroke texture and paint transfer are simulated by calculating the volume of paint transferred by each bristle. The brush can run out of paint, but can also pick up and mix with wet paint that can be blended into the brush. These interactions simulate realistic impasto and color blending effects.
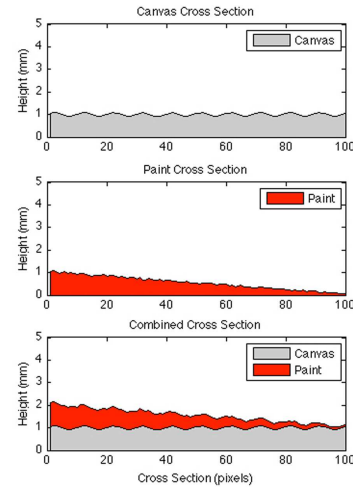

Figure 7. Cross sections though a textured substrate, paint stroke, and a painted substrate showing how the colors and textures of each combine to produce the final surface appearance.

Figure 7 shows how the substrate, paint, and brush interact to produce the textures and colors of the painted surface. The top panel shows a cross section through an unpainted canvas illustrating the height variations that define the substrate texture.

The middle panel shows an idealized brush stroke, illustrating both the initial thickness and runout of the paint as the brush is drawn across a substrate, and the texture imparted by the brush. Finally, the lower panel shows how the paint and substrate interact to produce the final colors and textures of the painting. The paint fills in the substrate texture and in turn has its texture modulated by the substrate.

Depending on the relative heights and thicknesses of the two layers, the substrate may be completely or partially obscured. Additional paint layers and brushstrokes then interact with the existing layers, adding or blending depending on the amount of paint on the brush. Constraints on the layering process place realistic limits on the thickness of the paint layer, transitioning from adding to blending/moving as thickness increases.

Lighting

It is the interactions of light with of the colored, textured surface created with the substrate, paint, and brush that give the painting its visual richness. To simulate light/surface interactions we render images of the painted surface using a GPU-based graphics shader. The shader implements an isotropic version of the Ward light reflection model [14] that represents the surface BRDF using the three parameter model shown in Figure 8. Light from a distant point light source is scattered over the hemisphere defined by the surface normal according to three parameters, ρ_d, the diffuse reflectance factor, ρ_s – the specular reflectance, and α – the spread of the specular lobe. Together these three parameters define the color and intensity of the light reflected toward the camera by the surface. Changing the light direction, surface normal, or viewing direction will all affect surface shading.
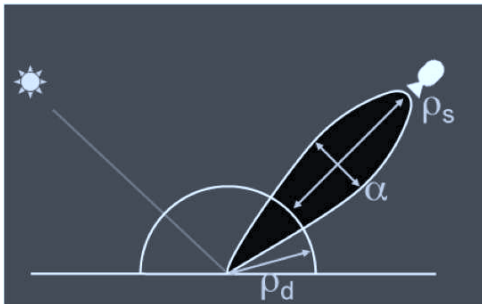

Figure 8. Surface shading model including light source, camera, and Ward light reflection (BRDF) model.

Interaction

The system utilizes the accelerometers built into mobile devices like the iPad and iPhone to interactively determine device orientation. As the device is moved, its orientation continuously updated and surface shading is re-rendered in real-time. As new paint strokes are added or material is removed, the colors and normal vectors of the virtual surface are modified in real time produce realistic renderings.
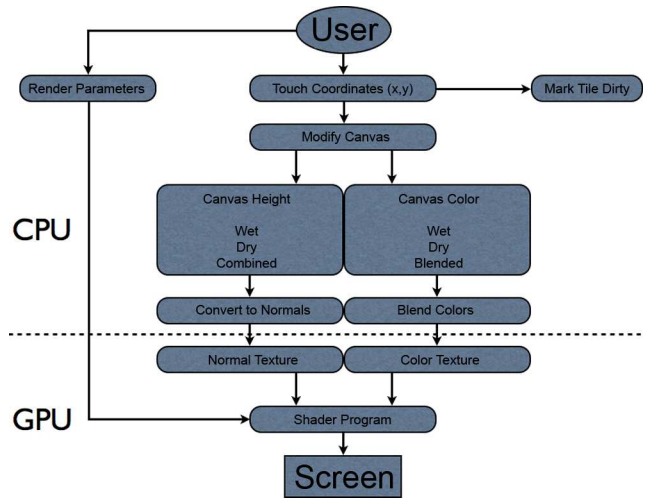
System Implementation

Figure 9. Flowchart showing implementation of the painting system. The rendering work is split between the CPU and GPU, and output to the display.

Figure 9 illustrates the rendering loop of the system. It utilizes both the CPU and GPU components to achieve the best performance and fastest response to user interaction. This loop runs continuously to display the most current lighting for each frame. Since the entire scene is affected by this lighting, the GPU renders each pixel in parallel, while the CPU handles sequential touch events.

As a user interacts with the system, through touches and swipes on the screen or setting various parameters, the canvas and rendering values are modified. As paint is applied or material is removed from the canvas, the color texture and height maps are modified and those locations are marked as being 'dirty', or changed. This dirty tile approach allows the system to only update the blended layer where needed, saving a significant amount of blending work and increasing performance. The heights and the colors of the modified wet layer and underlying dry layer of the canvas are then combined and blended before being submitted to the OpenGL shader programs. Any modified heights are then used to update the associated normal vectors for those locations. The normal vectors are calculated by taking the difference in height between neighboring locations.

These shader programs are executed on the GPU for every pixel. The updated textures are then passed to the final stage of the rendering process where a simplified Ward BRDF model is used to calculate the final color of each pixel. These final colors are output to the screen as the user interacts with the system in real-time.

Using the System

This section provides a walkthrough of the tangiPaint system and its user interface. The system menu bar and each menu are shown in Figure 10.
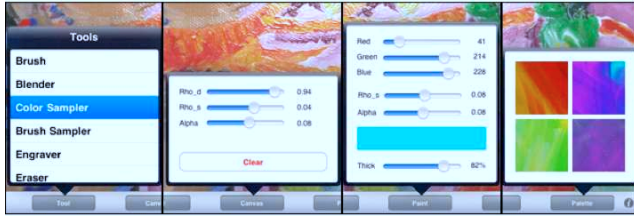
Figure 10. The user interface of the system with each menu expanded. From left to right is the tools menu, the canvas menu, the paint menu, and the mixing palette.
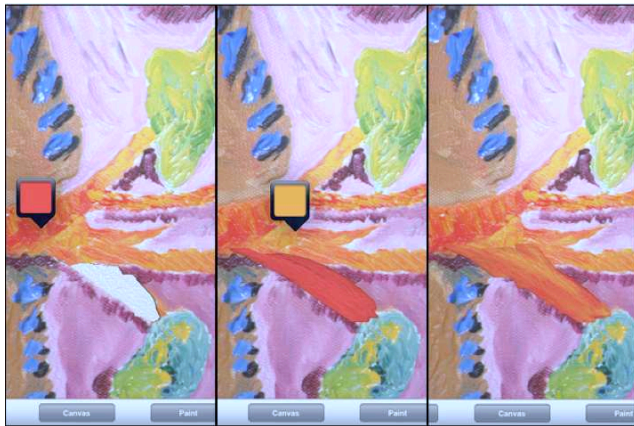

Figure 11. A painting being restored using colors selected with the eyedropper tool.

Figure 12. Subtractive color mixing using the simplified Kubelka-Munk model.

Choosing a substrate: A substrate can be selected from a pre-defined set of materials, such as canvas or paper. These have associated color and height maps. Users can also import new materials and base images into the system by providing their own color textures and height maps.

The reflectance properties of the substrate can also be modified in the application by using the options available in the canvas menu (Figure 10). The user can adjust the color and gloss using the ρ_d, ρ_s and α parameters of the Ward model. These parameters update in real-time and the user can view the effects on the canvas in the same screen.

Selecting a brush: Selecting the Tool menu (Figure 10a) presents a list of available tools. When the Brush tool is selected, that menu then drills down to the specific brush settings. Here the user can vary the size and shape of the brush, along with the pressure that it applies to the canvas and if the brush is automatically cleaned after each stroke.

Defining paint properties: By opening the paint menu, the user can choose the color and characteristics of the paint. Changing the red, green, and blue values updates a color preview, as shown in Figure 10. The user can also choose the gloss properties of the paint using the Ward parameters. The thickness of the paint can also be set, which allows for thin, evenly spread paint to thick globs.

In addition to choosing colors by value, an eyedropper tool allows the selection of an existing color from the canvas. The eyedropper is first selected from the tool menu, and then a point on the canvas is selected. A small color preview is shown next to the selected location. The eyedropper is shown in Figure 11, where a painting is being restored.

Paint mixing: Paint is automatically blended as strokes of different color and thickness are mixed. The resulting color is a function of the input colors, gloss parameters, and thicknesses, as explained in the previous section. Figure 12 illustrates blending of multiple colors.

Users can use the mixing palettes provided in the 'Palette' menu, shown in Figure 10. Here they can combine paints of different color, glossiness, and thickness to preview before adding to their work. These palettes can also be used to save colors and the user can return to sample them at any time.

Interactive viewing: As the screen is tilted, the device senses the movements reported by the accelerometer. The rendering loop uses the most current values from each to calculate the orientation of the surface and the elevation of the light source to calculate surface shading. Figures 1a and 1b illustrate this direct interaction with the device and its effects.

Interactive lighting: The scene light can be repositioned by the user to achieve a variety of lighting effects. Using a two-finger gesture, the azimuth and elevation of the light source can be changed interactively. This interaction is shown in Figures 1c,d. This feature allows the user to work on a flat surface while experimenting with directional lighting effects.

Sample Paintings

The sample paintings in this section illustrate some of the capabilities and uses of the tangiPaint system.

The painting shown in Figure 13 illustrates the buildup of impasto though multiple strokes and layers. The paint is glossy and the two panels in the figure show the painting illuminated from two different directions. This figure shows how the surface colors and textures interact with lighting to reveal the rich structure of the artist's brushwork.

The second painting, in Figure 14, shows some of the same textural and lighting effects illustrated by the first painting, but also shows the effects of color blending, that can be seen in the interior brush strokes that define the shape of the apple.

Finally, Figure 15 shows the possibilities of importing and modifying existing painting models. The painting shown Figure 15 (left) was imported from a model created by a laser scanner. The imported painting has its own color and texture properties. In

Figure 15 (right) new paint is added on top of this complex substrate and is blended both in color and texture. This gives the impression that this new paint is part of the original image. With this capability the tangiPaint system offers the possibility for doing digital restorations of damaged artwork.
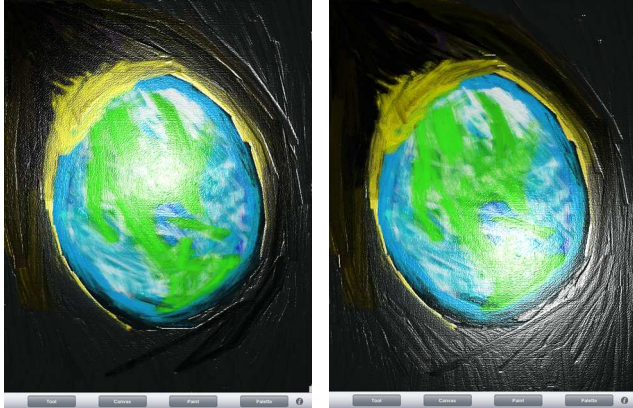

Figure 13. Sample painting showing impasto, gloss, and lighting effects.

Figure 14. Sample painting showing gloss, lighting, and color blending effects.

Figure 15. Overpainting an imported painting model. Note the realistic blending of colors and textures.

Conclusions and Future Work

In this paper we have introduced tangiPaint, a tangible digital painting system. The system allows artists to directly interact with realistic substrates and paints to create surfaces with rich colors and textures. Realistic lighting effects are achieved through a GPU-based shader. Direct manipulation of the painting object and its lighting are provided though accelerometer and touch-based interaction. The tangiPaint system represents a first step toward developing tools and media for digital artists that look and behave like real materials.

While in its current form the system provides novel capabilities there is still much room for improvement and development. In future versions we hope to provide a wider array of paint types (ink, watercolor, etc.) and to more accurately model the diffusion and mixing of translucent paint media. Improvements can also be made to the brush model to more accurately simulate the behavior of real brushes. Other tools such as pens, pencils, crayons, and palette knives could be added as well. In terms of illumination it would be interesting to provide rendering using real-world illumination maps that could even be captured by the device's camera. In interaction, it might also be possible to take advantage of the device's camera to track the user and provide viewpoint specific rendering. Finally all the concepts developed for the tangiPaint application could also be applied to similar tangible interactions systems for digital clay modeling, woodcutting, and engraving.

References

[1]   B. A. Darling and J.A. Ferwerda, The tangiBook: a tangible display system for direct interaction with virtual surfaces. Proceedings IS&T 17th Color Imaging Conference, 260-266, (2009).

[2]   I.E. Sutherland, Sketchpad, a Man-Machine Graphical Communication System. Ph.D. Thesis, MIT, 1963.

[3]   A. Kay and A. Goldberg, Personal Dynamic Media, Computer, 10(3), 31-41, (1977).

[4]   B. Atkinson, *Macpaint*. Apple Computer, Cupertino, CA, (1985).

[5]   C. J. Curtis, S.E. Anderson, J.E.Seims, K.W. Fleischer, and D.H. Salesin, Computer-generated watercolor. Proceedings SIGGRAPH '97, 421–430, (1997).

[6]   D. Rudolf, D. Mould, and E. Neufield, Simulating wax crayons. Proceedings Pacific Graphics, 163–172, (2003).

[7]   N.S. Chu and C.L. Tai, An efficient brush model for physically-based 3D painting. Proceeding Pacific Graphics '02. (2002).

[8]   W. Baxter, J. Wendt, and M. C. Lin. IMPaSTo: a realistic, interactive model for paint. Proceedings NPAR '04. (2004).

[9]   M.R. Davis and T.O. Ellis, The RAND tablet: a man-machine graphical communication device, AFIPS Fall Joint Computer Conference #26, part 1, 325-331, Spartan Books, Baltimore, Maryland, (1964).

[10] http://www.wacom.com/

[11] E.A. Johnson, E.A. (1967). Touch Displays: A programmed man-machine interface. Ergonomics 10(2), 271-277. (1967).

[12] R. Potter, L. Weldon, and B. Shneiderman, Improving the accuracy of touch screens: an experimental evaluation of three strategies. Proceedings ACM CHI '88. 27–32. (1988).

[13] B. Buxton, Multi-touch systems that I have known and loved. http://www.billbuxton.com/multitouchOverview.html

[14] G. Ward, Measuring and modeling anisotropic reflections. Proceedings SIGGRAPH '92, 265-272, (1992).