

PM

Pen and Touch Interactions on Graphical Operations

MASTER PROJECT

Roman Chubatyy

MASTER IN COMPUTER ENGINEERING



UNIVERSIDADE da MADEIRA

A Nossa Universidade

www.uma.pt

December | 2017

Pen and Touch Interactions on Graphical Operations

MASTER PROJECT

Roman Chubatyy

MASTER IN COMPUTER ENGINEERING

SUPERVISOR

Diogo Nuno Crespo Ribeiro Cabral



Pen and Touch Interactions on Graphical Operations

Roman Chubatyy

Constituição do júri de provas públicas:

Karolina Baras, Prof.^a Auxiliar da Universidade da Madeira, Presidente

Pedro Filipe Pereira Campos, Prof. Auxiliar da Universidade da Madeira, Vogal

Diogo Nuno Crespo Ribeiro Cabral, Prof. Auxiliar da Universidade da Madeira, Vogal

Dezembro 2017

Funchal – Portugal

Abstract

Pen and touch input on horizontal tablet devices imitate operations on paper performed with two hands, one of them holding a pen. In the physical world, a pen held on dominant hand might be used for writing and drawing on paper, while the non-preferred hand manipulates paper to support pen gestures. Modern tablets have evolved the way, so it is possible to use pen and touch simultaneously, allowing to create new tools using such combination.

Graphical tablets support multiple finger inputs, but every of them is limited to its X and Y position, while the pen also allows to measure its tilt and orientation angles relatively to tablet, and its analog pressure.

Past research showed that it is possible to use hand's thumb on the side of screen to change the actions of the pen. With pen, users are able not only to draw around the screen, but also move and copy items around, zoom them in and out, set up some of their characteristics like size and value, between others. While drawing using a pen, fingers can make various transformations for richer experience on drawing tools.

In this work, the openFrameworks toolkit was used and combined with other APIs that receive and process Wacom Cintiq 13HD tablet's pen and touch data, to create a high-fidelity user interface prototype that supports graphical operations performed with pen and fingers, separated and simultaneously. Besides of single input tasks that allow to create and transform graphical operations, there were introduced new bimanual tasks that are designed to be more natural than those with keyboard and mouse.

The evaluation results show that pen and touch interactions were faster for drawing a line but slower for selecting a color, when compared with mouse interactions.

Keywords

Graphical Tablets

Digital Pen

Multitouch

Pen Interfaces

Pen and Touch Interactions

Graphical Interfaces

Resumo

Interações com tecnologias de caneta e toque em superfícies horizontais imitam operações bimanuais em papel, com uma caneta numa das mãos. No mundo físico, uma caneta pode ser usada na mão dominante para escrever e desenhar em papel, enquanto a mão não preferida manipula o papel apoiando os gestos da caneta. As tablets modernas evoluíram de modo que é possível usar caneta e toque simultaneamente, permitindo criar novas ferramentas que usam esta combinação.

As tablets gráficas suportam múltiplas entradas de dedo, mas cada uma destas está limitada às suas coordenadas X e Y, enquanto a caneta permite também medir os seus ângulos de inclinação e orientação em relação ao tablet e a sua pressão analógica.

Investigação anterior demonstrou que é possível usar o polegar no lado do ecrã para alterar as ações da caneta. Com a caneta, os utilizadores são capazes não só de desenhar em torno do ecrã, mas também mover e copiar itens, ampliá-los e reduzi-los, configurar algumas de suas características, como tamanho e valor, entre outros. Ao desenhar usando a caneta, os dedos podem fazer várias transformações para uma experiência mais rica em ferramentas de desenho.

Neste trabalho, foi utilizado a plataforma openFrameworks, combinada com outras APIs que recebem e processam os dados de caneta e toque do tablet Wacom Cintiq 13HD, para criar um protótipo de alta fidelidade que suporta operações gráficas realizadas com caneta e dedos, separados e simultaneamente. Além de tarefas de modalidade única que permitem criar e transformar operações gráficas, foram introduzidas novas tarefas bimanuais que são desenhadas para serem mais naturais do que aquelas com teclado e rato.

Os resultados da avaliação mostram que as interações de caneta e toque foram mais rápidas para desenhar uma linha, mas mais lentas para selecionar uma cor, quando comparadas com as interações do rato.

Palavras - chave

Tablets Gráficas

Caneta Digital

Interfaces Multitoque

Interfaces com Caneta

Interfaces de Caneta e Toque

Interfaces Gráficas

Acknowledgements

Firstly, I want to thank my advisor Diogo Nuno Crespo Ribeiro Cabral for accepting to supervise my project and giving the materials needed. It was really an interesting experience in working that materials. I want to thank him for availability throughout some bumps that have happened in my work.

Thanks to all my university friends for believing in me, showing some interest and giving me some help whenever I needed.

Thanks to the users that have participated in user tests. Their availability for collaborating with me was one of the most important part that was needed to complete my work.

And most important, I give my thanks to my parents for giving me all the encouragement, belief and support. Without it, I would not be able to do my studies and thus I could not do my thesis.

This work was partially funded by Madeira-ITI, FCT/MCTES LARSyS (UID/EEA/50009/2013 (2015-2017)).

Contents

Abstract.....	1
Keywords.....	3
Resumo.....	5
Palavras - chave.....	7
Acknowledgements	9
Contents.....	11
List of Figures.....	13
1. Introduction	15
1.1 Contribution	16
1.2 Thesis Structure	17
2. Related Work.....	19
2.1 Pen and Touch Devices.....	19
2.2 Pen vs Touch.....	23
2.3 Pen + Touch Prototypes	24
2.4 Conclusions.....	28
3. Pen + Touch Interactions	29
3.1 Prototype implementation.....	29
3.2 User interface prototype.....	30
3.3 Single input operations	33
3.3.1 Selecting color on HSB ring.....	35
3.3.2 Rotating widgets with pen	37
3.3.3 Switching images on widgets using pen.....	38
3.4 Combining pen and touch	39
3.4.1 Scratching with finger	40

3.4.2	Resizing using pen and finger	40
3.4.3	Painting various widgets.....	41
3.4.4	Tool-assisted drawing with ruler	41
3.4.5	Bimanual straight-line tracing	43
3.4.6	Bimanual semi-ellipse tracing	44
3.4.7	Switching images using slider	44
3.4.8	Color palette	45
3.4.9	Opening menu from swipe gesture.....	47
4.	Evaluation.....	51
4.1	Tasks	51
4.2	Procedure	52
4.3	Results.....	54
5.	Conclusions	57
5.1	Future work.....	57
6.	References	59

List of Figures

Figure 1. Wacom Cintiq 13HD graphical table with its digital pen.....	16
Figure 2. One of the first touch screens, invented by Bob Boie in 1984.....	19
Figure 3. Neonode N1 Mobile Phone (2004), one of the first mobile phones to support touch interface.....	20
Figure 4. The first generation's iPhone (2007).	20
Figure 5. ActiveDesk (1985) in action	20
Figure 6. N-Trig, one of the modern non-Wacom pen and touch architectures.....	21
Figure 7. Internal composition of Wacom's pen [6].	22
Figure 8. Pen's pressure value typical changing during normal usage [15].	24
Figure 9. Various real-life pen and paper techniques studies by Microsoft Research team. [10].....	25
Figure 10. Tapping a widget with a pen while holding it with finger can stack other selected widgets beneath it. [10].....	26
Figure 11. While holding a widget with finger, the pen can brush some fragments of image of that widget. [10].....	26
Figure 12. Two fingers determinate the active canvas area, where pen options can be performed [13].	27
Figure 13. While holding a desired option with left thumb, moving the pen up and down can perform that function [14].....	28
Figure 14. Prototype architecture.	30
Figure 15. Some widgets being manipulated on graphical prototype.	30
Figure 16. The main prototype's canvas and its composition.	31
Figure 17. Dragging various squared widgets with fingers.....	34
Figure 18. Zooming a widget in using fingers	34
Figure 19. The HSB ring [21].....	35
Figure 20. Touching color ring with different orientation angles produces different colors.	36
Figure 21. Different shades of blue produced by different tilt and pressure of pen.	36
Figure 22. Tilt color palette positioned on a comfortable position.	37
Figure 23. Rotating a widget by spinning the pen around its puck.	38

Figure 24. Switching images on a widget can be performed by tilting the pen from side to side.	39
Figure 25. Scratching fresh paint using fingers.	40
Figure 26. Resizing a widget using pen and finger.	40
Figure 27. Painting two widgets chosen by fingers with a color chosen by pen tilt and pressure.	41
Figure 28 Two fingers indicate ruler's position and orientation.	42
Figure 29. When holding a ruler with a finger, tracing on top of the ruler will produce the ink right near ruler's surface.	43
Figure 30. A straight line can be traced also using a finger and pen.	43
Figure 31. It is possible to trace a semi-ellipse as well by choosing two points and tilting the pen within the needed angle and direction.	44
Figure 32. When holding a widget, users can move tiny slider using the pen to switch images.	45
Figure 33. Palette appears on screen on the finger's position.	46
Figure 34. The palette on right side of finger, if used on right side of screen.	46
Figure 35. Different saturations produced from different pen tilt on palette.	47
Figure 36. Palette appeared after moving the pen on right side from held widget.	48
Figure 37. Mouse prototype that allows limited mouse interactions for user testing. It features a color palette that works the same way as Microsoft Paint.	53
Figure 38. Tracing a line time comparison chart.	54
Figure 39. Picking a color from palette time comparison chart.	55

1. Introduction

Digital pen and finger touch are natural and familiar inputs for humans, as they imitate physical interactions, like well-known drawing on paper using pen or pencil. Modern touch devices support up to ten finger touch points, each of them having only its cartesian position, while pen inputs also provide pressure, tilt and two additional buttons. Digital pens are usually used for drawing if it were on a paper, while with fingers, it is possible to select options and move the things around the screen. There are also multiple finger manipulation and gestures that allow resizing thing, rotating them, and more.

The touch and pen technologies have been evolving very quickly, allowing new interactions, e.g., for annotation [22], media editing [24] and exploration of information [23], but only the recent ones, like graphical tablets by Wacom, permit use both kinds of interactions simultaneously and combine them in various ways so they can produce tools with good usability and performance [11]. But what kind of thing could we do if we used pen and touch inputs together? This is an unanswered question.

Bimanual interactions with pen on preferred hand and touching with other hand on horizontal surfaces can provide not only natural feeling of daily pen and paper usage, but also, if used with various kinds of media, like colors and images, we can think of many new ways to work with them.

This work presents different pen and touch interactions implemented on a prototype for graphical operations.



Figure 1. Wacom Cintiq 13HD graphical table with its digital pen

1.1 Contribution

As explained at the beginning of this section, the main goal of this project is to develop a prototype that produces some fluid and rich interactions from using pen and touch bimanual co-operation on a Wacom Cintiq [1] graphical tablet.

The developed prototype is a graphical software tool allowing manipulation of drawings pictures using pen and finger combined in a natural way.

The application prototype done with Microsoft Visual Studio C++ development environment, openFrameworks [2] and Wacom APIs [3] [4] allow to perform various styles of drawing, picture viewing, organizing and annotating that are performed not only by using pen input alone or touch input alone, but also exploiting those two kinds to inputs and perform combined interactions.

1.2 Thesis Structure

This document explains everything made within the academical year, from its beginning until its last point. It is divided by five chapters, each one of them is organized according to respective stage of development.

The second chapter is called Related Work, and it describes the different pen and touch technologies, and the past projects for similar tasks that helped me concentrating on achieving my goals.

The third chapter is called Pen & Touch Interactions, describes the prototype implementation, which styles interactions were chosen to implement, for each of those styles, there are some photos, advantages and disadvantages in terms of functionality, comfort and productivity and possible alternatives to those interactions.

In the fourth chapter, Evaluation, it is shown the results and analysis of the comparison between pen+touch interaction and mouse interactions.

In the final chapter we present the main conclusions and future work to be developed.

2. Related Work

Before starting creating new interactions that exploit combination of pen and touch interactions, related research was consulted.

This chapter talks about different multitouch and pen technologies used in all times, and how can it be used to create natural interactions.

2.1 Pen and Touch Devices

Touch devices like notebook touchpads, tablet computers including iPads already are very familiar today. The devices that support finger touch input first appeared yet in 1965, and multitouch support appeared in 1984. These devices already used capacitance instead of optical detection. With touch technology evolving to multitouch, touch interfaces were available on ATMs, active desks, tablets, and since 2007 they are available much more widely on smartphones and tablet PCs. Touch devices were sensing not only fingers, but also blobs, pucks and digital pens – one of the most important solutions nowadays [5].

In figures below there are some of most important milestones in touch screen evolution in devices used today mostly, done by Bill Buxton [9] [18] [19] on his research.



Figure 2. One of the first touch screens, invented by Bob Boie in 1984.



Figure 3. Neonode NI Mobile Phone (2004), one of the first mobile phones to support touch interface.



Figure 4. The first generation's iPhone (2007).

Because of popularity of pen and touch interactions most of modern active desks, mobile phones and tablet PC, for example, Microsoft Surface, had dual-mode digitizers to sense pen and touch separately and produce rich bimanual interactions, but it has some downsides that will be explained in next subsection.

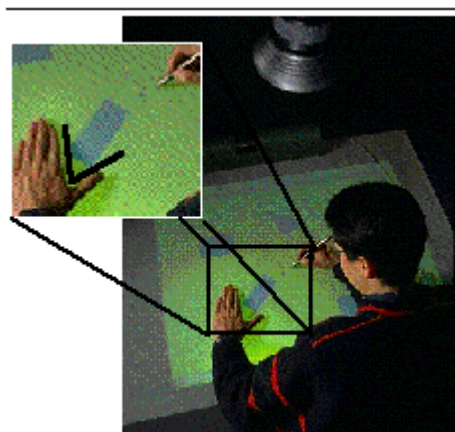


Figure 5. ActiveDesk (1985) in action



Figure 6. N-Trig, one of the modern non-Wacom pen and touch architectures.

2.2 Digital pen hardware

Now we will compare various digital pen hardware by their architectures.

There are generally four types of pen architecture: the capacitive pens, the electromagnetic pens, the optical pens [16] and camera pens [17].

Capacitive pens [8] use the skin conductance, i.e. detecting the pen and other objects by detecting low electronic signals, usually conducted by the human skin. They can be found on many old PDA's and smartphones, on many modern Android and iPad tablets, and on some game systems like Nintendo DS, 3DS and Wii U.

These pens react to the touch screen the same way as fingers do, not providing any additional functions because of being identical to finger, so there is no pressure measuring and palm rejection support on those devices.

Wacom provides an electromagnetic architecture initially used in Rand tablets [7] to detect the pen input and its additional important functions. Here are those principal functions [8]:

Pressure sensitivity: The most important feature that a Wacom-based digitizer layer offers is the ability to detect different levels of pressure. For example, Microsoft's Surface Pro Pen advertises 1024 levels of pressure, and Wacom's pressure is counted until 2014. The screen

doesn't detect how much pressure is put on it, instead, the tip of the pen retracts while pushing down harder on the screen, and the digitizer layer can detect a difference in signals.

Palm rejection: When using stylus to draw on your screen, the tablet can perform so called "palm rejection", ignoring touches and allowing palm to rest on the screen. For developers, it affects so called "confidence" characteristics that is affected by that palm rejection and allows to program an action that should be performed in that case.

Other Additional features: These types of styluses can just do more things. For example, there is an eraser on the other end, and flipping it over and rubbing the eraser on the screen will send an "erase" signal to erase things drawn in drawing applications. The digitizer layer can also detect when hovering the pen over the screen, allowing to perform hover actions. There are two programmable buttons as well. As default setting, pressing those buttons performs on Windows right-click or double-click functions.

Though, applications must be coded to detect this information. For example, it is not possible to use different levels of pressure in Microsoft Paint. And a driver should be installed on the system to support Wacom's API development.

The figure below shows the important component of Wacom electromagnetic pen [6].

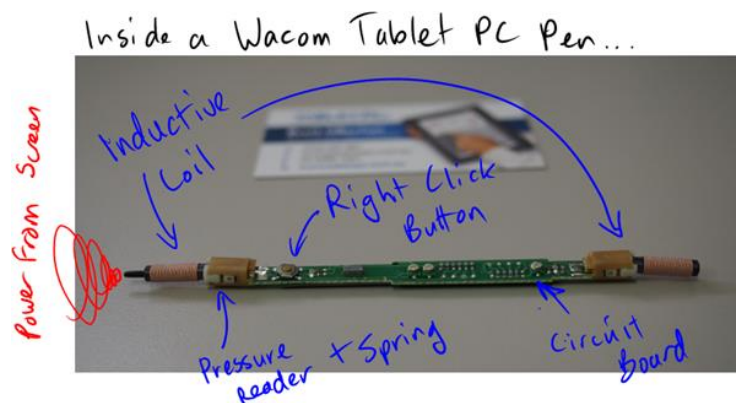


Figure 7. Internal composition of Wacom's pen [6].

The optical pen hardware [16], which is generally developed by MultiTaction, uses the touch screens based on infrared light called "MT cells" that are capable of detecting pens, fingers and other supported objects, and distinguish them.

The camera pen hardware [17] can work with special software installed, but require a piece of paper for working. Each piece of paper has a dot pattern, recognized by the camera, helping the users to fill digital forms.

2.3 Pen vs Touch

In a table below, we are going to make some comparison of pen and touch input in terms of the most important characteristic that are useful to choose interactions to implement and test.

Property	Pen	Touch
Contacts	A single well-defined point.	Up to 10 contact regions.
Occlusion	Small (pen tip), but hand still occludes screen.	Moderate (fat finger) to Large (pinch, palm, whole hand)
Precision	High - Tripod grip & lever arm for precision, writing, sketching. High sensing resolution.	Moderate, which can cause accidental inputs. Fat fingers cause low precision.
Hand	Preferred hand	Either or both hands
Elementary Inputs	Tap, drag, draw path, crossing	Tap, hold, drag, pinch, 2-finger hold (thumb + index)
Intermediary	Mechanical intermediary Takes time to unsheathe the pen. Pen can be forgotten. Two buttons.	None: bare-handed input. Nothing to unsheathe or lose. No lever arms. No buttons.
Acquisition Time	High (first use: unsheathe pen). Moderate on tucking pen between fingers.	Low: no mechanical intermediary to acquire.
Activation Force	Non-zero Tip switch/ minimum pressure.	Zero. Contact area (a proxy for pressure) often can be sensed.

False Inputs	Palm rejection (while writing) Palm triggers accidental inputs, fingers drag on screen, etc.	Midas touch problem Fingers brush screen, finger rests on screen while holding it.
---------------------	---	---

Table 1. Pen and touch input comparison [11] [12]

The naturality of pen's pressure input is that it is simulating force applied to the surface. It is measured by a spring inside pen's digitizing. So, if we want to store certain value of pressure in an operation other for drawing, it is needed to use an external input for this, which can be a touch with a finger, or one of the buttons on pen. The graph below shows how pen's pressure is changed when pen touches the screen [15]:

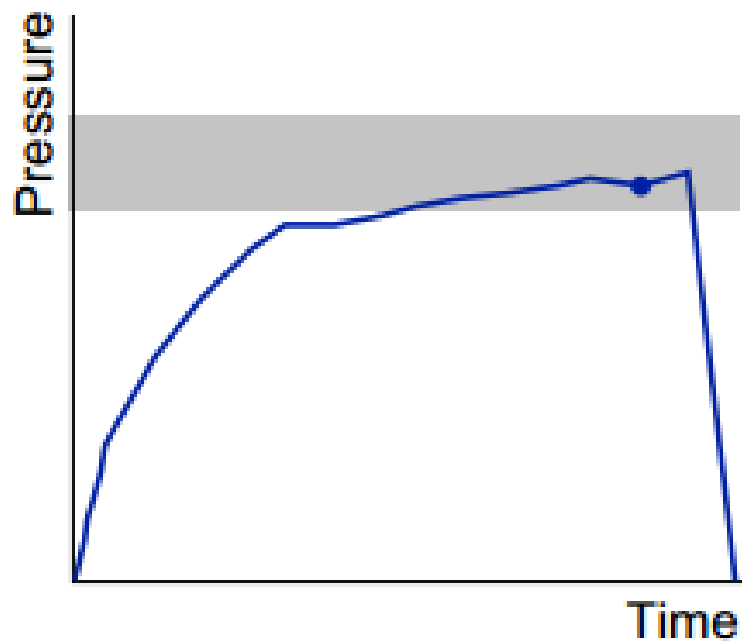


Figure 8. Pen's pressure value typical changing during normal usage [15].

2.4 Pen + Touch Prototypes

Previous research has shown how pen and touch interaction can be combined using cartesian coordinates (x,y) on a horizontal surface.

Manual Deskerty application [10] work talks about how real-life things that can be performed with preferred hand holding a pen and other hand doing useful things like handling the paper can be implemented on a haptic device like Microsoft Surface.

Here, the researches started with observing various things that people do usually with two hands and some physical objects including paper, pen, scissors, drawing tools, each one with its own function. They have selected some of them that allow easy implementing on tablets that use pen and finger for those goals. They are shown on the figure below.

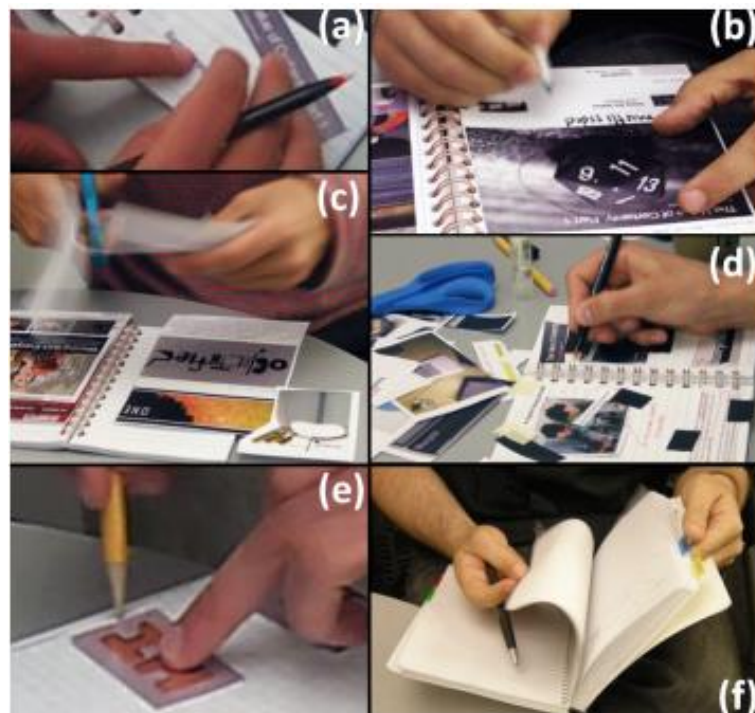


Figure 9. Various real-life pen and paper techniques studies by Microsoft Research team. [10]

Here is the official description of each operation illustrated: “(a) Users tuck the pen between fingers while manipulating items. (b) Thumb and forefinger grasp an item while writing about it. (c) Cut scraps fall onto the work surface. (d) Users often pull tools and new content onto the notebook from above. (e) Drawing a border around a clipping by holding it with the nonpreferred hand while using its edge to constrain the path followed by the pen held in the preferred hand. (f) Tearing a page by anchoring it with a thumb while pulling it away using fingers of the opposite hand.” [10]

After learning about those operations and working with core tasks where “pen writes, touch manipulates” [10], such as manipulating, zooming, selecting objects and leaving ink strokes with pen, new bimanual graphical operations have been implemented.

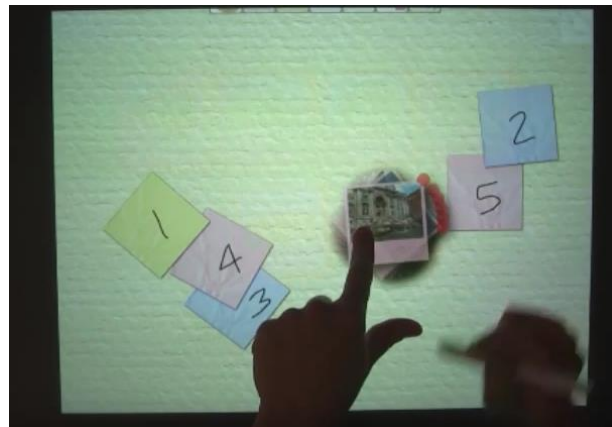
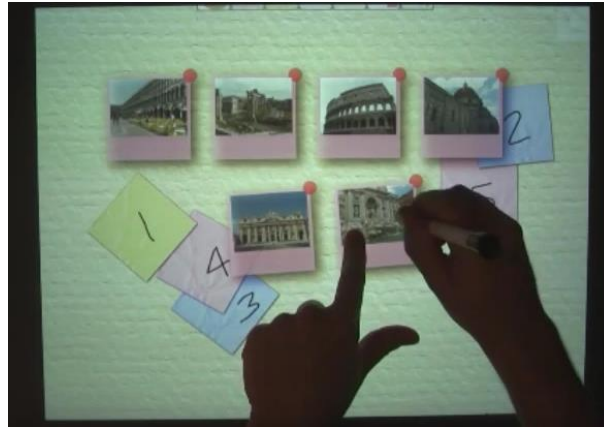


Figure 10. Tapping a widget with a pen while holding it with finger can stack other selected widgets beneath it. [10]

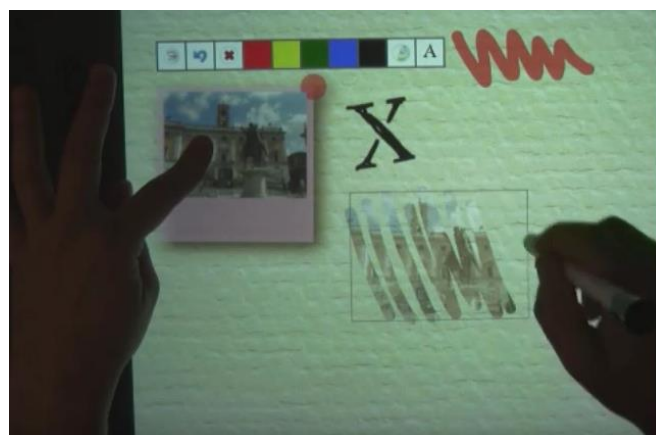


Figure 11. While holding a widget with finger, the pen can brush some fragments of image of that widget. [10]

WritLarge [13] is a prototype that allow to convert texts, canvases, and tables into well-organized printed structures by zooming that are with two fingers of non-preferred hand and performing operations on menu using pen or finger on preferred hand.

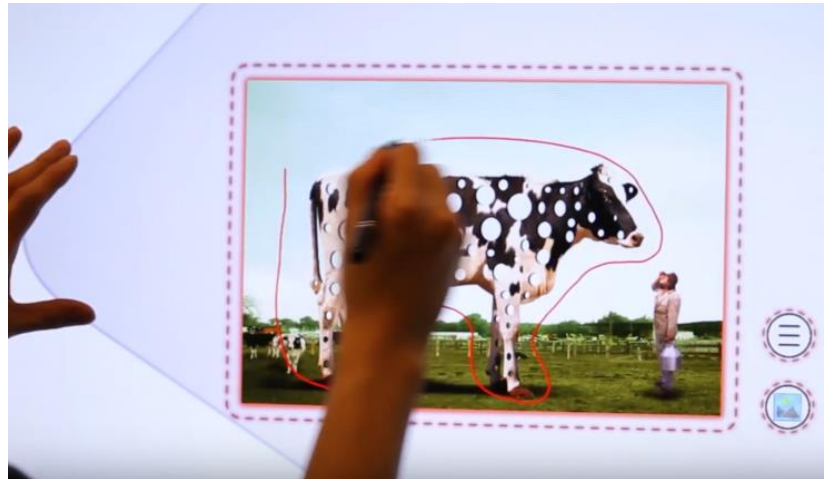


Figure 12. Two fingers determinate the active canvas area, where pen options can be performed [13].

The Thumb+Pen prototype [14] works together with simple business tools, mostly for working with spreadsheets and web browsers. Using just a thumb of non-preferred hand, users can select between various tools to be performed with pen, like clipboard copy/ paste, zooming in/out and increasing/decreasing values on tables by sliding the pen up or down, besides of simple inking. There are some options that can be adjusted by moving thump and/or pen horizontally and vertically, meanly those from browser, that can be activated with keyboard Ctrl, Alt and Tab keys. One of the main benefits of this prototype is that it allows to perform business actions with more relaxing on handheld tablets.

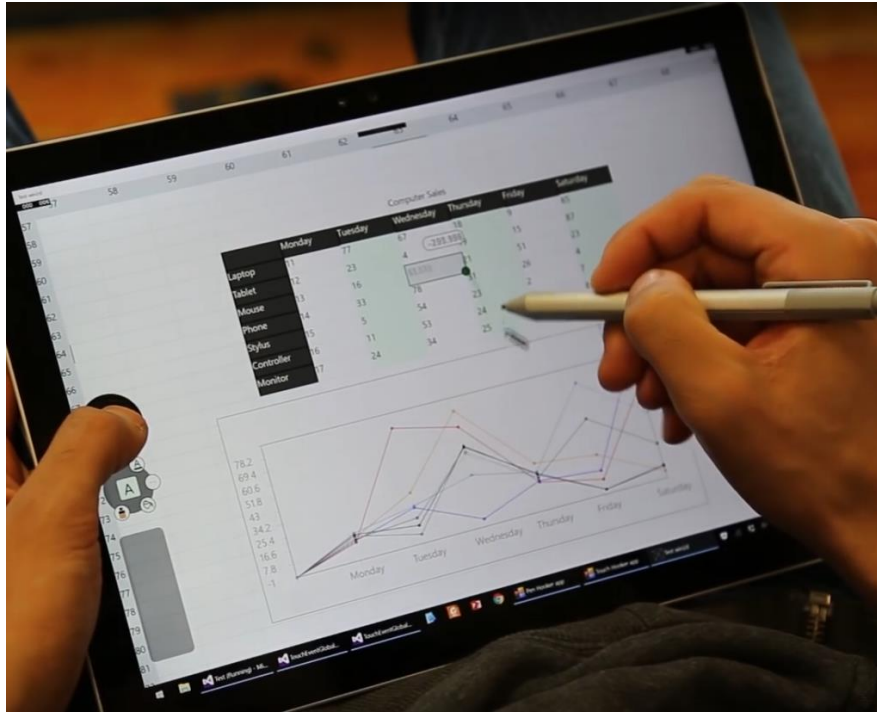


Figure 13. While holding a desired option with left thumb, moving the pen up and down can perform that function [14].

2.5 Conclusions

In this chapter, we have shown which benefits do pen and touch interactions, and checked some of research prototypes that already exploit pen and touch simultaneous interactions. However, they are limited to use screen cartesian coordinates. In this work, we go further by combining pen pressure and tilt combined with cartesian coordinates of pen and fingers.

3. Pen + Touch Interactions

The developed prototype allows pen drawing on screen and manipulation of widgets that can hold images to be viewed, scaled, rotated and annotated using not only single input operations but also by exploiting combinations of pen and touch inputs and producing rich, powerful and easy-to-use tools that allow natural interactions with two hands, one of them holding a pen.

3.1 Prototype implementation

The current project has been done for Wacom's Cintiq 13HD graphic tablet, which allows to detect pen and touch API.

The graphic tablet that used during this project support up to 10 finger touch points, this number is the same as the number of fingers of a human, and a pen with two touch sides – the main one, and the eraser one, making it look like an actual pencil, with support of pressure, tilt and two custom auxiliary buttons.

A graphical software prototype that accesses Wacom pen and touch data has been coded on C++ programming language, using Microsoft Visual Studio software, openFrameworks, bbTablet and Wacom Multitouch APIs.

The openFrameworks is a tool kit, which is written on C++ language, and provides a well-structured documentation with great diversity of functions implemented, makes the graphical prototyping work much easier to complete.

bbTablet (bare-bones tablet) API is based on the standard Wacom API for pen data, Wintab (written on C++), created by William Baxter. It is used so the prototype can receive pen location, pressure, tilt and buttons, and process it whenever interactions start and end. Unlike Wintab, bbTablet was made to be platform-independent and easier to use.

Wacom Multitouch API, also written on C++, is the official API made by Wacom manufacturers, which designed to program routines that should be executed once user touched the tablet with fingers. In those routines, fingers' positions are calculated during their contact with tablet. For each finger, there is also a special boolean data called confidence, which changes its value when pen is active for its palm rejection functionality.

The figure 14 shows how are the APIs connected in the application and which is their importance.

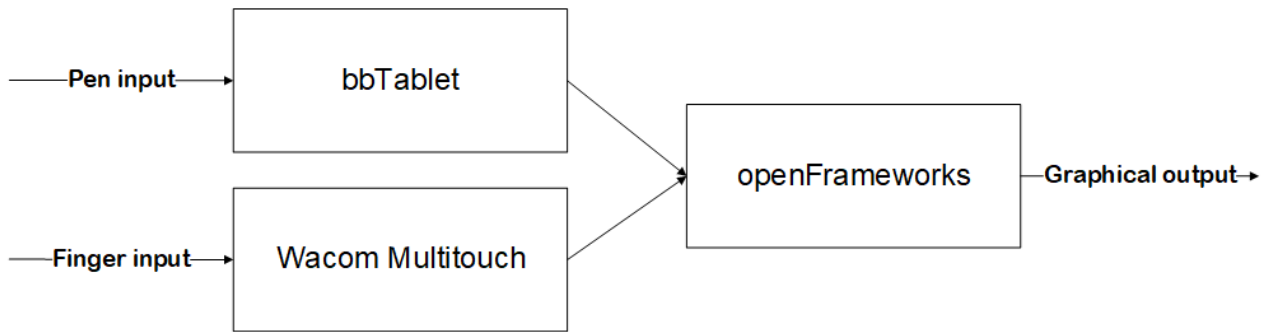


Figure 14. Prototype architecture.

3.2 User interface prototype

The prototype was designed to perform various graphical operations, including drawing, erasing drawings and adding rectangular mini-canvases that allow one to display pictures and sketches. On main canvas, users can draw several things, as if it was on paper. The same thing can be done on those widgets as well.



Figure 15. Some widgets being manipulated on graphical prototype.

To make it possible to implement those interactions, they have been structured in following way:

The drawings in the main canvas are composed of a set of points, i.e. traces of pen's tip, and each one of them has its unique color and size. Once a point is added to that set, a small circle

filled with selected color and with a small size (generally, up to 5 pixels, corresponding nearly to 1-2 mm) is drawn on the screen. The color can be chosen using one of three palette tools, while the size is determined by pressure applied to tablet's surface.

The multifunctional widgets that can be created and removed by the users to display their own drawings in a similar way to sticky notes on a table or a wall, and/or show some pictures that can be switched in two different ways: by rotating the pen, just like turning pages on an album, or using a slider movable with a pen and visible when widget is touched by a finger, with a size close to pen's puck.

Each widget is drawn as a rectangle, it has its start cartesian position, length and width in pixels, they can be rotated in a determinate angle using the secondary functionality of pen. If there are no images on a widget, it is shown in a filled rectangle, by default in white color, that can be change using main palette. The drawings have the same characteristics as the canvas drawing. But they are unique to each widget and displayed on top of images, if available. While moving widgets to other position or rotating them, those drawings do the same thing.

To be able to change the operations performable with pen and touch on the prototype, there is a small toolbar on the right of the screen. It can be used to switch operations for fingers and pen. To prevent accidental tools change, these tools can be activated only when touching them without pen or touch on main area already active.



Figure 16. The main prototype's canvas and its composition.

The items of the toolbar are marked as letters on the figure 16, and have the following functions:

- a) **Quit icon**: closes the prototype and returns to Windows.
- b) **New widget tool**: after activating, touching the screen on an area without any widgets will create new non-tilted widgets on touched positions with default size.
- c) **Ruler tool icon**: after activating, allows to perform line tracing bimanual interactions. After once ruler can be used, after activating in once more another time, users can trace straight lines and semi-ellipses with just a finger and a pen.
- d) **Color palette select tool**: after activating, allows to put color indicator and once, the color selector rings be put anywhere on screen. After activating twice, users can make a palette appear by tapping with finger.
- e) **Picture tool**: after activating, user can tap a widget, so a picture appears on it.
- f) **Finger brush tool**: after activating, user should be able to scratch paint with finger,
- g) **Active color indicator** (a smaller one): show which color is currently active and with which user can draw using a pen.
- h) **Primary color palette** (a big filled circle): after touching this area with pen, user should be able to choose a color with tilt and pressure functions. Tapping it with the finger will between pen main functions: drawing and others.
- i) **Delete icon**: tapping it will delete all drawing and widgets from screen. It is possible to drag a widget onto that icon to delete only that widget.

The activated tool is shown on toolbar in green color. During prototype implementation, ruler and color select tools needed to be enhanced in terms of usability, and after activating them twice, they will appear in red color. Primary function can be restored by tapping that icon once more.

The interactions are classified by inputs used as **single input operations** (i.e. the ones which are done with pen only or with fingers only), and **pen and touch input operations** (where pen and fingers are uses simultaneously).

The operations performed using the implemented interactions are the **drawing operations** (such as simple drawing with pen, scratching with finger tracing straight lines and semi-ellipses), **option selecting operations** (selecting tool items on toolbar or opening a menu through pen swiping), **color selecting operations** (choosing color on HSB ring or through the palette), **widget**

transforming operations (moving, rotating, resizing), and **image manipulation operations** (adding images, switching images by tilting the pen or through a slider).

3.3 Single input operations

To create a bimanual graphical operations prototype, it was needed first to familiarize on how the simple and core “pen writes, touch manipulates” tasks are created and work.

Usually, the tablet pen, when touches the tablet screen with its main tip, performs its basic operation – leaves its traces on screen in form of ink whose size is determined by pen’s pressure – the bigger it is, the bigger trace is left. The pen normally has 2048 levels of pressure, but its level is returned in normalized way – as a quotient between current and maximum pressure, measured as percentage.

Using pen’s eraser side, also with 2048 levels of pressure, it is possible to erase the pen’s trace. The bigger pressure is applied on eraser side, the bigger area can be erased.

When pen is being detected by tablet, all new fingers inputs detected by it are marked as “non-confident”. This is useful to prevent non-voluntary finger input.

The two buttons on the pen are useful for interactions like pressure-based stamp, since when the pen is taken away from tablet, its pressure level drops quickly to zero.

All other pen operations that are related with pressure, tilt and combinations with finger are explained in next sections.

The basic usages of fingers on tablet device are choosing items on menus, moving things, zooming them in and out, and many others, except the ones related for drawing.

The tablet supports up to 10 fingers, each of those fingers is given an ID number, depending of touching order. They have three states: down (i.e. just touched), hold, and up (i.e. just taken away from tablet). Also, there is a confidence bit on each finger, that is deactivated when pen is active. It can be used to filter unwanted inputs (for example, while drawing, an artist can put a palm onto the screen for comfort, and its input should be rejected). To prevent accidental tool change, the toolbar has been programmed so only confident finger with id 1 could activate it, and when that finger is released on that item (its state is up).

As for widgets, they can be created with tapping on the screen when new widget tool is active (can be various widgets), and an image loads when tapping a widget with image tool active. This operation also is filtered by confidence, because creating new widgets is unwanted when drawing.

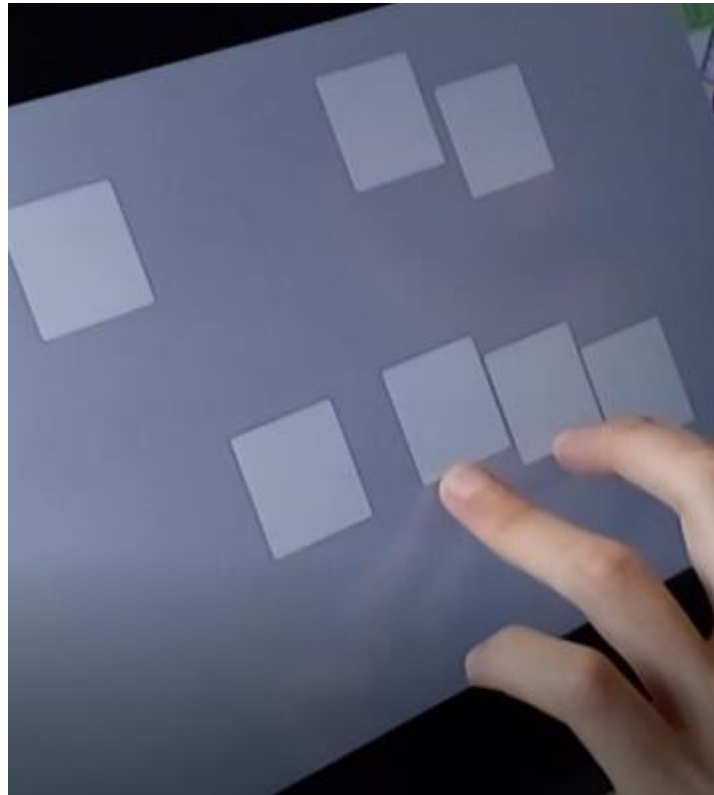


Figure 17. Dragging various squared widgets with fingers

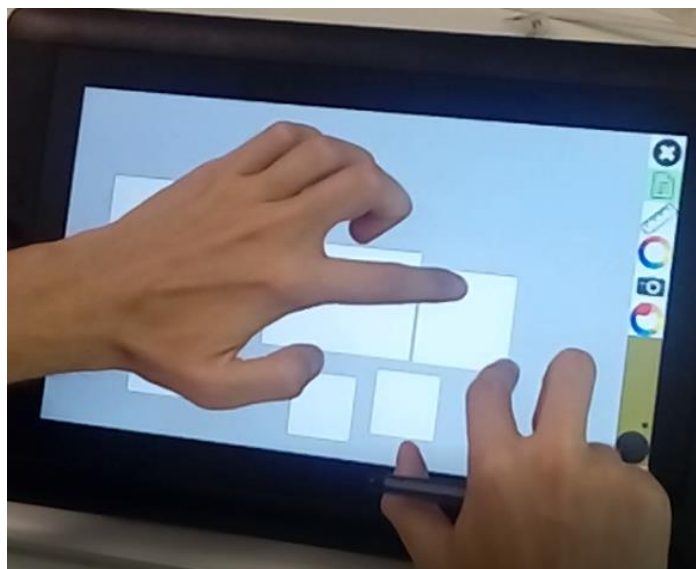


Figure 18. Zooming a widget in using fingers

Widgets can be dragged around and resized with pinch, those gestures are the same as in modern mobile phones. That operation can be also done with various widgets simultaneously.

As mentioned before, Wacom Cintiq 13HD allows to detect not only pen position, but also its pressure and rotation/tilt angle. Here are some interactions where are combined just those two interaction styles:

3.3.1 Selecting color on HSB ring

There are many models to produce colors on computer monitors. The most known one is RGB, which mixed shades of red, green and blue in values from 0 to 255 for each of those shades and created the needed color. But there is one more understandable model for humans, which is based on RGB – some mathematical operations are done for red, green and blue shades, producing the values of hue (i.e., what color is it at all), saturation and brightness, those characteristics also from 0 to 255. This model is abbreviated as HSB.

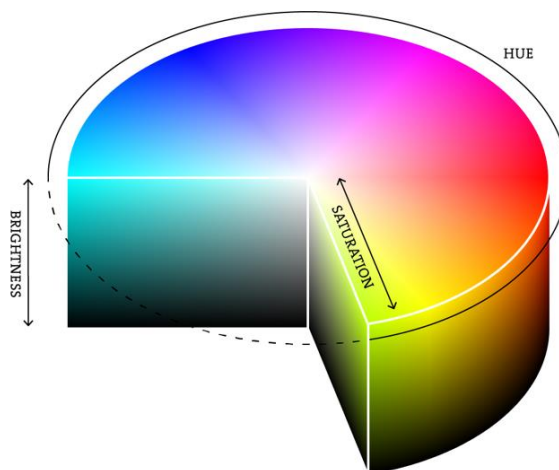


Figure 19. The HSB ring [21]

When touching a primary palette on bottom right of the screen, shown as painted circle, the characteristics calculated above are calculated the following way:

1. Hue value depends on pen's horizontal/vertical orientation (i.e., there is specific orientation for red, green, yellow, blue, etc.)
2. For more natural color sensing, pen's angle with tablet surface determines color saturation (i.e., the smaller angle is, the darker will be color)
3. And finally, with the pen's pressure the saturation is calculated (the stronger is the pressure, the stronger is the color).

This interaction is using pressure that resets after taking the pen away, so it was programmed to activate the color after positioning the pen in correct way and pressing one of two buttons.

An advantage is that infinity of colors is accessible with that style of palette, and a disadvantage is related with a comfort – depending on tablet position, it is not always easy to reach the palette when it is on bottom right of the screen. That disadvantage can be solved by positioning the palette on a more accessible position or creating a new version of palette.

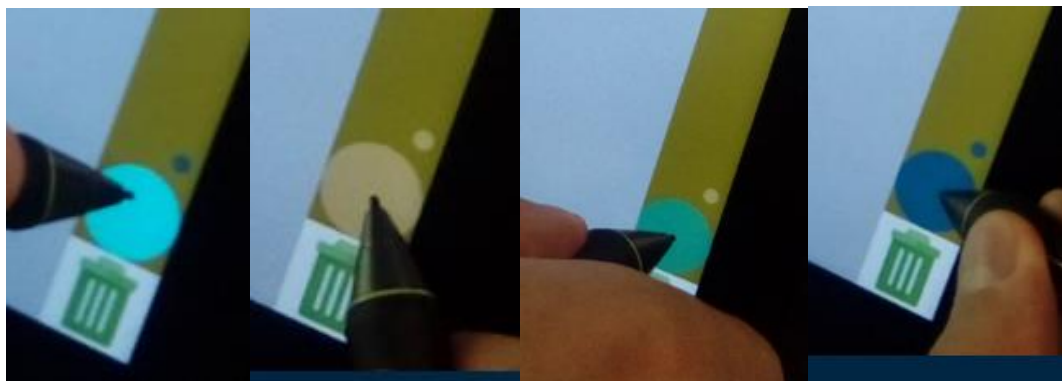


Figure 20. Touching color ring with different orientation angles produces different colors.

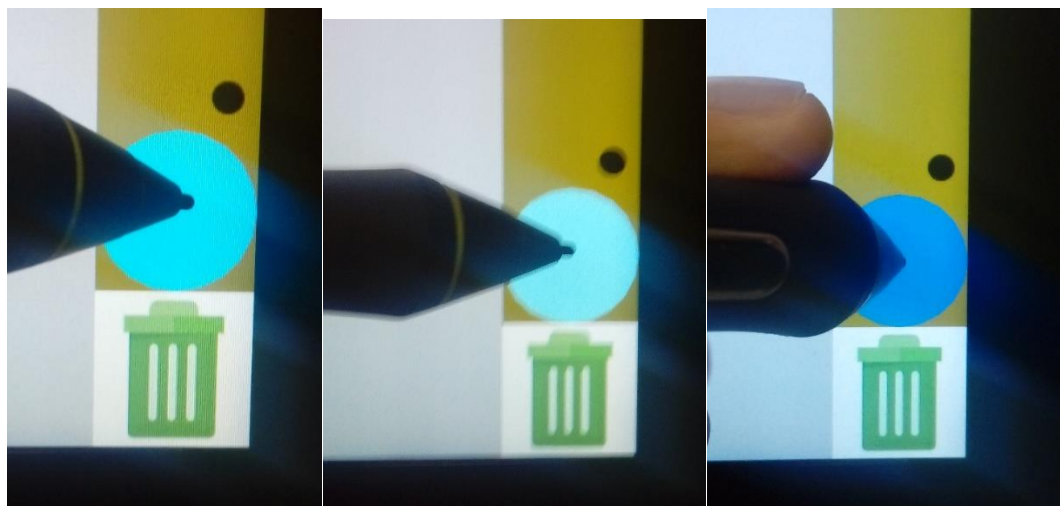


Figure 21. Different shades of blue produced by different tilt and pressure of pen.

As a solution for making comfortability for choosing some colors that require tilting the pen below the screen or on some other hard-to-reach zone, the users can touch the desired zone

for more space when using that tool. For example, as seen on a figure below, the color ring was centered on the screen, producing more space for users to tilt the pen into any orientations, allow activating any color easily.

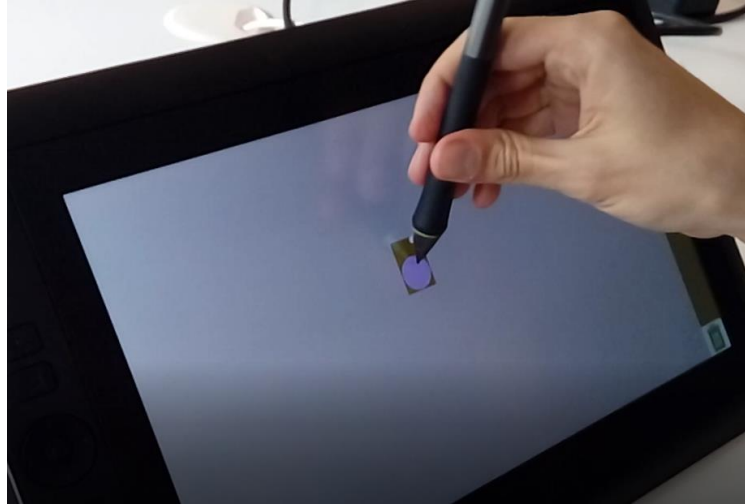


Figure 22. Tilt color palette positioned on a comfortable position.

3.3.2 Rotating widgets with pen

When the pen is set to secondary functionality, it is possible to rotate a widget around pen position by rotating a pen while tapping that widget in that mode. Since pen tilting is used for two different operations (see section 3.3.4, a pressure threshold was defined to differentiate them). The best threshold for those two operations was 0.8, i.e., widgets are rotated when pen is touching the screen with pressure less than 80%.

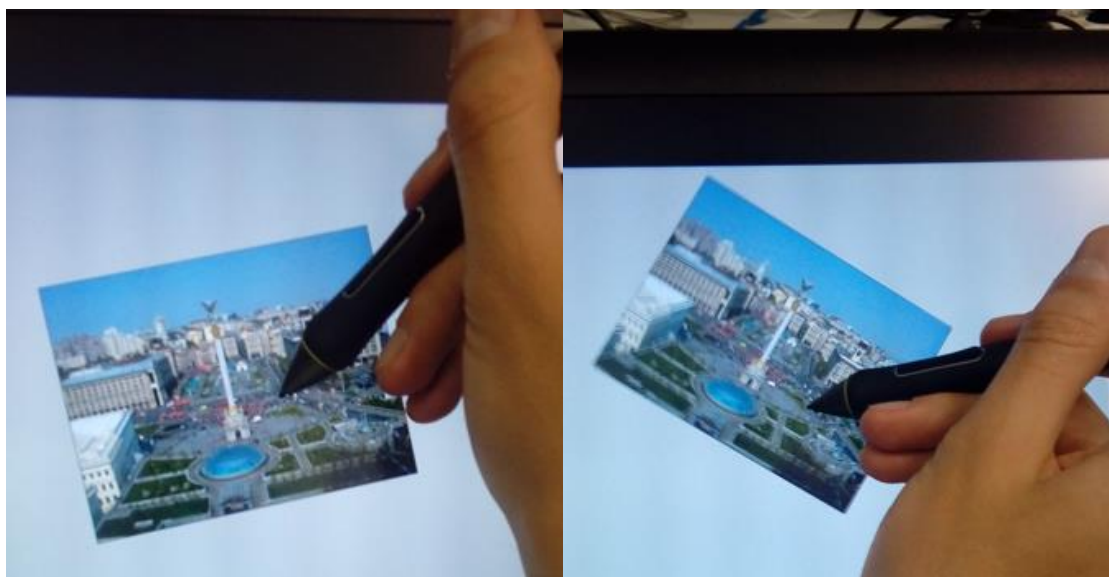


Figure 23. Rotating a widget by spinning the pen around its puck.

3.3.3 Switching images on widgets using pen

In the same setting of pen, and if more than one image were loaded to one widget, it is possible to look between all the images on that mini-album by touching it with pressure above the threshold of tilting, since we do not want tilting to be done, and then tilting a medium amount of degrees (the best value was 20 degrees). Tilting left switches to previous image, and tilting right switches to next image. This function requires enough pressure force to be performed, it was made to separate different techniques, which could cause some difficulties for beginners, which can be solved in one of the pen and touch exploits implemented.

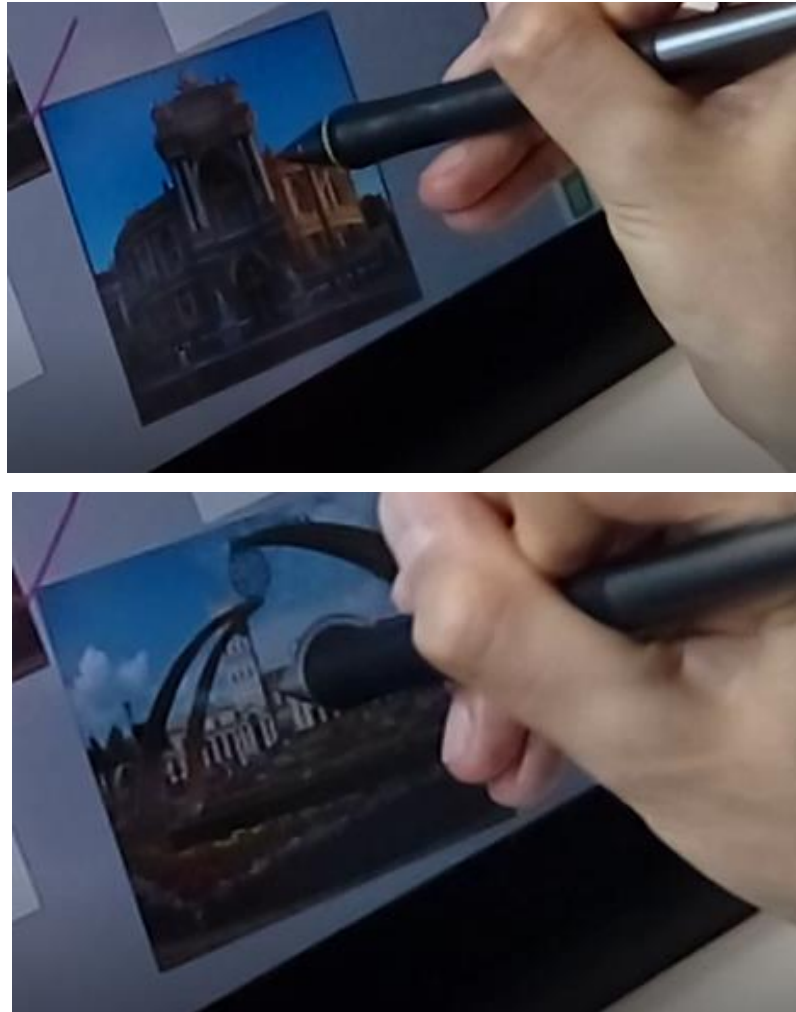


Figure 24. Switching images on a widget can be performed by tilting the pen from side to side.

3.4 Combining pen and touch

After learning how to perform single hand operations that can be performed on tablet computers using just pen or just touch, we passed to the main task of the project – find out which interactions can be done with pen and touch cartesian data combined.

This sub-section explains some solutions for exploiting new tools by combining two different interaction techniques.

3.4.1 Scratching with finger

If someone were to draw something using fresh paint, touching some points by accident or voluntarily could cause increasing of brush spots. Sometimes artists do that to scratch something with a pure or mixed portion of color, for additional effects. The same thing has been done in tablet drawing, too: whenever we paint with that tool on, touching a drawn point with 5 or less pixels of radius, it is possible to produce similar effect on the screen, i.e. rubbing some spots by boosting their size 4 times.

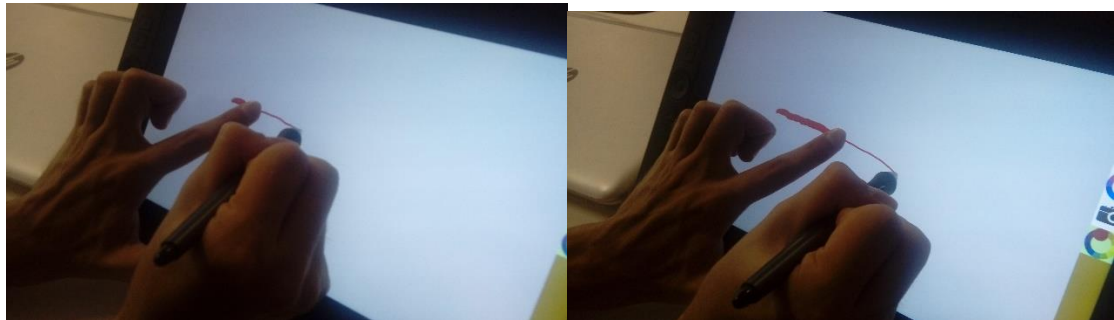


Figure 25. Scratching fresh paint using fingers.

3.4.2 Resizing using pen and finger

This interaction is improvement for two-finger. While holding a widget with pen, it is possible to pinch that widget using a finger of the hand that is not holding the pen. The benefit of this improvement is that it gives more space for the hand that is holding pen, and the pen can be held as highly as users want.

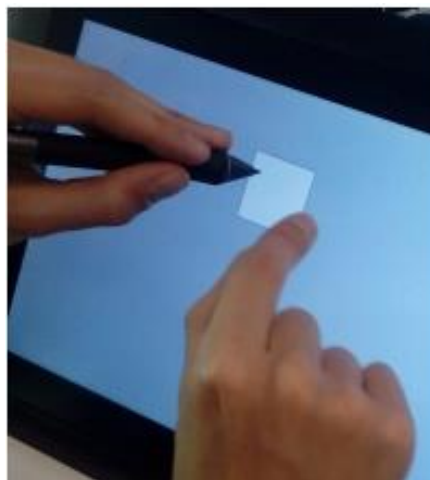


Figure 26. Resizing a widget using pen and finger.

3.4.3 Painting various widgets

As shown before, pressure and tilt usage allow to choose between infinity of colors. But we were have found a technique that can be yielded if combined with finger input. With fingers of non-preferred hand, it is possible to drag some widgets and bring them to desired position. While still holding them with touch, using pen on preferred hand, by performing color choosing with three characteristics, clicking one of two buttons (not the one that has been programmed for activating that color for drawing) widgets user is touching can be painted into the same color that was selected by pen.

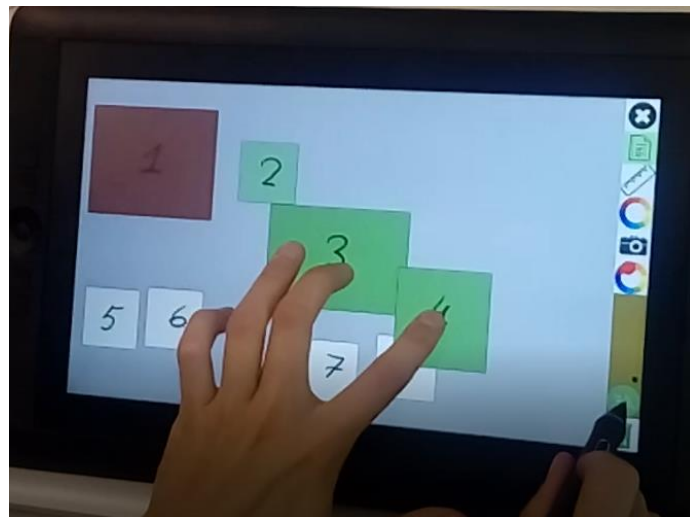


Figure 27. Painting two widgets chosen by fingers with a color chosen by pen tilt and pressure.

3.4.4 Tool-assisted drawing with ruler

A common graphical operation is to draw straight lines. For this, people usually handle a ruler that is graduated in centimeters over drawing surface and trace the lines along this ruler. This way of drawing lines is one of the new interaction technique implemented on the prototype.

The ruler used for manipulating allows to meter distances up to 12 inches, or 30,48 cm.

Here are steps to draw a straight line using ruler tool:

1. Tap the ruler icon on toolbar.
2. To define the ruler drawing vector, first touch and hold the position of ruler's far left side.

3. Then touch a point that should define ruler's direction and it will be drawn on screen. The ruler appears on screen whilst first finger still touches it. By moving the finger around, it is possible to move the ruler as well.

4. Drawing with pen on ruler's surface will make traces appear near on a straight defined by of its length side (depending on which half on the ruler the user is drawing lines – on centimeter one, or on inch one).

Pen's pressure is still applicable while drawing using ruler.

If a user desires to make ruler still visible even while not touching the tablet, it can be done by tapping the ruler with third finger. Then, to hide the ruler, it will be needed to tap it with three fingers again. Useful when user is tired of touching tablet.

Although user can decide himself which length should be his line, either on centimeters or in inches, the size of ruler is limited, and if users wants to trace a non-completely strong line, it may appear in most cases non-uniform, because it is very hard to regulate pressure level by people, and their hands are vibrating while active. This issue should be resolved in a second version of tool-assisted drawing.

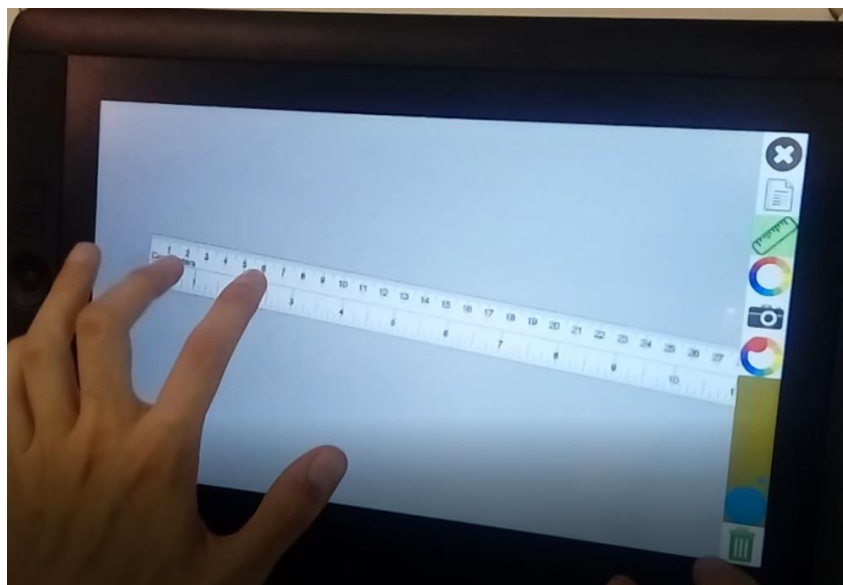


Figure 28 Two fingers indicate ruler's position and orientation.

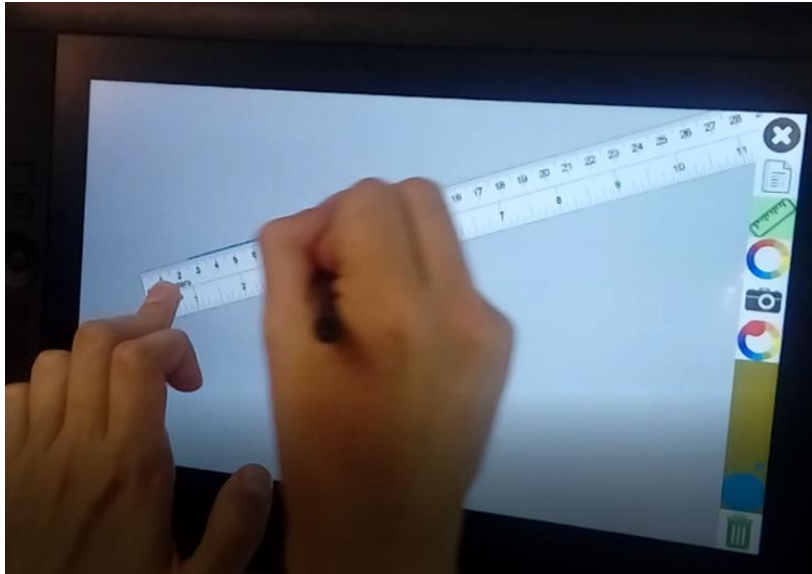


Figure 29. When holding a ruler with a finger, tracing on top of the ruler will produce the ink right near ruler's surface.

3.4.5 Bimanual straight-line tracing

The problem of making the straight lines with non-uniform point sizes can be solved in the newer version of straight-line tracing. Users can select line's first point with a finger of non-preferred hand, then use the pen on preferred hand to choose its end point. Since size of stamped line depends on its pressure level, the user should press one of the buttons on pen to draw it. The line should be now traced with points of the equal size between two points defined, and it is also possible to clear some of those points with pen's rubber.

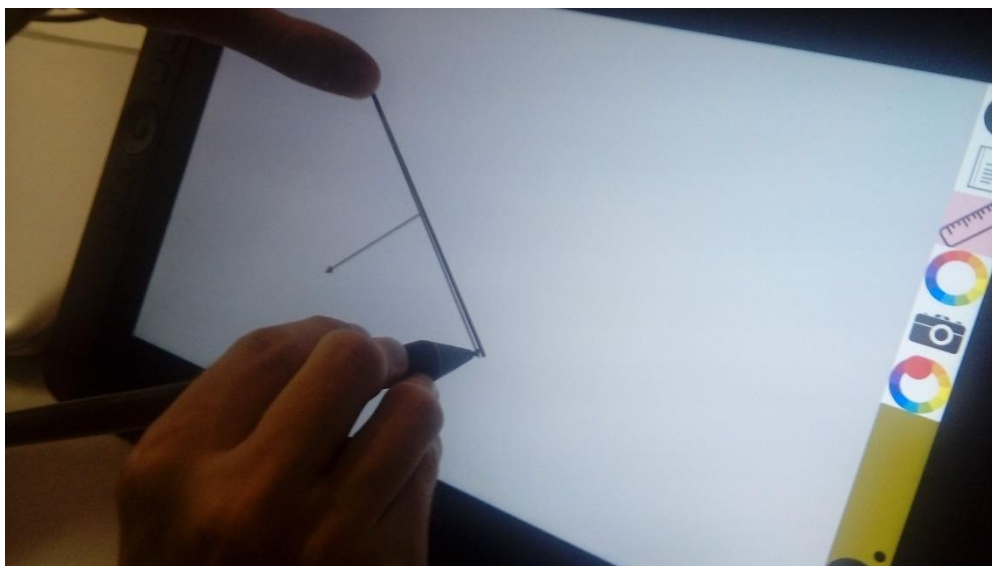


Figure 30. A straight line can be traced also using a finger and pen.

3.4.6 Bimanual semi-ellipse tracing

With that technique, it is possible to trace not only straight lines but also semi-ellipses. Tilting pen into one of the line's margins will cause another arrow grow on that margin. That margin shows the second semi-ellipse diameter. Clicking one of two buttons that is not the one for the straight line also allows stamping, but this time a semi-ellipse will be drawn. That semi-ellipse passes through three points, two defined by pen and touch, and other one, the vertex, which is defined by tilt side and angle.

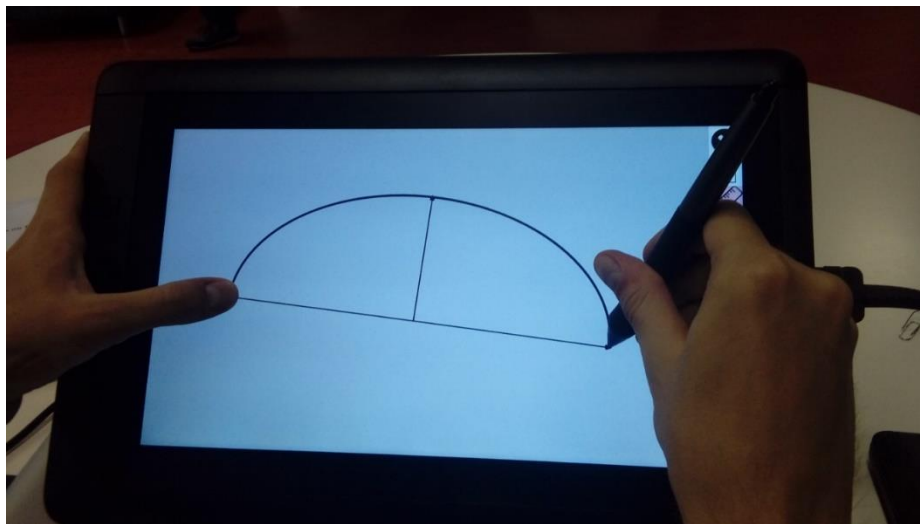


Figure 31. It is possible to trace a semi-ellipse as well by choosing two points and tilting the pen within the needed angle and direction.

3.4.7 Switching images using slider

The pen has a tiny tip which allow to hit small objects with big precision. While holding a widgets or various widgets with at least one widget on each, and that widget/those widgets is/are not rotated, a seek bar appears on top of it/them, with a small slider with 6 pixels of radius showing the relative position of current image among others, just like on software audio and video players. That slider can be moved around horizontal bar to seek between images on the widget or widgets that are held by fingers.



Figure 32. When holding a widget, users can move tiny slider using the pen to switch images.

3.4.8 Color palette

This palette to choose between nine of basic colors – red, orange, yellow, green, sky blue, blue, violet, black and white, also it allows choosing of some recent colors of first palette. To make it open, the users only needs to touch a point on working area with non-preferred hand, and then they can choose color using pen on preferred hand.

Once user takes the finger away from the screen, the palette vanishes.

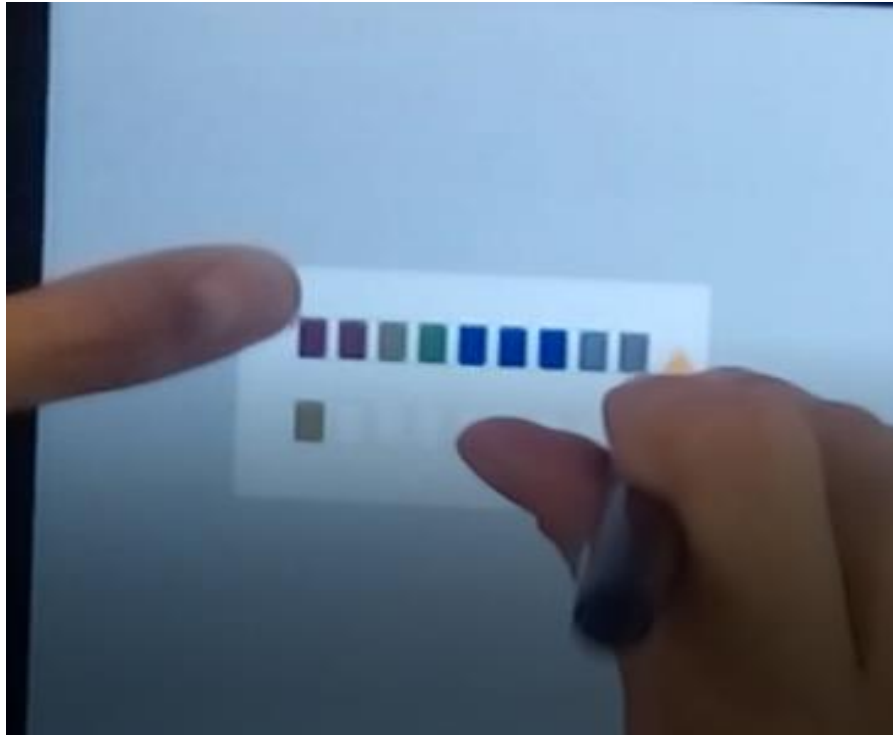


Figure 33. Palette appears on screen on the finger's position.

Specially for left-handed people, if that palette is activated on right side of screen, the colors and tilt control appear on the left of finger instead of being right for usability comfort.

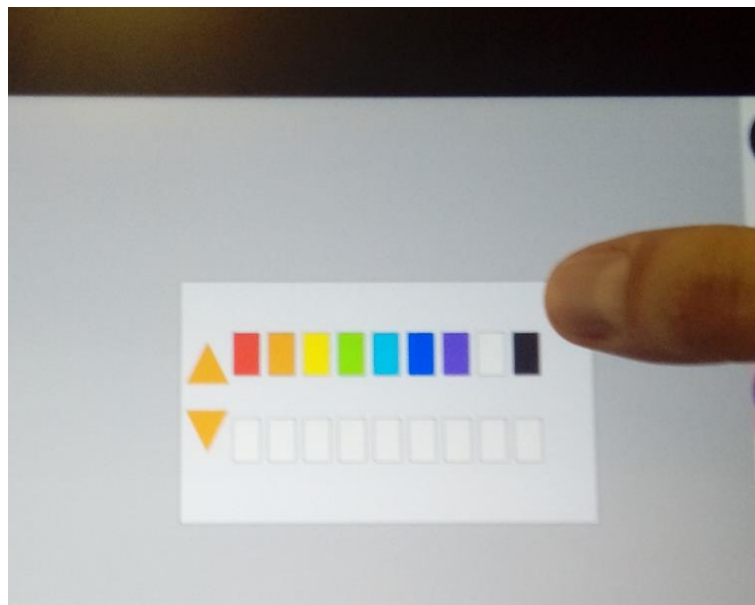


Figure 34. The palette on right side of finger, if used on right side of screen.

Users can also use vertical tilt of the pen on the place marked with arrow to increase or decrease basic colors' lightness, which allows to increase the variety of colors for drawing operations.

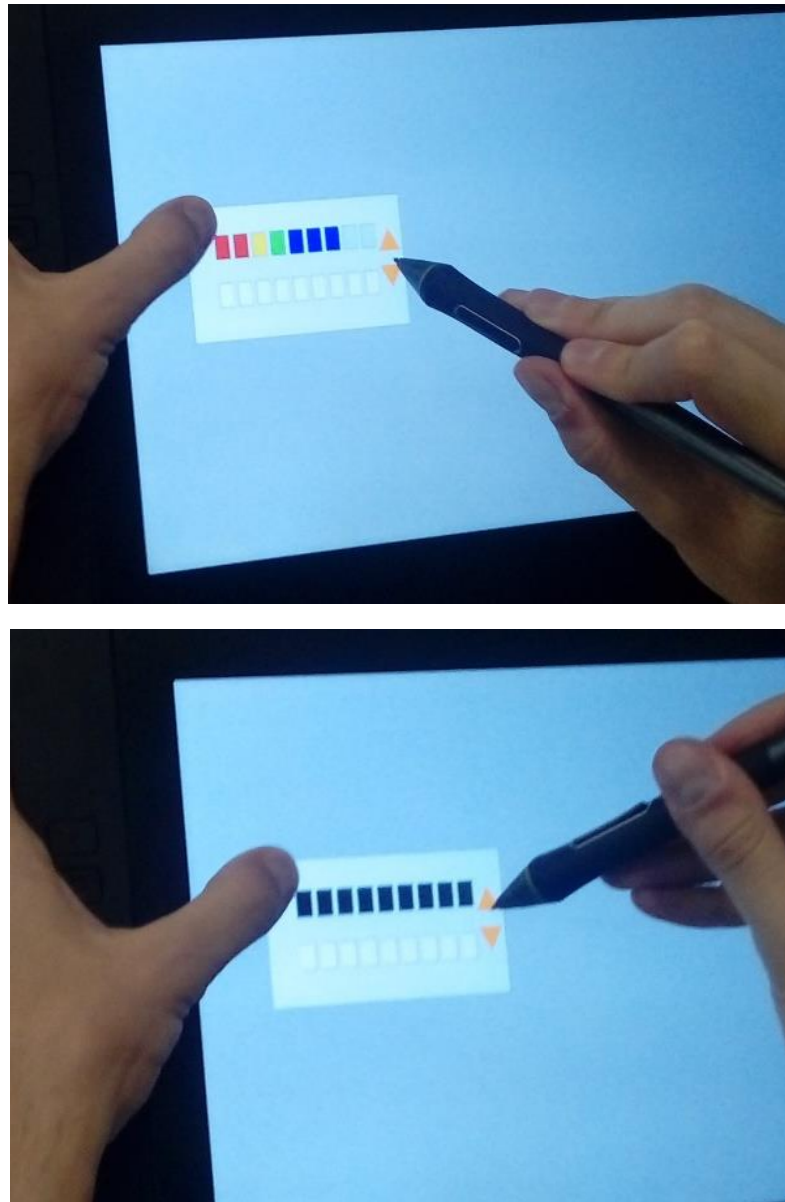


Figure 35. Different saturations produced from different pen tilt on palette.

3.4.9 Opening menu from swipe gesture

There are some paper sheets with notes that can be pushed from a side. A similar interaction can be implemented on this prototype as well.

For that of interaction, users simply need to hold a widget with a hand, then make a swipe gesture upwards, downwards, left or right. Depending on side the gesture was performed to, a menu appears near the pen. Because this artistic prototype, the menus that appear after swiping are the same palettes explained on previous sub-section. After choosing the color or swiping back to widget, the menu disappears.



Figure 36. Palette appeared after moving the pen on right side from held widget.

4. Evaluation

The evaluation included operations that could be performed and compared using pen and touch drawing inputs with standard drawing with mouse. Generally, on pen and touch drawing, a user simply touches one point with finger and another with pen and performs the needed operation. If the operation also uses pen's button, it takes some amount time to find that button and press it.

4.1 Tasks

The tasks tested allow to compare pen-touch inputs with mouse input and can be performed on any classic PC drawing software. Two interactions were implemented on the prototype. These can be done with pen and touch, but also with mouse.

A: Draw straight line between certain points using pen and touch

B: Draw straight line between certain points using mouse

C: Select certain color using pen and touch

D: Select certain color using mouse.

Here is set of sub-operations for each of those tasks:

A: Tap and hold the start position with finger-> tap and hold the end position with pen -> press the pen as strongly as needed for desired width -> press button on pen.

B: Move mouse to the start position -> click and hold -> move mouse to the end position while still holding the pen -> release button/

C: Tap and hold a position for palette with finger -> tap needed color with pen.

D: Move mouse for needed color on palette -> click and release.

4.2 Procedure

It was decided to ask for testing four selected tasks the same number of different users – four users from within-subject experiment that already are familiar with tablet devices.

Before proceeding with user tests, some statistic data about these users were collected. According to them, there are two male and two female users, their average age is 27 and the standard deviation is 1.

To make user evaluation results unbiased, we applied a Latin Square [20] counter balance method – i.e., scroll the order of tasks to perform to the left or to the right for each user. The table below shows the order generated of each task for every user:

	First	Second	Third	Fourth
User 1	A	B	C	D
User 2	D	A	B	C
User 3	C	D	A	B
User 4	B	C	D	A

Table 2. The Latin Square method applied to user evaluation.

As for mouse tasks, instead of using incorporated closed-source Windows drawing applications, another prototype for those tasks has been created. It is also based on openFrameworks, but uses mouse for interactions. It only supports straight line tracing and color selecting, similarly to Microsoft Paint, because we measure only line tracing and color choosing.

The screen positions to touch/click should be equal for each user because everyone is measuring the same task, but in different order. They have been marked on screen and explained for users before they can proceed to run those tasks.

Summarizing, pen and touch graphical operations were performed on the main project's prototype, with some timestamp code added and task key points marked, and mouse operations were made on the prototype illustrated below.

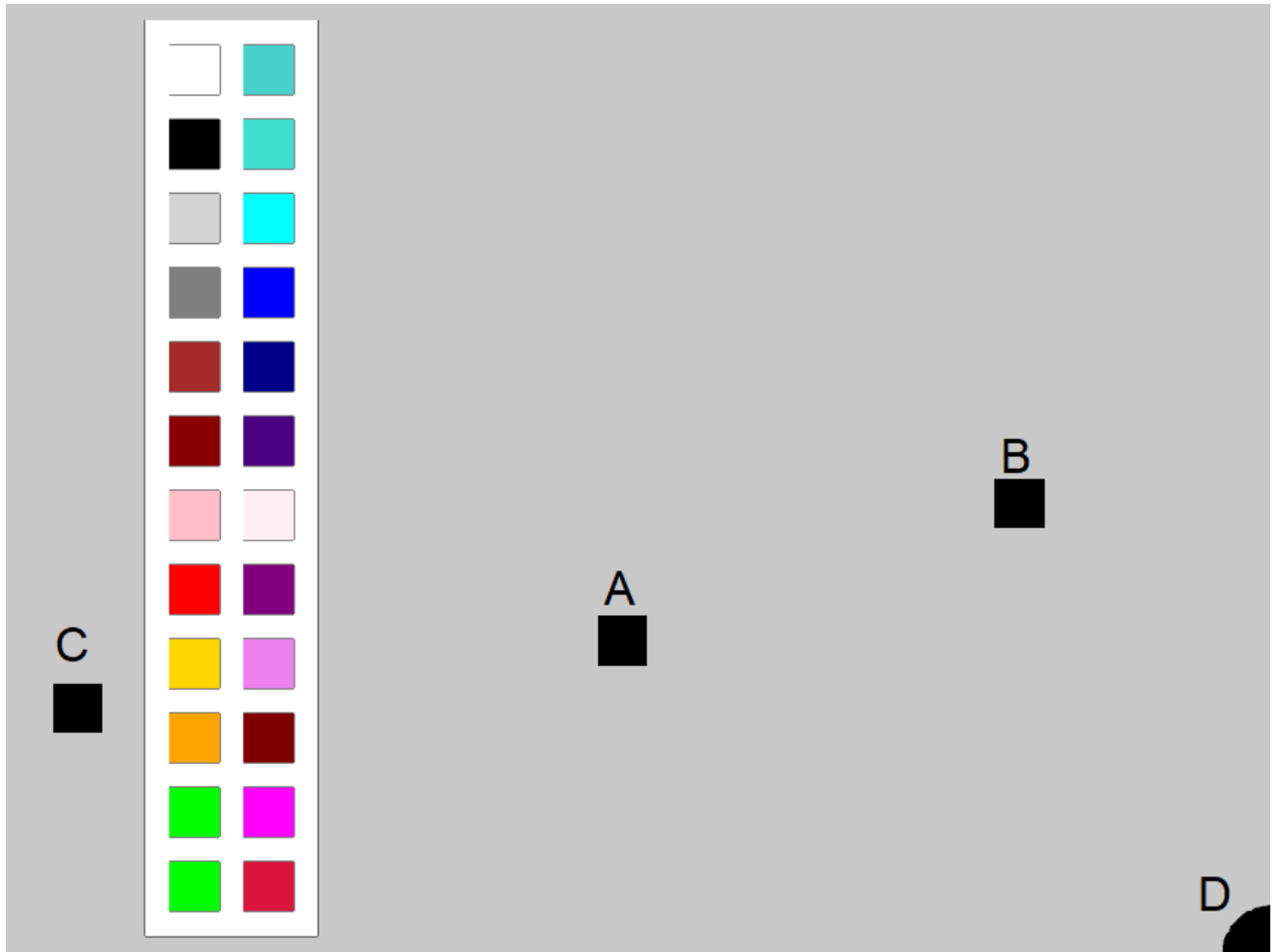


Figure 37. Mouse prototype that allows limited mouse interactions for user testing. It features a color palette that works the same way as Microsoft Paint.

On that figure were marked with letters:

A, B – two estimated locations of point between which users were asked to trace a line. The first point is on A, and the second point is on B;

C – original position of mouse from which users had to move the mouse to the needed color on palette;

D – indicator of color selected for drawing.

Because we need to calculate the time automatically when a task is performed by a user, without making strongly clear observation and using stopwatches, the process of time calculating has been enhanced by logging the time in prototype's source code in key events defined in the

tasks to run by users, that means, on each of key events of the operations, i.e. tapping on clicking needed position of the screen, it was written to a text file a set of data needed for task studying: the task performed, the cartesian position of tapped/clicked point and the current time in milliseconds to calculate tasks performing time.

4.3 Results

The users performed all the tasks successfully. Below there are the charts of everyone's total time that was needed to complete the tasks, as well as the average time for each task.

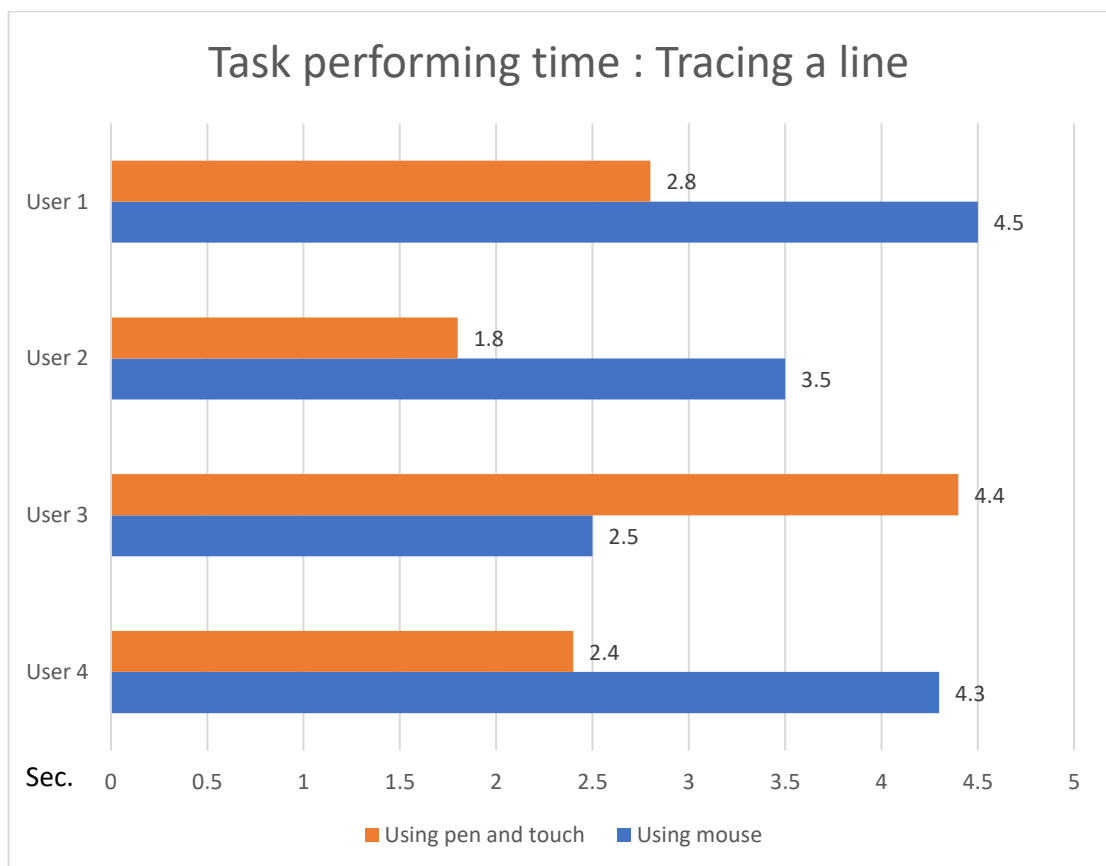


Figure 38. Tracing a line time comparison chart.

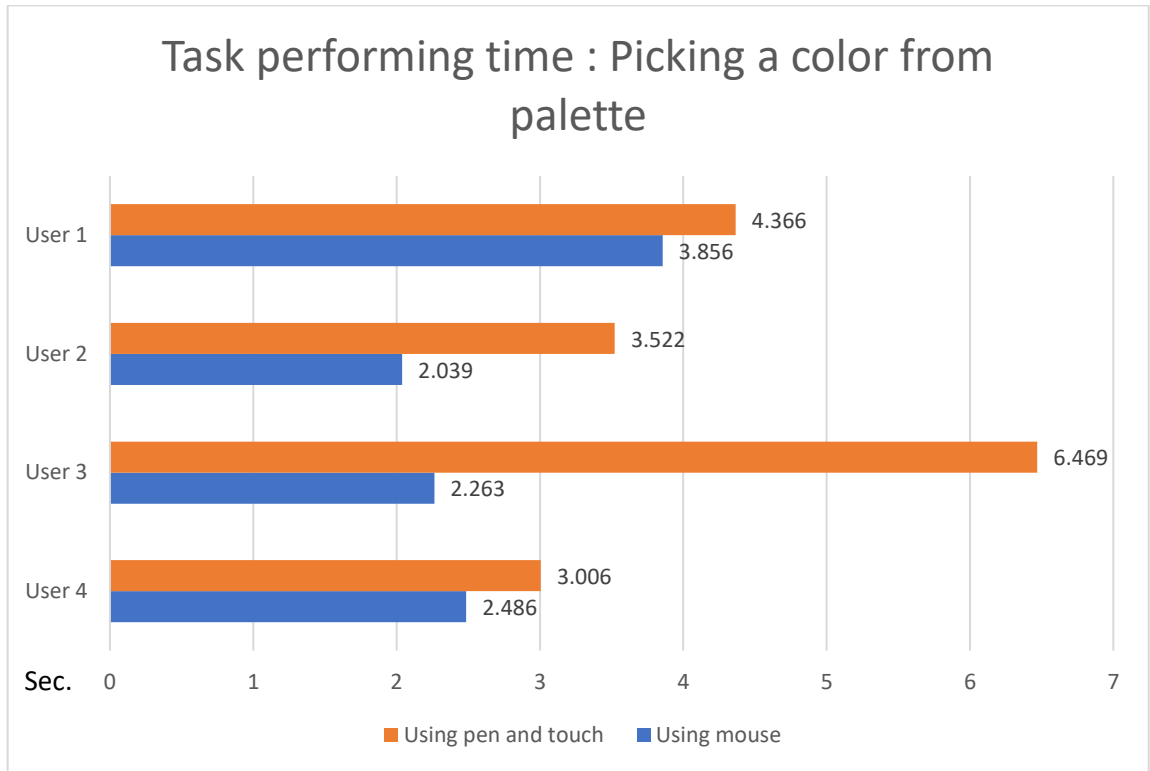


Figure 39. Picking a color from palette time comparison chart.

The charts show that line operations are faster with pen and touch color selection operations, but color selection operations are slower with this input combination.

Task	Average value	Standard deviation
Line with pen and touch	2.85	1.11
Line with mouse	3.7	0.91
Color with pen and touch	4.34	0.81
Color with mouse	2.66	1.53

Table 3. The averages and standard deviation of all values obtained.

5. Conclusions

The prototype was designed to perform bimanual graphical operations on a tablet display, supporting widgets that are designed for drawing distinguishing and transforming, and also by loading and managing some images on them. All graphical operations tools are performed with pen and multitouch, and have been implemented in the way so the users can feel naturality of their interactions.

For example, widget can be turned around with their drawings and or photos just by spinning them with pen. Drawing using ruler seems even more natural, as on paper.

Some of single hand operations, for example, choosing color from ring by tilting the pen on that ring's fixed position bring the least usage comfort because of pen sliding on screen and difficulty to reach small angles on bottom orientation when tablet is on a table.

There are unlimited techniques that could be implemented for using on tablet, and even existing techniques need to be adjusted for certain types of users or usage environments.

The results show (see figures 38, 39 and table 3) that task performing time in using pen and touch interactions was better than in using the mouse in line trace task, but choosing color was performed more quickly on mouse prototype.

The factors that affected the time in performing operations are generally the touch precision, the item size and thinking speed.

5.1 Future work

According to work that has been done already, and that there are unlimited ways to exploit combination of pen with pressure/tilt and multitouch data, some new tools are planned to implement in future.

For example, painting on 3D surface, where it should be possible to spin the 3D sprites with one finger, zoom in and out with two fingers, and make some transformations by tracing the determinate locations with pen holding some options with touch.

Since it was implemented switching images on widgets using some sliders, that kind of interactions could be applied to some video clips as well.

It is planned also to increase the number of tools for graphical operations, to be able to trace some figures evolving straight lines and circles.

It can be possible to extended widget functionality by creating some smart charts, making some significant relation between widgets.

6. References

1. *Wacom Cintiq 13 HD Graphic Pen Tablet for Drawing*. Visited September 2016 at <http://www.wacom.com/en/products/pen-displays/cintiq-13-hd>
2. *openFrameworks*. Visited September 2016 at <http://openframeworks.cc/>
3. Baxter, W. (2010) *bbTablet Download Page*. Visited September 2016 at <http://www.billbaxter.com/projects/bbtablet/>
4. *Wacom Feel Multi-Touch API*. Visited October 2016 at <http://www.wacomeng.com/touch/WacomFeelMulti-TouchAPI.htm>
5. Florence, I. (April 2013) *From touch displays to the Surface: A brief history of touchscreen technology* from *ArsTechnica* at <https://arstechnica.com/gadgets/2013/04/from-touch-displays-to-the-surface-a-brief-history-of-touchscreen-technology/>
6. Gilbertson, B. (July 2013) *How does the Wacom Tablet PC Pen Work?* (July 2013) from *TabletPC* at <http://blog.tabletpc.com.au/2013/07/19/how-does-the-wacom-tablet-pc-pen-work/>
7. *Wacom Graphics Tablet History: Touch Computing's Early Icon* (September 2017) Tedium at <https://tedium.co/2017/09/21/wacom-tablet-history/>
8. Hoffman, C. (December 2013) *Not All Tablet Styluses Are Equal: Capacitive, Wacom, and Bluetooth Explained* from *How-To Geek* at <https://www.howtogeek.com/177376/not-all-tablet-styluses-are-equal-capacitive-wacom-and-bluetooth-explained/>
9. Buxton, B. (January 2007) *Multi-Touch Systems that I Have Known and Loved*. Retrieved January 2016 at <http://www.billbuxton.com/multitouchOverview.html>
10. Ken Hinckley, Koji Yatani, Michel Pahud, Nicole Coddington, Jenny Rodenhouse, Andy Wilson, Hrvoje Benko, and Bill Buxton. 2010. *Pen + touch = new tools*. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology (UIST '10)*. ACM, New York, NY, USA, 27-36. DOI: <https://doi.org/10.1145/1866029.1866036>

11. Peter Brandl, Jakob Leitner, Thomas Seifried, Michael Haller, Bernard Doray, and Paul To. 2009. *Occlusion-aware menu design for digital tabletops*. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems (CHI EA '09)*. ACM, New York, NY, USA, 3223-3228. DOI: <https://doi.org/10.1145/1520340.1520461>

12. Peter Brandl, Clifton Forlines, Daniel Wigdor, Michael Haller, and Chia Shen. 2008. *Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces*. In *Proceedings of the working conference on Advanced visual interfaces (AVI '08)*. ACM, New York, NY, USA, 154-161. DOI: <https://doi.org/10.1145/1385569.1385595>

13. Haijun Xia, Ken Hinckley, Michel Pahud, Xiao Tu, and Bill Buxton. 2017. *WritLarge: Ink Unleashed by Unified Scope, Action, & Zoom*. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 3227-3240. DOI: <https://doi.org/10.1145/3025453.3025664>

14. Ken Pfeuffer, Ken Hinckley, Michel Pahud, and Bill Buxton. 2017. *Thumb + Pen Interaction on Tablets*. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 3254-3266. DOI: <https://doi.org/10.1145/3025453.3025567>

15. Gonzalo Ramos, Matthew Boulos, and Ravin Balakrishnan. 2004. *Pressure widgets*. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 487-494. DOI=<http://dx.doi.org/10.1145/985692.985754>

16. *MultiTaction – Hardware*. Visited September 2017 at <https://www.multitaction.com/hardware>

17. *Anoto – Digital Writing Solution. How does it work?* Visited September 2017 at <http://www.anoto.com/enterprise/how-does-it-work/>

18. Buxton, B. *Some Milestones in Computer Input Devices: An Informal Timeline*. Visited September 2017 at <https://www.billbuxton.com/inputTimeline.html>.

19. Buxton, B. *Buxton Collection by Microsoft*. Visited September 2017 at <https://www.microsoft.com/buxtoncollection/browse.aspx>

20. *A Field Guide to Experimental Designs*. Visited September 2017 at <http://www.tfrec.wsu.edu/ANOVA/Latin.html>

21. Gore, T. (February 2016) *Color Theory and Landscape Photography* at <http://www.tedgorecreative.com/blog/2016/2/8/imc0w2lc0ie6b1jtixfht4jgqyjn>

22. Diogo Cabral, João G. Valente, Urândia Aragão, Carla Fernandes, and Nuno Correia. 2012. *Evaluation of a multimodal video annotator for contemporary dance*. In Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12), ACM, New York, NY, USA, 572-579. DOI=<http://dx.doi.org/10.1145/2254556.2254663>

23. Khalil Klouche, Tuukka Ruotsalo, Diogo Cabral, Salvatore Andolina, Andrea Bellucci, and Giulio Jacucci. 2015. *Designing for Exploratory Search on Touch Devices*. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15). ACM, New York, NY, USA, 4189-4198. DOI: <https://doi.org/10.1145/2702123.2702489>

24. Diogo Cabral and Nuno Correia. 2017. *Video editing with pen-based technology*. *Multimedia Tools Appl.* 76, 5 (March 2017), 6889-6914. DOI: <https://doi.org/10.1007/s11042-016-3329-y>