University of Central Florida
STARS

Electronic Theses and Dissertations, 2004-2019

2014

SketChart: A Pen-Based Tool for Chart Generation and Interaction.

Andres Vargas Gonzalez
University of Central Florida

Part of the Computer Sciences Commons, and the Engineering Commons
Find similar works at: https://stars.library.ucf.edu/etd
University of Central Florida Libraries http://library.ucf.edu

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation

Gonzalez, Andres Vargas, "SketChart: A Pen-Based Tool for Chart Generation and Interaction." (2014). *Electronic Theses and Dissertations, 2004-2019*. 4752.
https://stars.library.ucf.edu/etd/4752

Showcase of Text, Archives, Research & Scholarship

SKETCHART: A PEN-BASED TOOL FOR CHART GENERATION AND INTERACTION

by

ANDRÉS VARGAS GONZÁLEZ
B.S., Computer Science, Escuela Superior Politécnica del Litoral, 2010
Guayaquil - Ecuador

A thesis submitted in partial fulfillment of the requirements
for the Degree of Master of Science
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at University of Central Florida
Orlando, Florida

Summer Term
2014

Major Professor: Joseph J. LaViola Jr.

© Copyright 2014 by Andrés Vargas González

ABSTRACT

It has been shown that representing data with the right visualization increases the understanding of qualitative and quantitative information encoded in documents. However, current tools for generating such visualizations involve the use of traditional WIMP techniques, which perhaps makes free interaction and direct manipulation of the content harder. In this thesis, we present a pen-based prototype for data visualization using 10 different types of bar based charts. The prototype lets users sketch a chart and interact with the information once the drawing is identified. The prototype's user interface consists of an area to sketch and touch based elements that will be displayed depending on the context and nature of the outline. Brainstorming and live presentations can benefit from the prototype due to the ability to visualize and manipulate data in real time. We also perform a short, informal user study to measure effectiveness of the tool while recognizing sketches and users acceptance while interacting with the system. Results show SketChart strengths and weaknesses and areas for improvement.

ACKNOWLEDGMENTS

First of all thanks God, then the example of my grandmother Lola and her efforts is my greatest motivation in every step I successfully complete in my life. Special thanks to my mom Beatriz, aunts (Paulina & Alejandrina) and uncles whom in several ways have shaped my character and determination.

Many thanks to my faculty advisor, Dr. Joseph LaViola, for his guidance and directions along my graduate studies and for his financial support the last term. Additionally, I greatly appreciate the members of the Interactive Systems and User Experience Research Cluster of Excellence for their help. Furthermore, I would like to thank the members of my committee, Dr. Kien Hua, and Dr. Hassan Foroosh for their contributions.

Finally, but not least I am sincerely grateful to the Fulbright program and University of Central Florida for their great support during my stay in USA. Being a Fulbright scholar has been one of the best experiences of my life.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 – INTRODUCTION

Statistical graphs are a type of data visualization that have gained interest with the rise of big data and better computational capabilities [20, 21, 22]. Graph comprehension and perception [1, 2] have been subjects of study for education and analysis of scientific data. These representations are often used to express quantitative (e.g., frequency of rolling a dice or scores in a GRE test) and qualitative (e.g., days of the week or list of operating systems) information. In a sense, we can say that a good graph is the one that expresses a clear idea of the patterns and relations of the variables involved. Additionally these studies analyze different factors that improve the viewer's interpretation of the data [1, 2].

Most of the traditional ways to invoke graphs and interaction make use of graphical user interfaces (GUI) such as buttons, menus and icons. Microsoft Excel, GNU Octave, MATLAB, R, [23] and so on provide a GUI to interact with the graphs. With the advent of tangible surfaces, it is important to explore new means of interaction for graphing and data visualization.

We propose a framework for the generation and interaction of statistical charts through the use of a pen-based prototype tool. SketChart uses a recognition engine to support 10 different bar based graphs. The sketch is read, analyzed and processed to obtain as output the closest graphical representation to the drawing. Users can modify values in the different axes, change max and min values, add and remove series of data, etc. At the same time they can see these changes in the visualization.

1.1 – Motivation

The main motivation of this thesis is to explore how to best leverage pen-based computing in interactive chart generation and editing. The use of multi-touch and pen-based devices have started to change the traditional interaction paradigm. Natural user interfaces (NUI) are a new trend based in this approach. We consider an exploratory analysis of graph generation and interaction for this paradigm. Current work have focused more on the interaction and basic generation through gestures [3, 5]. Our approach goes beyond this and can be expanded to support more graphs and the load of an external dataset.

Additionally, SketChart can be combined with a brainstorming tool to have a better visualization of the data; it can also be used in live presentations to do real time modifications and changes on the go. A user can easily use SketChart to generate the graph and export it to the most convenient type of file. We expect that SketChart should let users perform operations on the prototype that otherwise, in a traditional system, would require more time and more cognitive load.

1.2 – Research Question

By developing SketChart, we wanted to explore the effectiveness of pen-based computing for creation and interaction of charts. How easy is this paradigm for the user to interact via sketch and touch? What is the perception of the user while interacting with SketChart?

In order to find an answer to these questions we developed software that allows users to interact with the data in a more natural way, and we assigned to them some small tasks that they performed using the SketChart prototype. By observation and evaluation we believe we gathered some insight into these questions.

1.3 – Contributions

The contribution of this work is as follows:

1. Exploratory analysis about graph generation and interaction.
2. Interaction techniques through gestures and drawing.
3. Algorithm
    a. Integration of sketchREAD (Ink Parsing) with MEGIS (Segmentation).
    b. Generation of low, intermediate and high level hypotheses.

1.4 – Reader's Guide

In this section a brief explanation of each chapter is given.

Chapter 2 – Analyzes some related work especially in the area of charts and sketch interaction, ink parsing recognition is also covered.

Chapter 3 – Covers the different types of graphs that will be used in this research.

Chapter 4 – Presents the elements of the user interface, limitations and strengths.

Chapter 5 – Presents the architecture of the system and how the different components interact with each other.

Chapter 6 – Discusses the results from the informal user study.

Chapter 7 – Presents conclusions and possible future work.

CHAPTER 2 – RELATED WORK

2.1 – Charts and Sketch Based Interaction

Microsoft Research has done some exploration in the field of automatic generation of graphs supplied with data while interacting with the elements on the charts with an intuitive user interface, SketchVis [3] and SketchStory [5]. In between SketchVis and SketchStory a Wizard of Oz study was performed to analyze how people can combine touch and sketch interaction while exploring data on a chart [6]. This study concludes that sketch interactions per se were not natural to the user at the beginning but they can be natural once learned. Touch-based interactions especially the ones close to real world actions such as pinch, zoom, scale and drag, were successfully performed by users. The rest of the study was focused on which actions were best performed by pen and which ones are best performed by touch. In SketchVis, the interaction is basically performed by gestures to redraw the same data in a different graph, by changing some values, or applying some functions onto the graphs (i.e. MAX and MIN functions). In Baur, Lee, and Carpendale's paper, they explore interaction over massive data through multi-touch events, this scenario is similar but more narrow to just interacting with a stacked chart [4]. A set of gestures is defined and the graph is generated automatically from a set of data. From this set of gestures they can drag and drop stacks from the graph, zoom in and zoom out of a selected area, change between graphic representation and orientation, etc.

Usability and interaction has been the main purpose of this work. However, this thesis proposes to focus research on sketch recognition for such graphs. Rather than automatically generate a visual representation from data, it is expected of the user to draw a sketch; the system to recognize it and fit the data according to the user drawing.

2.2 –Sketch understanding

Sketch is the most natural way used for humans to express emotions, feelings, knowledge, etc [24]. Human beings can understand sketch using their perceptual system and decoding the information with myriads of pattern recognizers interconnected in the neo-cortex, which fires when some special feature or set of features is identified [25]. Following this approach, a number of solutions have been proposed in order to make computers understand sketch through recognition algorithms [7, 14, 26 and 27]. In [26] an attempt to understand how humans sketch objects is presented and a dataset of sketches is gathered and released.

Sketch recognition can be categorized as multi-domain or specific domain. SketchRead [7] is a hierarchical multi-domain recognition system engine that works under different domains. It works by levels of recognition at the lower level; primitive shapes are recognized such as lines, squares, circles, etc. Higher level components are produced by the union of primitives with other primitives or with other high level components and some constraints applied to these elements. They evaluated the proposed method on family trees and circuit diagrams. The results show that in the family tree domain, SketchREAD recognition performance was better than in circuit's domain. They state that due to the high overlap degree between circuits' elements and the greater number of them it was difficult to disambiguate. Another issue was the processing time, which was longer in circuits as well, yielding to a non-real-time run.

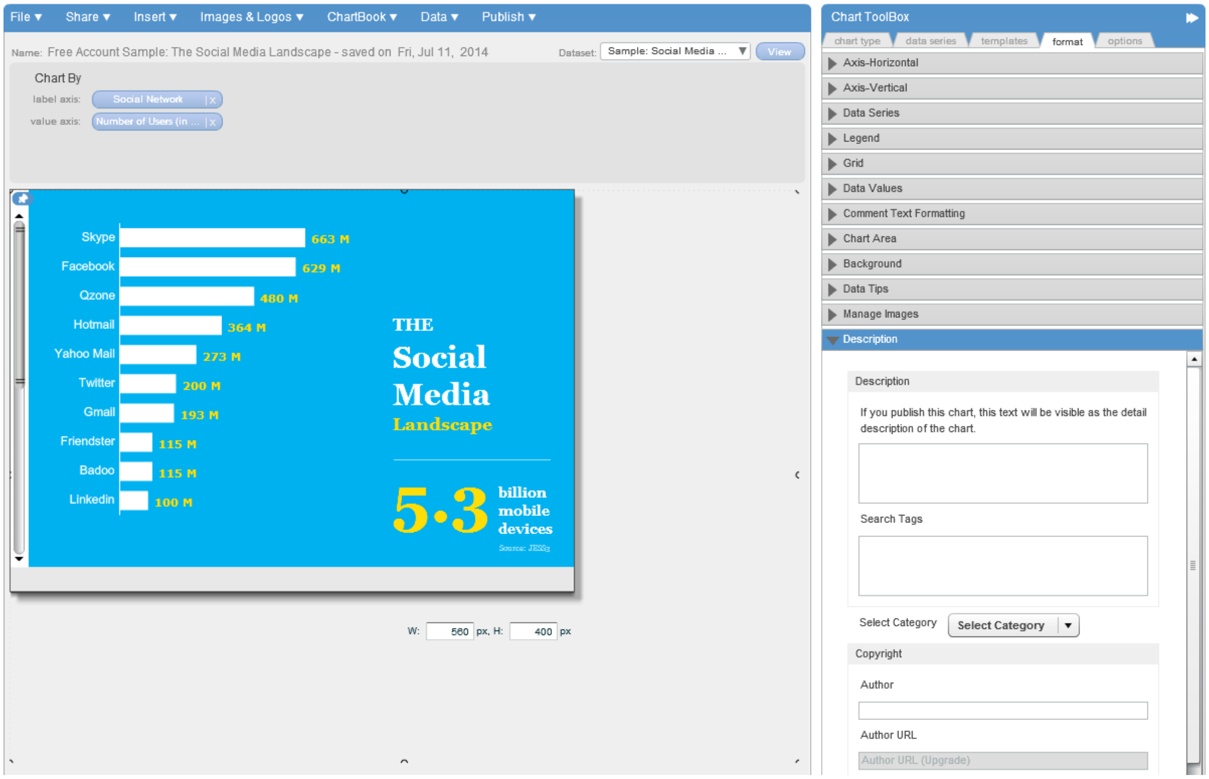
Specific domain recognition engines have been developed in several domains. For mathematics, a system for solving mathematics problems from equations written by the user is proposed in mathpad^2 [8]. In chemistry, a real time recognition engine for chemical drawings is implemented in chemink [9]. In physics, a system for solving physics problems using a sketch-based interface is called PhysicsBook [10]. In artificial intelligence, a prototype, OntoSketch [11], to generate ontologies from the user sketch is presented; it creates and extends formally represented knowledge. In the programming domain, [12] proposes a modification to a sketching tool InkKit [13] to make it domain specific and allow the recognition of entity relation diagrams.

These are just a few examples from a variety of solutions proposed to analyze sketches generated by users. SketChart uses SketchRead as a recognition engine along with a segmentation algorithm [17] to reduce the recognition search space.

2.3 –Traditional graphical user interface interaction

In traditional interfaces like Microsoft Excel or any online tool such as in [27,28,29] or a standalone application like Tableau [30], the generation and interaction of charts is performed by the use of keyboard and mouse (See Figure 1). For the user the amount of elements or controls on the interface could result in an unnecessary cognitive load. The type of interactions on these systems is limited by drag and drop elements on the interface (e.g., Tableau), combinations of buttons clicked, and textbox filled. These are definitely valid means of interaction, but, due to the advance of interaction technologies it is necessary to go beyond mouse and keyboard. In [31] an analysis is performed extending design considerations based on the characteristics of information visualizations and current interaction techniques.

Another aspect of visualizations is that they are often not accessible for blind people. In [32], a tool for improving accessibility in charts and graphs is proposed. The tool uses keyboard and speech handling commands as means of interaction. Finally, in [33], a taxonomy of interactive dynamics (operations on the graph) for data visualization is defined.



Name: Free Account Sample: The Social Media Landscape - saved on Fri, Jul 11, 2014

Dataset: Sample: Social Media ...

Chart By

label axis: Social Network

value axis: Number of Users (in ...

W: 560 px, H: 400 px

Description

If you publish this chart, this text will be visible as the detail description of the chart.

Search Tags

Select Category: Select Category

Author

Author URL: Author URL (Upgrade)

Figure 1 – Graphical User Interface of ichart. An online tool to generate and interact with data.

CHAPTER 3 – STATISTICAL GRAPHS

A set of graphs [15] have been selected for evaluating our recognition engine and the user interface. Bar based graphs were chosen due to their popularity and the different types of them. Additionally, the features between each graph are similar which makes them suitable for exploring an interesting sketch recognition problem.

3.1 – Bar based Graphs

Bar based graphs are types of charts that represent quantitative information through a graphical visualization. They are composed by series of rectangles (bars), arranged differently depending on the nature of the bar graph and the information. The height or width of these rectangles will depend on the numbers they represent and the graph scale.

3.1.1 – Types of Bar Graphs

Bar graph: A sequence of rectangles (series), which can be arranged horizontal or vertical (See Figure 2). The end of each rectangle represents a numerical value. One axis in this graph represents a category and it is most of the time a string value such as, months, days, products, etc. The other axis represents a number. Each bar is separated from each other by a constant value.

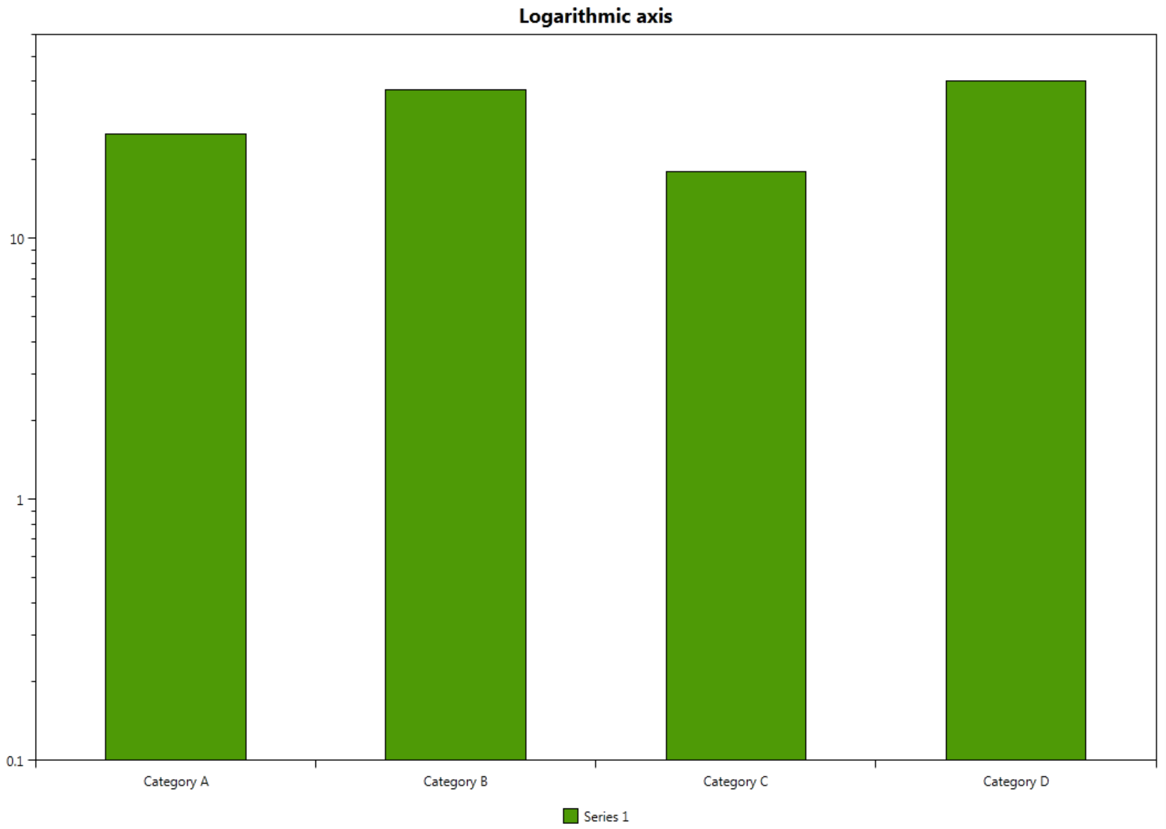


Figure 2 –Bar graph

Grouped bar graph: A sequence of multiple data series, which can be arranged horizontally or vertically (See Figure 3). There is no gap between each series that belong to the same category. Good for individual comparison of series for each category.

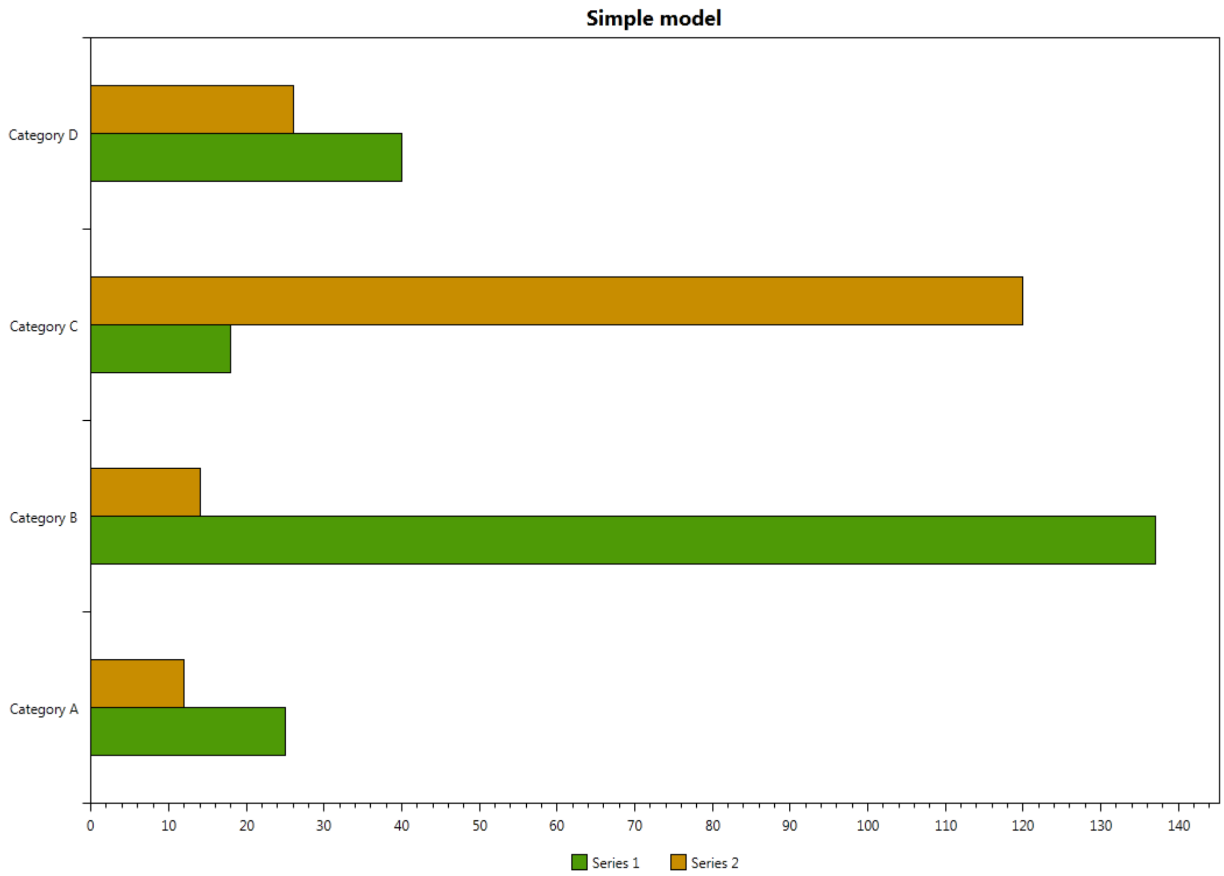


Figure 3 – Grouped bar graph

Stacked bar graph: This is a type of Grouped bar graph in which the series are on top of one another rather than next to (See Figure 4). This representation is useful to analyze how much each series is representative to the total amount. The order of the series when rendered does not follow a rigid rule.

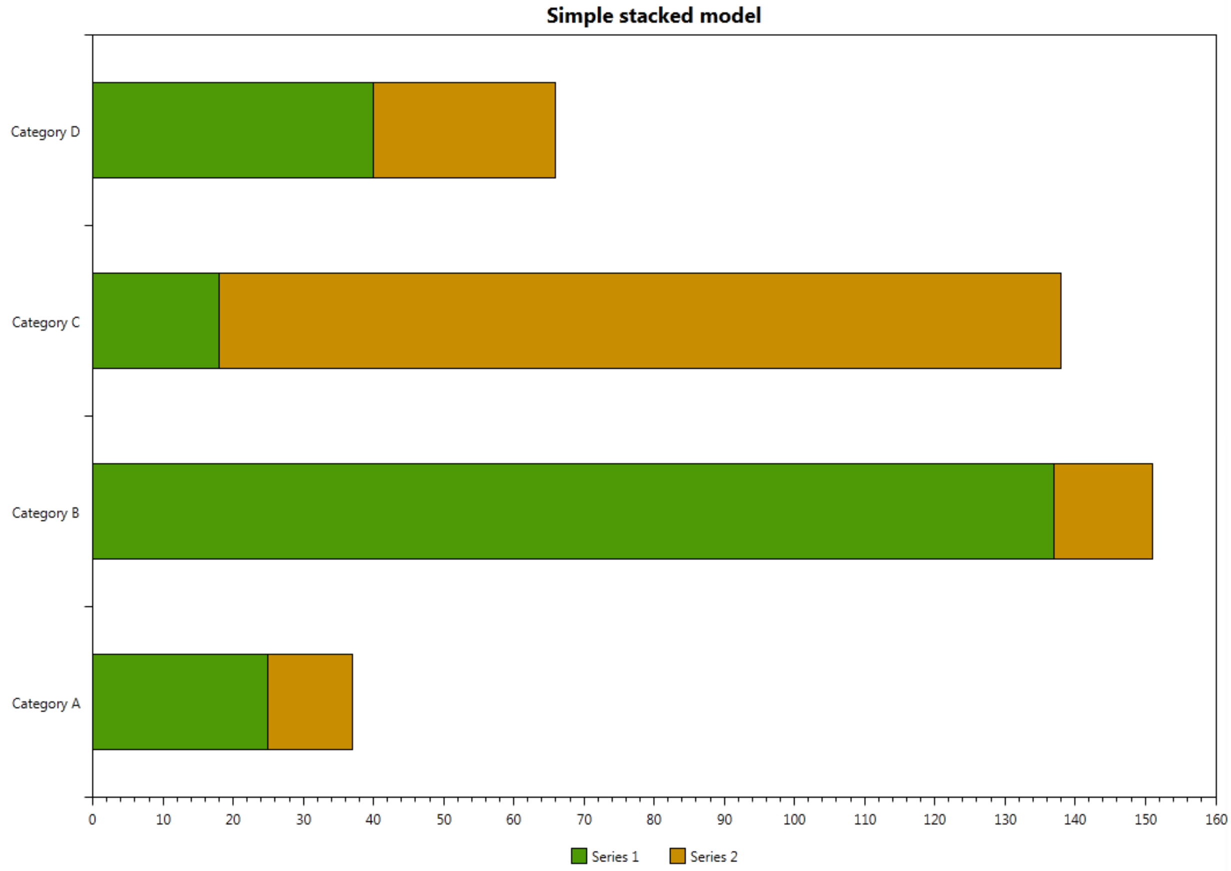


Figure 4 – Stacked bar graph

100% stacked bar graphs: This graphical representation is a stacked bar graph with a maximum value of 100% (See Figure 5). Each of the values for each series is scaled to percentages. The gap between each sequence of data series could be set to 0 to understand better the nature of the data.

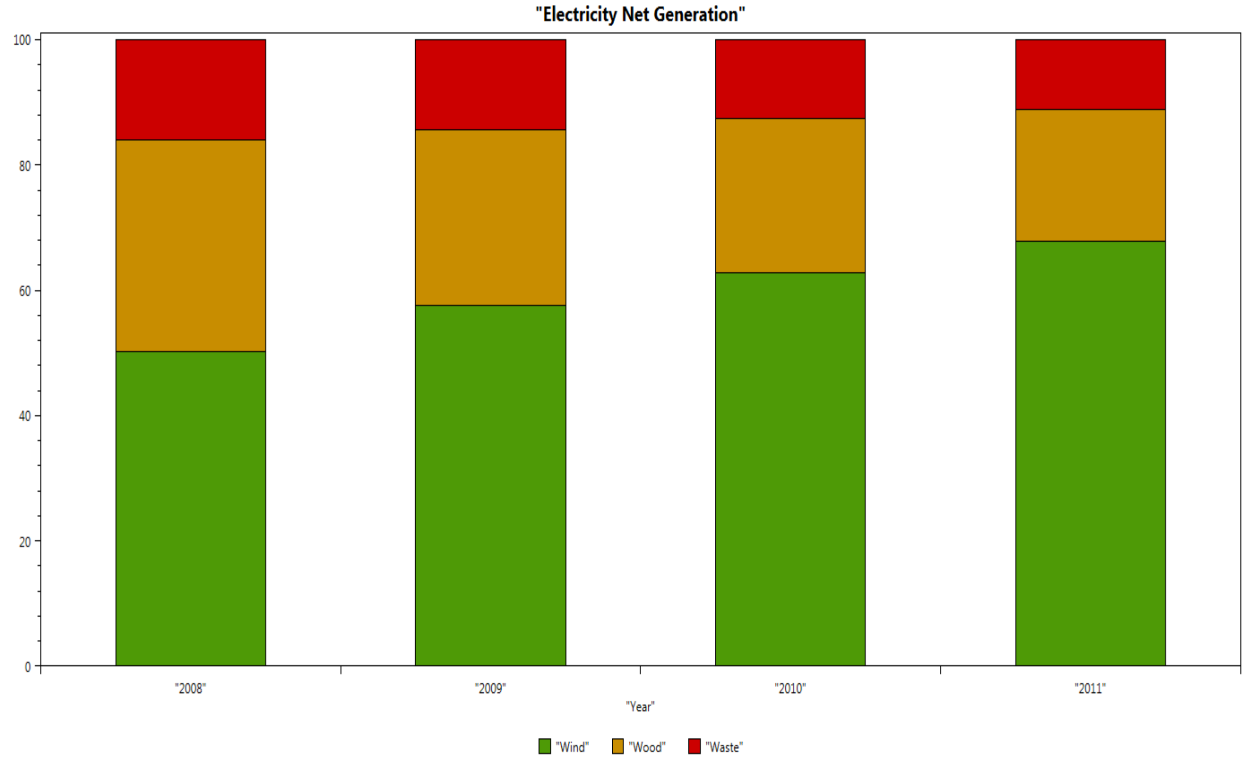


Figure 5 – 100% Stacked bar graph

Range Bar Graph: A sequence of bars with a minimum and maximum value (See Figure 6). The lowest value represents the left end of the bar and the highest value represents the right end of the bar. Additionally an inner line inside the bar can represent the mean or median of the data.

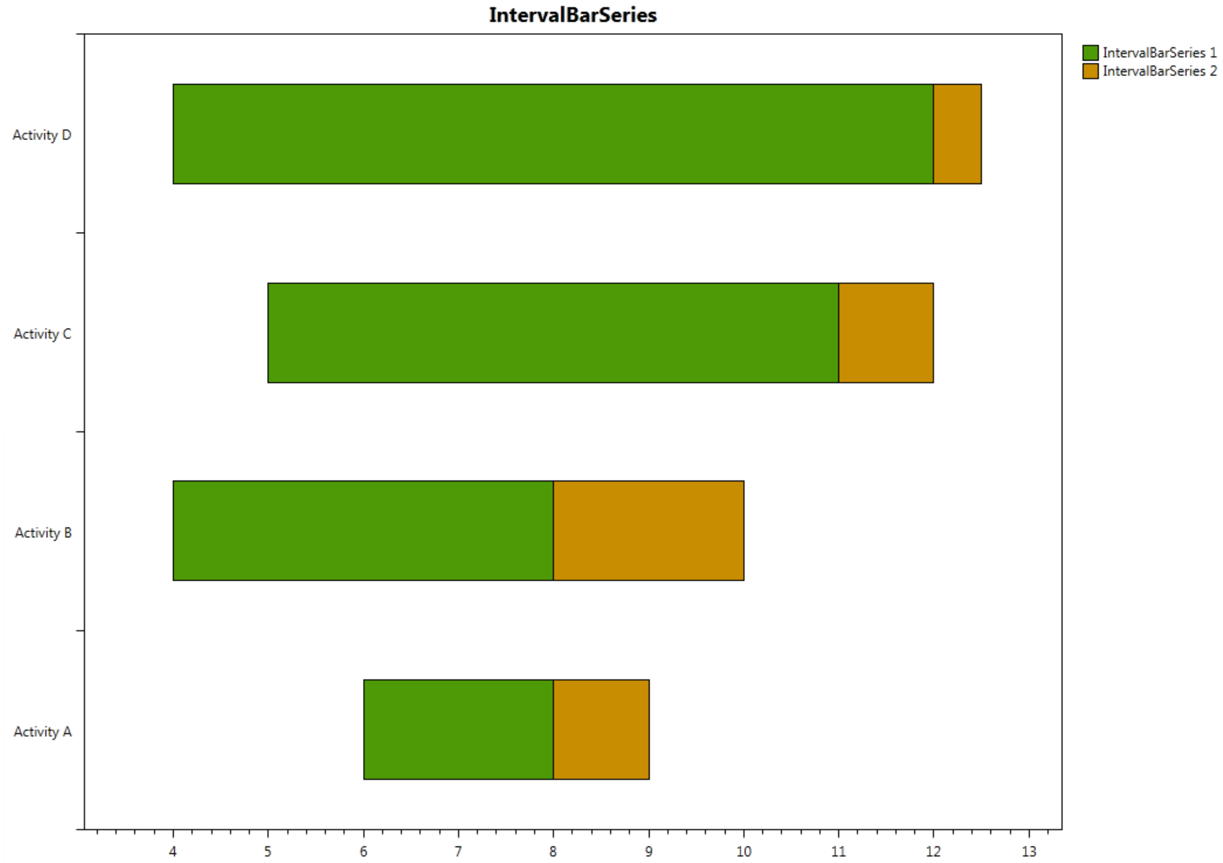


Figure 6 – Range bar graph

Box Plots: This is a range bar representing more information about the distribution of the data (See Figure 7). Besides the mean, confidence intervals are defined with max and min error values.

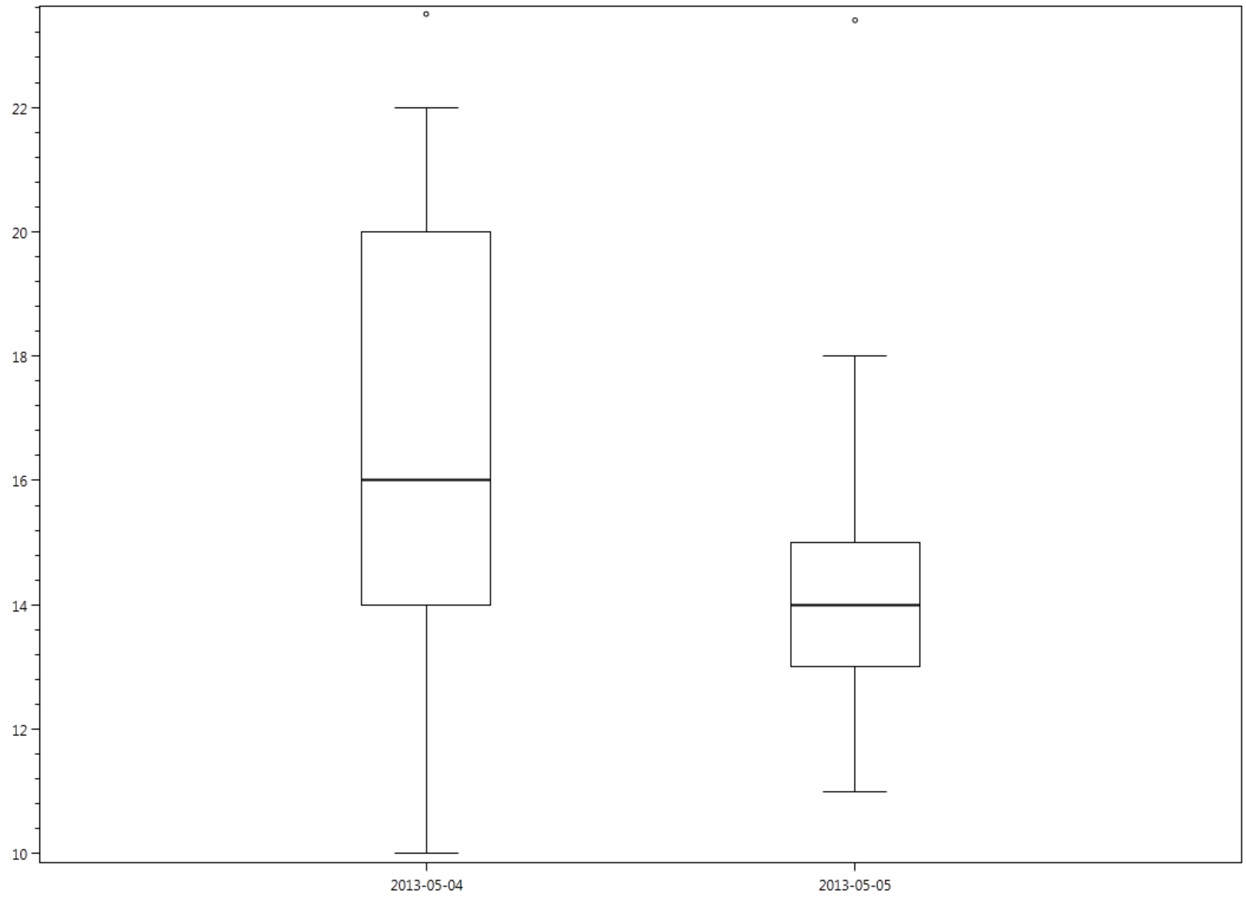


Figure 7 – Box Plot

Histogram: When there is no gap between a sequence of bars and the category axis is numerical then this indicates a histogram (See Figure 8). In a histogram the bars are called bins and usually they represent the cardinality of elements that fall under that category or range values.

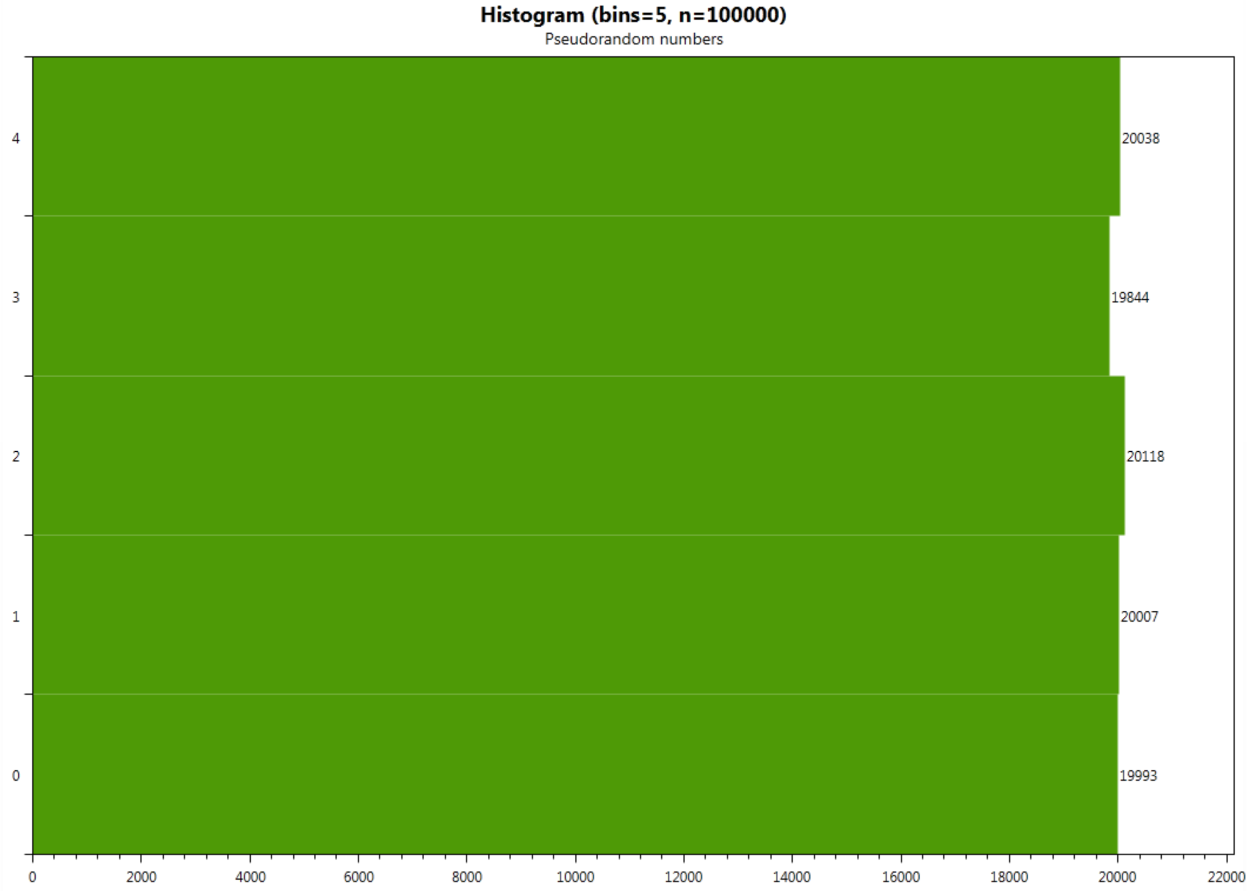


Figure 8 - Histogram

Area bar graph: Bars in a bar graph have different widths (See Figure 9). The width of the bars have significance. The axis can be categorical or numerical in some cases and the width usually has some correlation with the length of the bar.

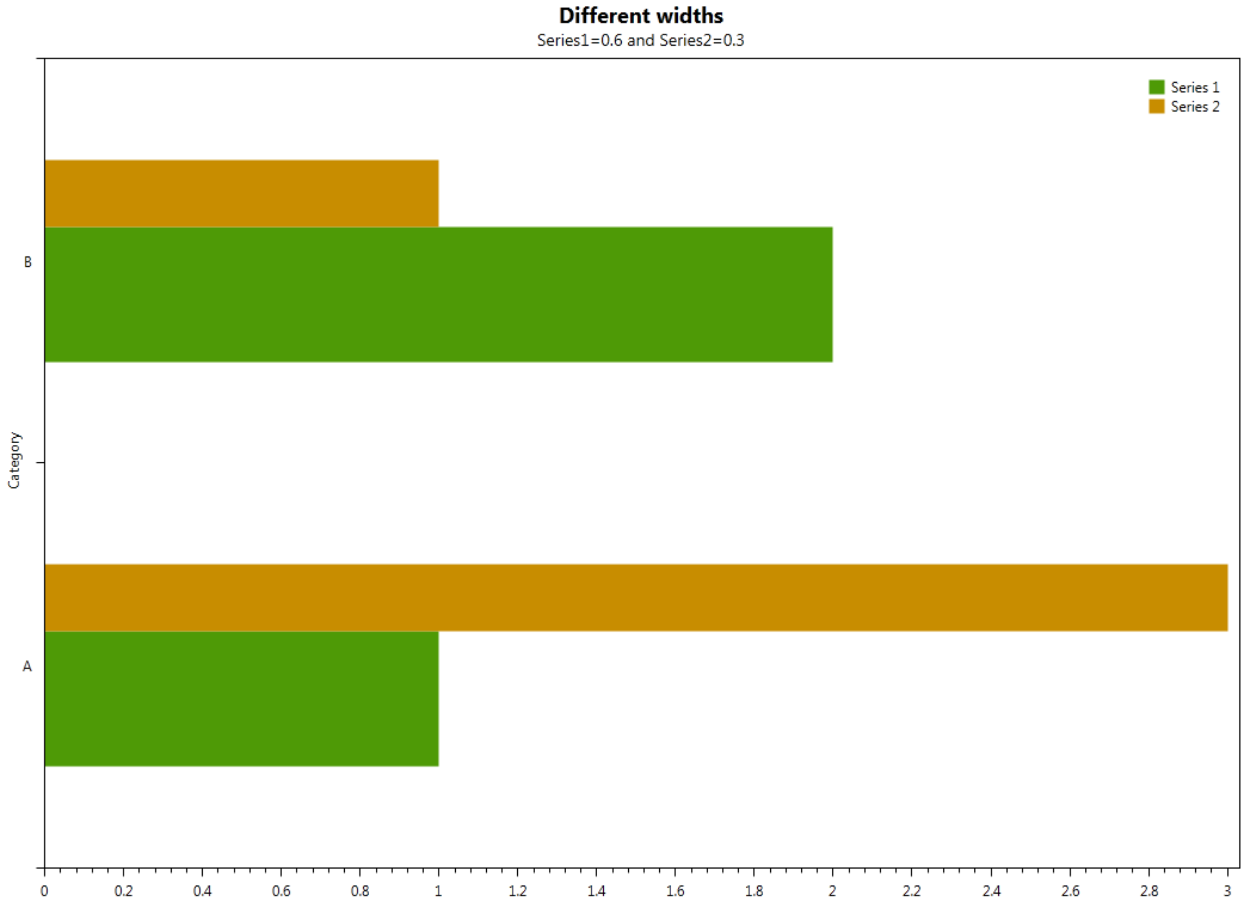


Figure 9 – Area bar graph

Paired bar graph: A variation of a bar graph in which two data series are plotted, one to the right and one to the left (See Figure 10). The purpose of such representation is to compare two data series and find correlations or relationships in the data. The scale of each series does not necessarily have to be the same.

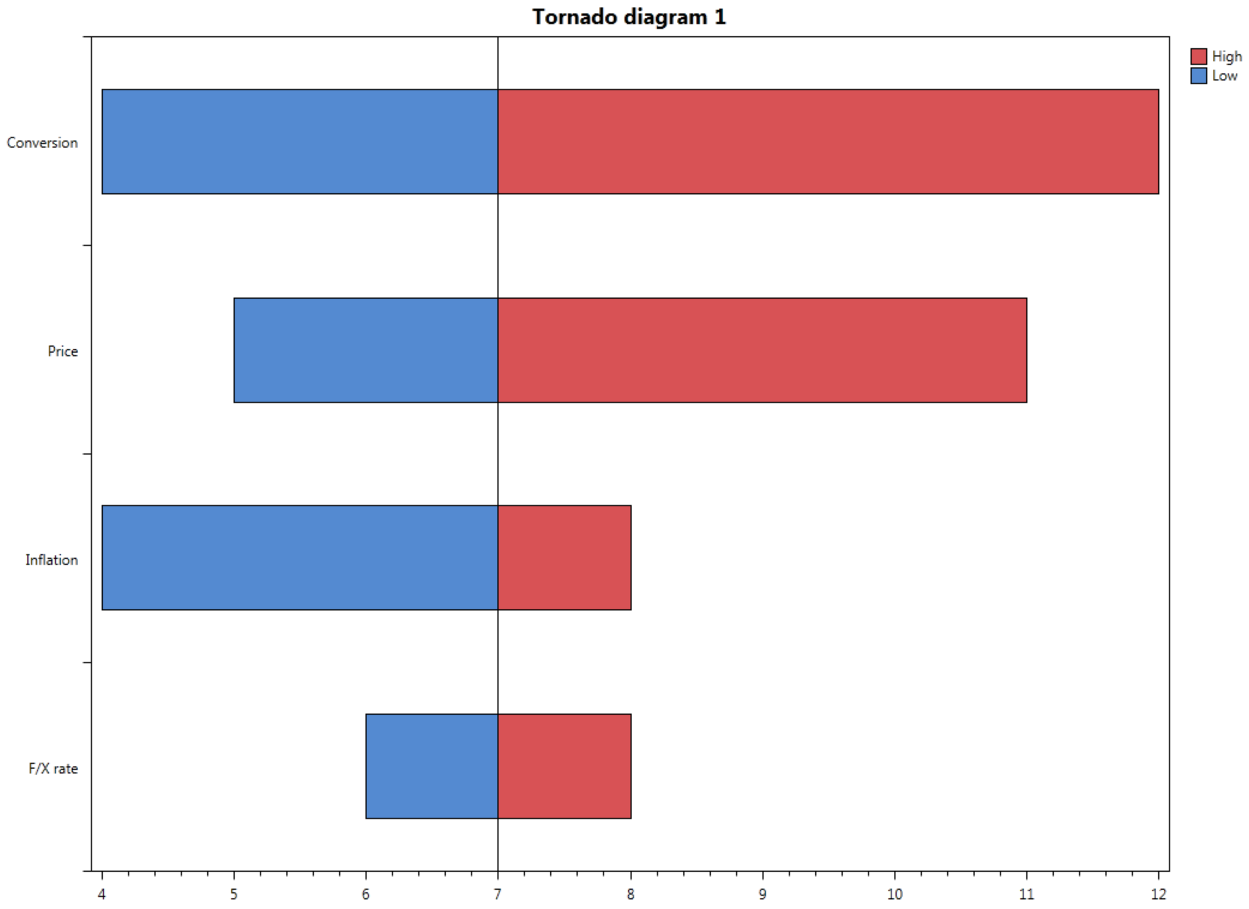


Figure 10 – Paired bar graph

Deviation bar graph: This is a type of bar graph where positive and negative values can be plotted (See Figure 11). Most of the time the values are not plotted in a deviation graph but their differences are. They are usually compared against a certain reference. If a value passes that reference then it is plotted either vertically or horizontally in opposite direction.

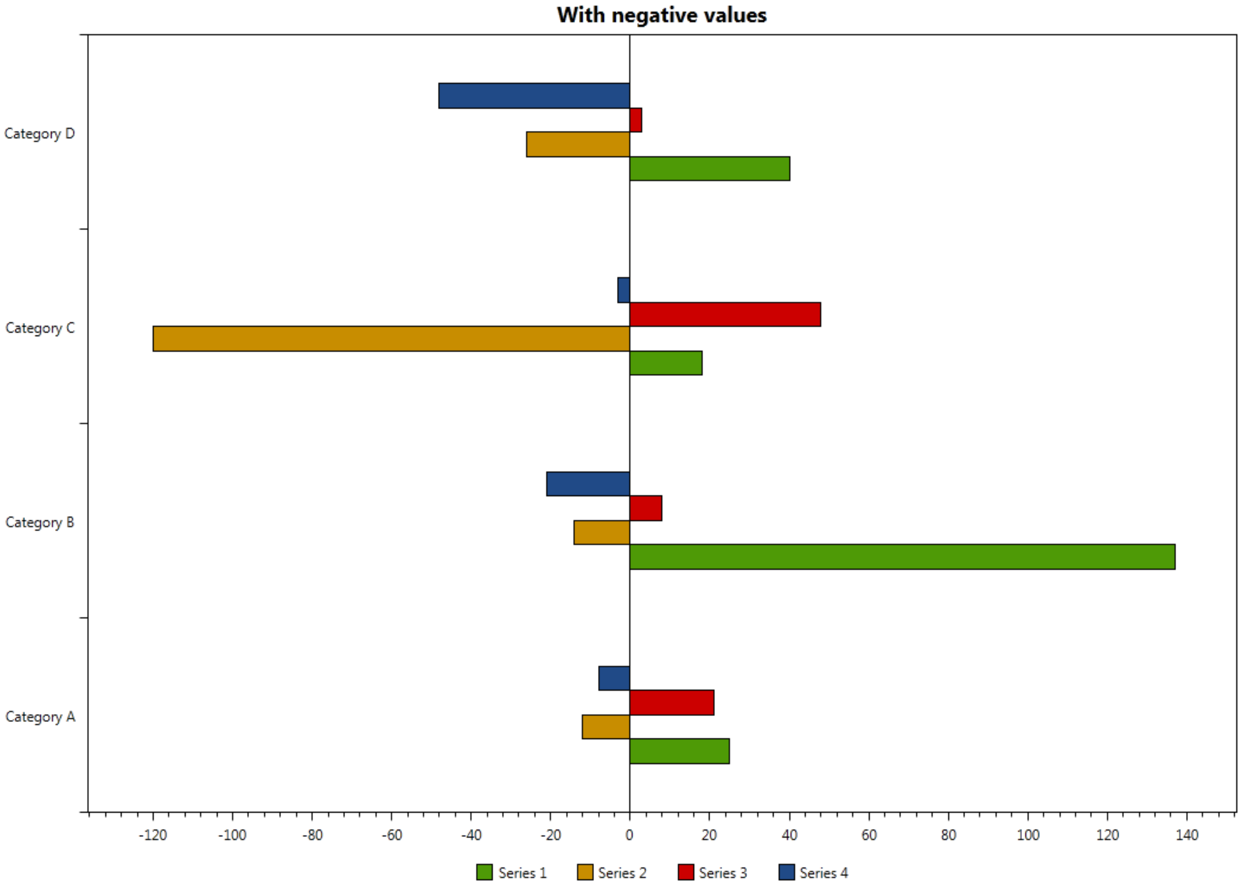


Figure 11 – Deviation bar graph

CHAPTER 4 – USER INTERFACE

In this chapter we will analyze the user interface of the SketChart prototype. Since our goal is to produce a natural user interface for graph generation and interaction, the interaction is based on sketch and gestures. There are two sketch areas A and B. Custom buttons are implemented to interact with the data and a set of gestures is also defined to interact with the recognized graph.

4.1 – User Interface elements

The user interface is composed by two sketching areas. The first area, which we call A, is located on the main window and supports graph rendering and interaction with the graph. The second area, which we call B, is located on a slide panel which is hidden by default. B supports graph sketching and it is opaque to allow area A visualization. Figure 12 shows each one of the user interface elements.

4.1.1 – Sketch Area A

Sketch area A is mainly for interactions with graphs. In this area, graph and sketch buttons are displayed and gestures are performed. A button for the slide panel is positioned on top of the window to show area B. Gestures that can be performed in this area include: erase, sum, selection and filtering.

4.1.2 – Sketch Area B

Sketch area B is located on the slide panel and handles the sketch performed by the user in order to invoke a graph. It is hidden by default but it can be visible by tapping on the arrow located on top of area A. In this interface, buttons to load data, to save strokes, and clear strokes are present.

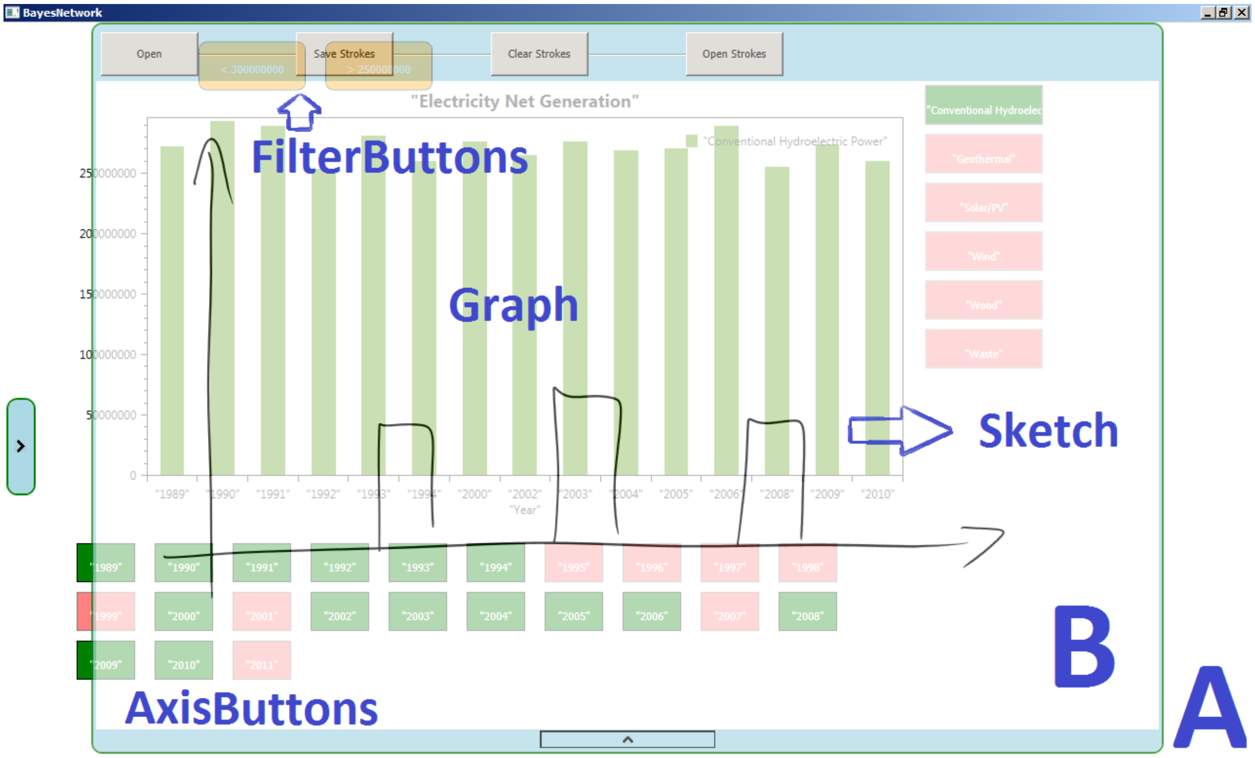


Figure 12 – User Interface elements on the main window of SketChart.

4.2 – Graphical Representation

Sketch Area A will display a chart according to the sketch performed by the user on area B. This graph will be rendered according to the size of the drawing and it could be any of the 10 graph types used for this study. The 10 graphical representations will contain some common user interface elements such as sketch buttons and gestures to interact with it. However, due to the nature of histograms, an additional UI element was added, to change the number of bins. Figure 13 shows a slider set to 10 bins in a histogram.

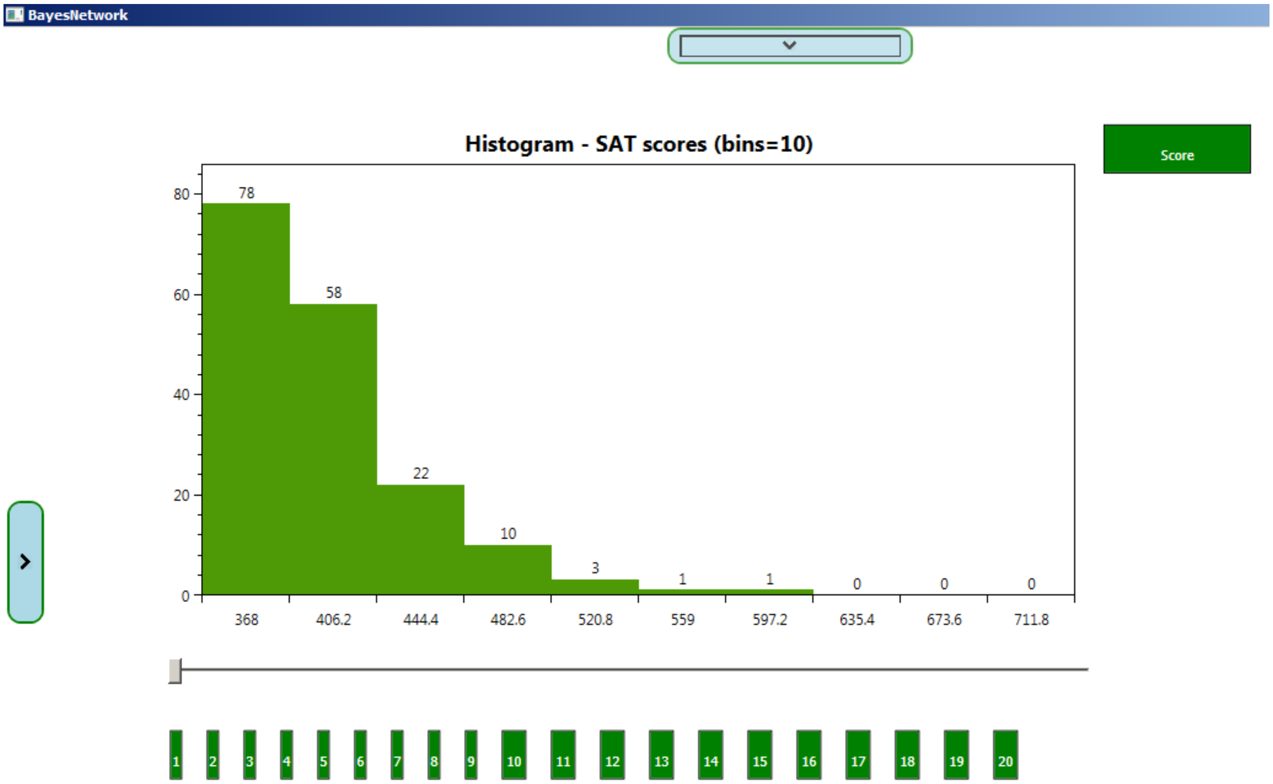


Figure 13 – Histogram with a slider between Sketch buttons and Graph.

4.3 – Sketch Buttons

The use of buttons in the interface was limited by the framework. Normal buttons can be added to the InkCanvas but interacting with them was discouraged due to the type of interaction we wanted to achieve. Customs buttons which we call Sketch buttons where implemented and sketching over them is possible. Two sketch buttons were defined: one for the categories and axis values and the other for saving filtering history. Axis buttons and Filtering buttons can be seen in Figure 12.

Axis buttons are related to the headers and series labels of the dataset. They are in three states: green if the information is present on the graph; otherwise if not present they are in red, the last state is orange which puts them on hold waiting to perform a consequent gesture. These elements support gestures like: tap on a button for changing between red and green state, horizontal or vertical lines that can cross more than one element and toggle the state. Finally, a non-closed circle over them will change their state to orange. In orange state an additional gesture represented by “+” can be performed to merge the data representation of elements.

Filtering buttons are created when a filter gesture is invoked. They are positioned on top of the graphical representation and their color by default is orange when activated and red otherwise. The only gesture they support is tap to change states. Figure 14 shows Sketch buttons in three different states and the gesture drawn to change the state of three of them to orange.

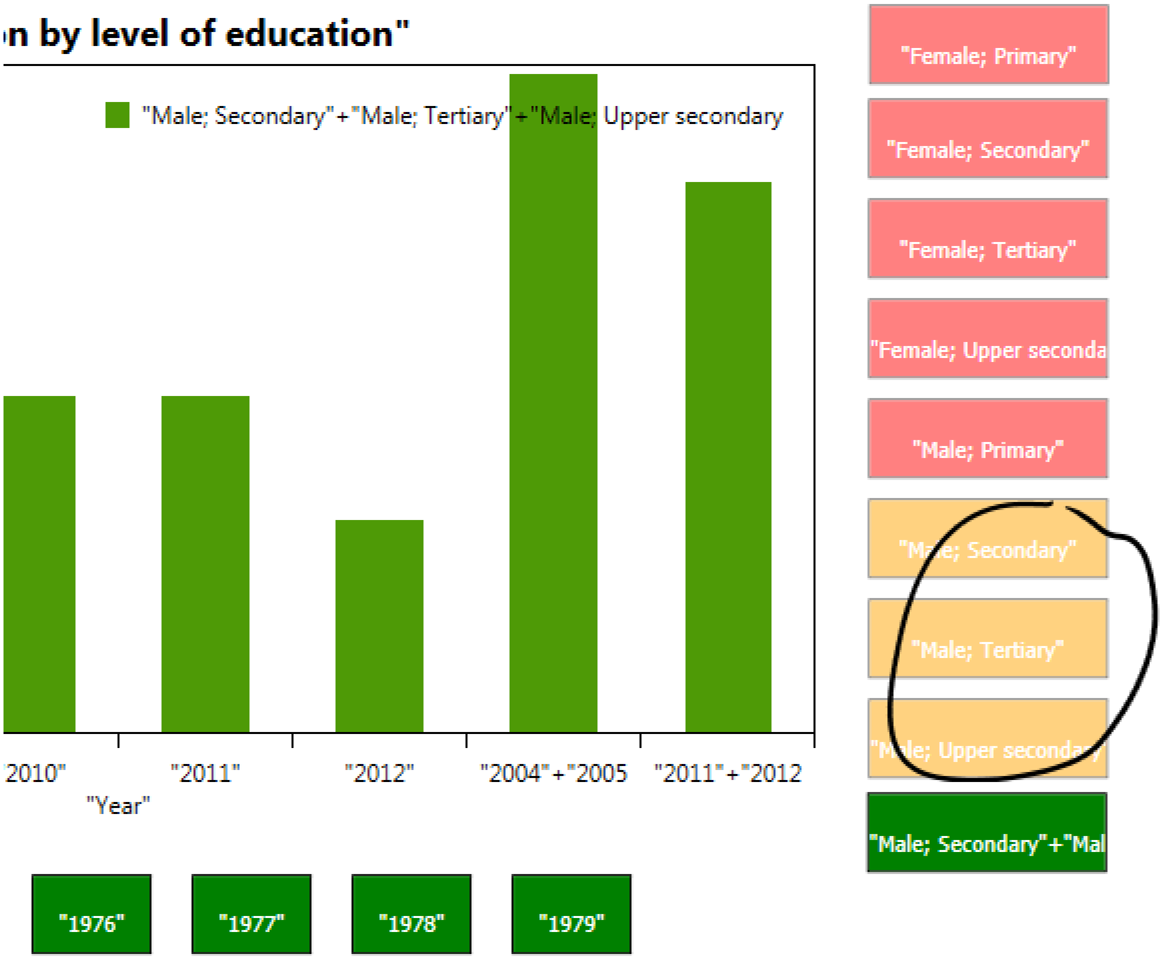


Figure 14 – Sketch buttons in three different states. Circle changes the state to orange, waiting for a gesture to be performed on selected buttons.

4.4 – Gestures

4.4.1 – Scribble

Scribble is a gesture performed to delete strokes. It is invoked by scratching over the strokes. The scratch should contain more than 4 lines and the angle between the lines should be between -45 and 45 degrees. An example is shown in Figure 15.

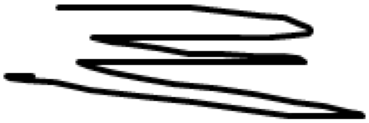


Figure 15 – Example of a scribble gesture

4.4.2 – Tap

Tap has different functionality depending on context. A tap over sketch buttons change the state of them. A tap after performing a gesture executes the gesture functionality. A tap involves a touch with a finger or with the stylus over sketch area A.

4.4.3 – Group Selection and Merging

The axes and the bars on graphs can be merged together by using the “+” gesture. In the axes case, the first is to change the state of the buttons to orange and then perform the gesture followed by a tap. If elements of series on the graph are continuous they can be merged by also placing the “+” gesture on top of the bars in the graph. As can be seen in Figure 16, in the left image non-consecutive years are selected, and then they can be merged by placing a “+” on the graph area. On the other hand on the right image two consecutive years are being merged “2011” and “2012”. The last bar on this image was merged previously and its label is the summation of the labels of the original merged bars.

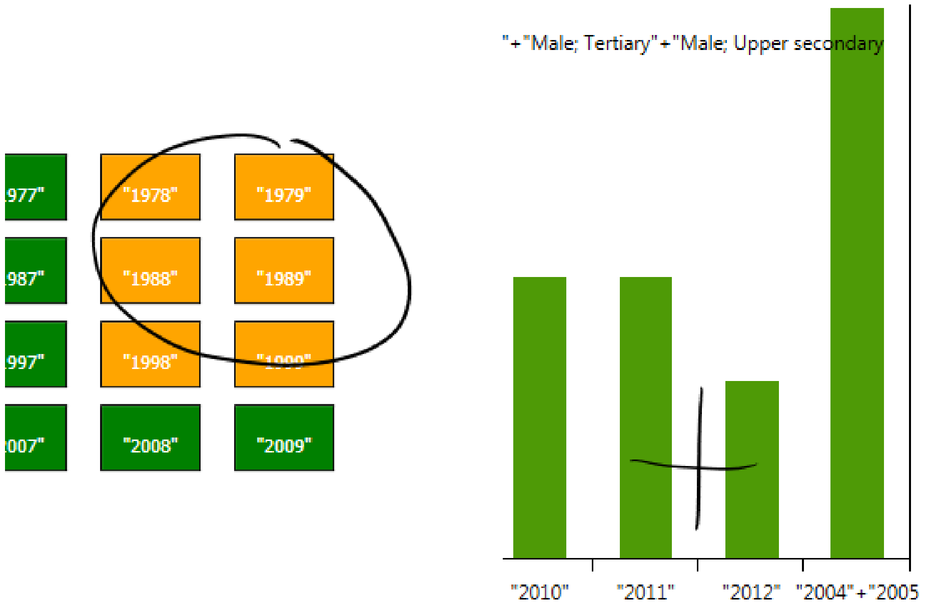


Figure 16 – Example of group selection and merging.

4.4.4 – Hiding bars

Bars on the graph can be hidden by placing an “x” on top of each one or on a set of them. This gesture changes the state of the series from green to red. In Figure 17, three series are hidden by the gesture “x” followed by a tap.

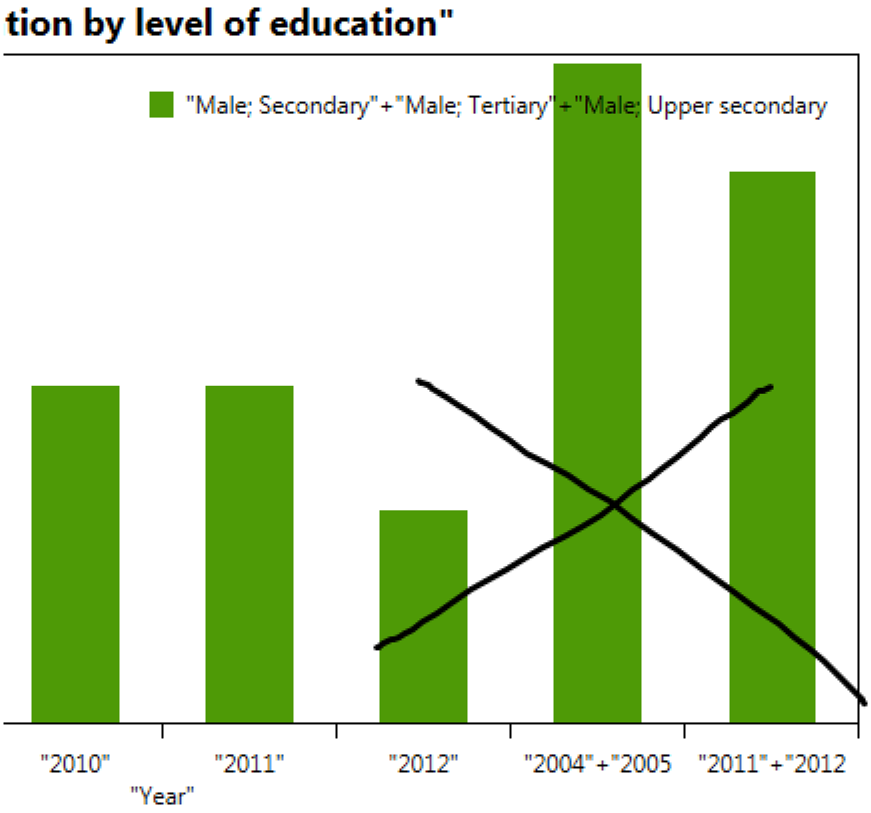

tion by level of education"
🟩 "Male; Secondary"+"Male; Tertiary"+"Male; Upper secondary
"2010"
"2011"
"2012"
"2004"+"2005
"2011"+"2012
"Year"

Figure 17 – Hiding bars from the graph

4.4.5 – Filtering

The series on the graph can be filtered by gestures like "<" and ">". First of all the range value should be underlined and then the gesture should be performed followed by a tap. Figure 18 shows the value underlined and the "<" gesture before tap. Figure 19 shows the result after tap.

An additional interaction technique for filtering is to perform the gesture and then handwrite the number to which the gesture would be applied. A tap is necessary to finish the interaction.

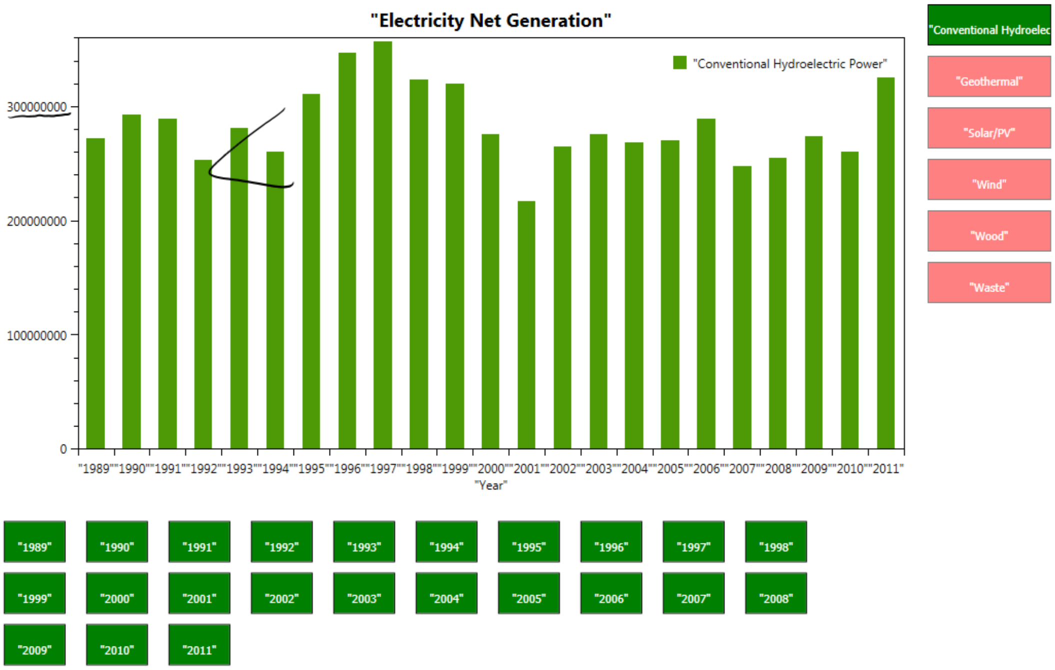


Figure 18 – Performing a filtering gesture on the graph

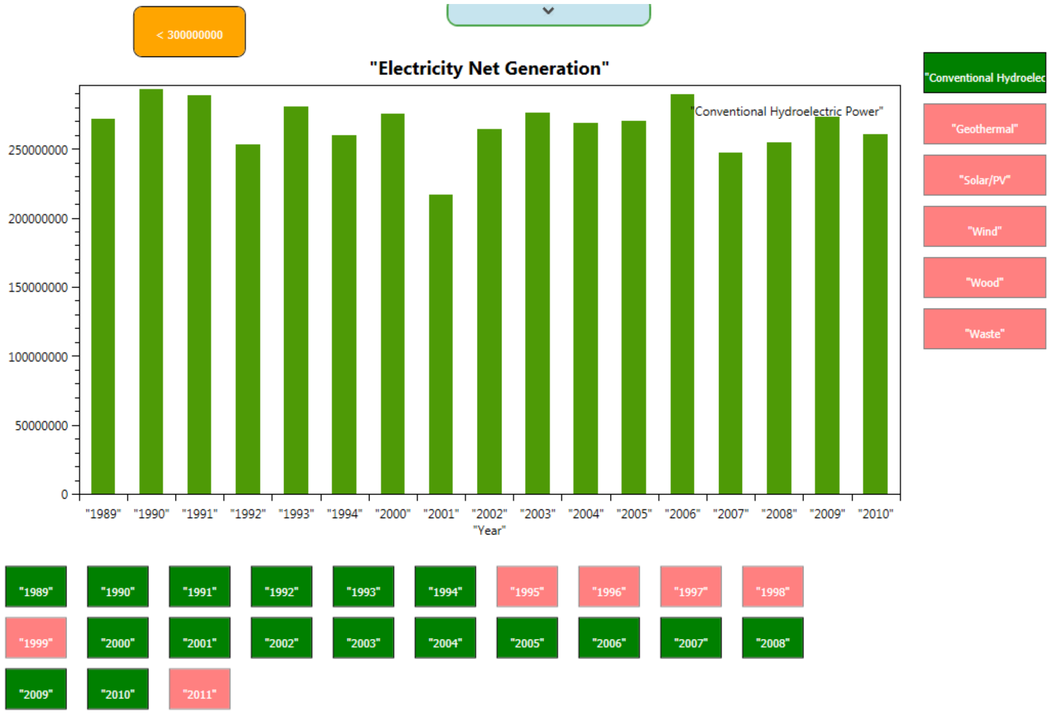


Figure 19 – Result after performing a filtering gesture.

4.5 – Interactive Graph Generation and Manipulation

The key feature of SketChart is that users can generate in an interactive way statistical charts and later manipulate data directly on the graph. To start using the application a file containing the data to plot should be loaded. In real time while the user sketches a graph on area B, a chart, based on the sketch features is rendered on area A. The user is free to modify his initial sketch and the recognition function will adjust the result based on the input given. By hiding the sketching panel, a set of interaction techniques can be applied over the graph. Currently SketChart recognizes 10 charts described in Chapter 2. In Figure 20, a normal flow of interaction is presented: a) the initial window is displayed; b) Sketch area B is pulled from top; c) Axes are drawn and recognized on area A; d) Two groups of bars are drawn and a grouped bar is recognized on the graph canvas; e) Area B is hidden and interaction on the area A can be performed.

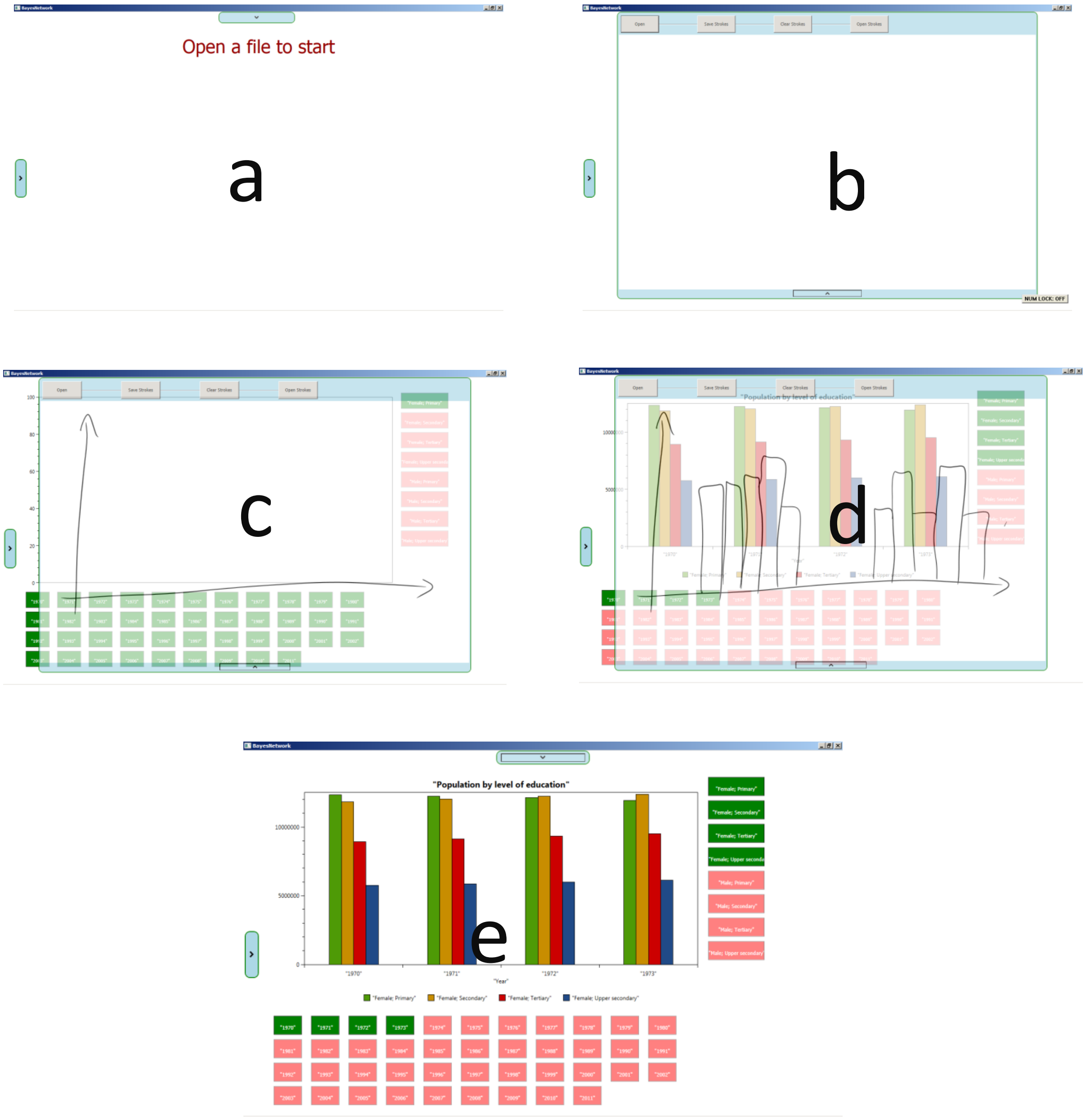


Figure 20 – Interaction flow of SketChart

CHAPTER 5 – ARCHITECTURE

The architecture of the system is based on traditional sketch processing. There are three steps performed before processing a new stroke. We call these steps preprocessing. A filtering algorithm is run over the points to remove noisy data [34]. A corner detector algorithm is then used to divide a stroke in lines which are primitive elements in our system [16]. To complete the preprocessing a segmentation algorithm called MEGIS [17] clusters the strokes by proximity based on the size of the sketch. The recognition engine is based on SketchRead [7] which implements a Bayesian Network to model each one of the components and subcomponents of the system. At the same time, the segmentation algorithm along with SketchRead for ink parsing reduces the search space in the network. The Bayesian network leads to a hierarchical definition of the elements involved in the recognition by the use of a hierarchical description language [18]. Hypotheses are defined as the possible set of components that can be recognized at each level. We have classified them in three categories: low, mid and high level. A data model is defined in order to find an object representation for it. Finally the plots are rendered using an open source library called oxyplot which has a rich variety of graphs and provides mechanisms to interact with the data [19]. In Figure 21, the main modules of the application are colored green.

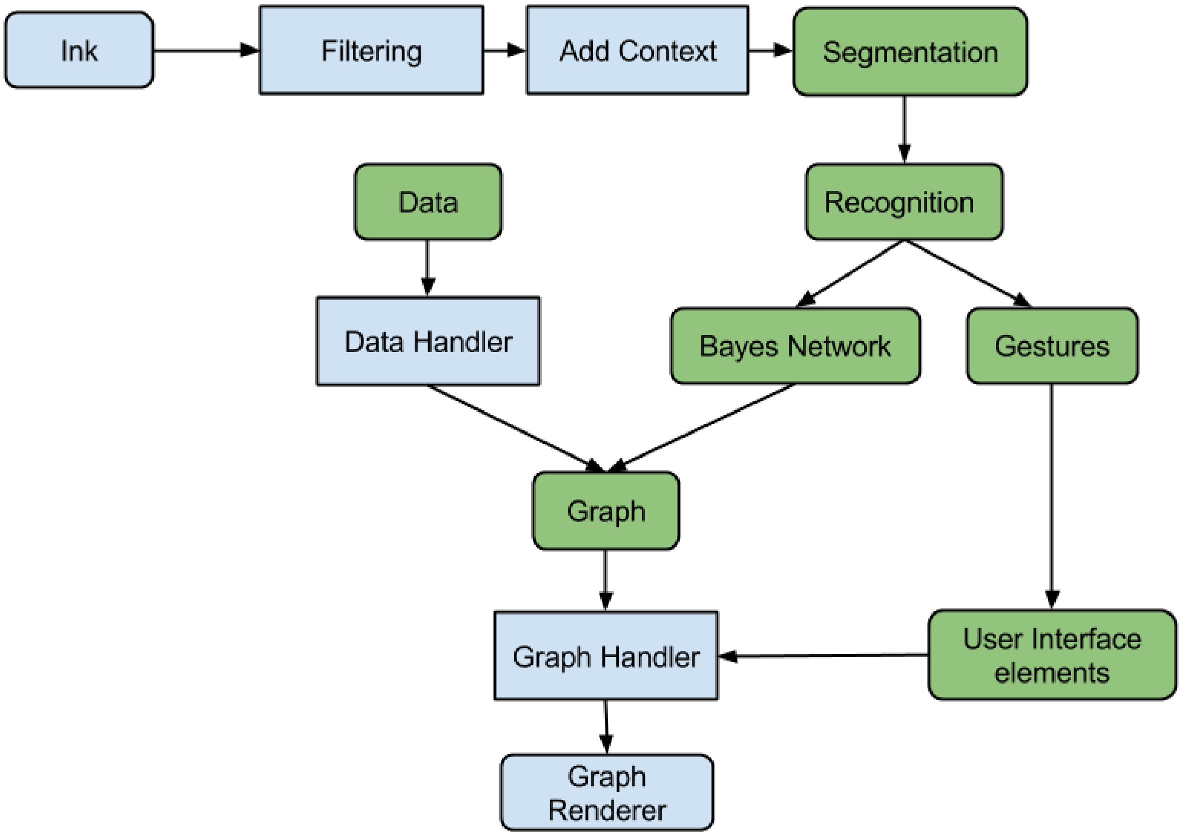


Figure 21 – SketChart Software Architecture

5.1 – Preprocessing

Strokes are a list of points that are displayed on a special Canvas called InkCanvas. This information gathered from the user, most of the time presents duplicate points, self-intersections, strokes segmented unintentionally by lifting unintentionally the pen, mistakes, etc (See Figure Figure 22). Each step of pre-processing is described in the following section.

5.1.1 – Filtering

The filtering process implemented on SketChart involves removing duplicate points and self-intersection. Whether a point is in the same position or closes enough to other point along a stroke, then this point is removed from the stroke [34].

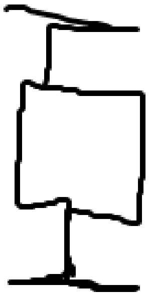


Figure 22 – Example of duplicated points and self-intersections

5.1.2 – Strokes to Lines

After the pre-processing stage, finding corners in our application is fundamental, since lines are our primitive elements in the recognition hierarchical structure. IStraw [16] is the technique chosen to find corners on a stroke and, consequently from the corners, get a list of lines a stroke is composed of. IStraw solves the limitations of a previous technique called ShortStraw [35]. The improvements are related to: add timing information, change dynamically the threshold for a collinear check and improving performance on lines with curves and arcs. Lines are part of every

Shape element in the Bayesian Network structure. In Figure 23, red points on the blue lines indicate a corner found. In total 9 lines are found on this drawing.

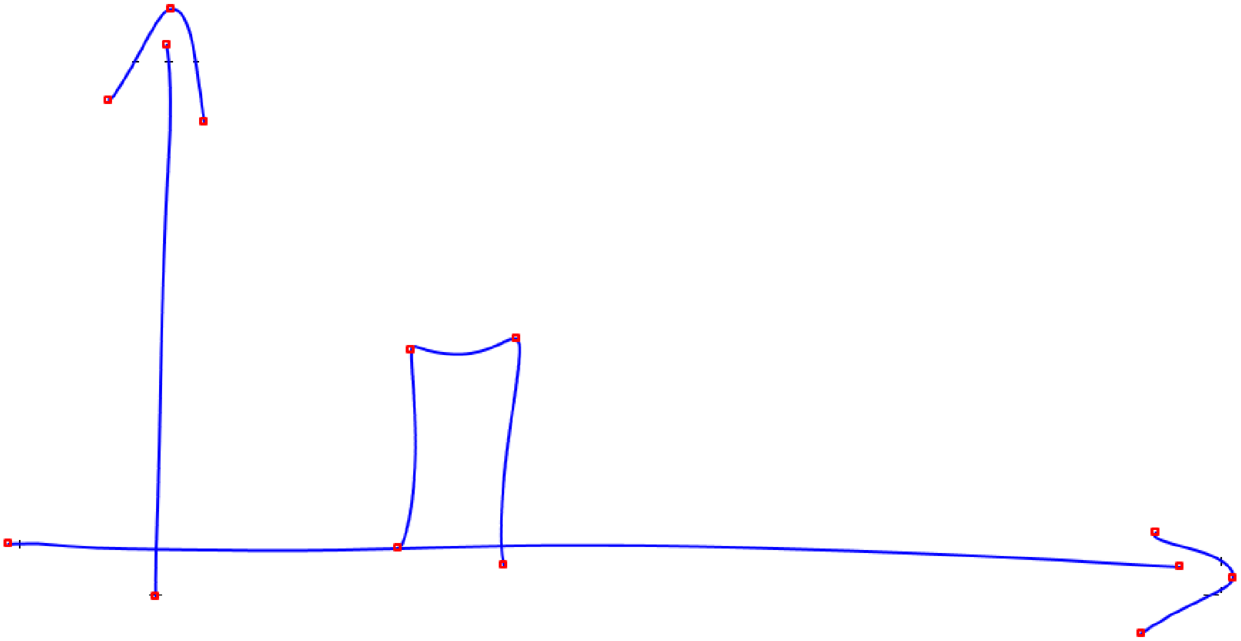


Figure 23 – Corners detected using IStraw

5.1.3 – Segmentation

Segmentation is the process of isolating a stroke or a group of strokes according to some predefined constraints or characteristics. The two common approaches to segmentation are proximity-based [17] and timing [36]. In the first one, the strokes are clustered in base of the proximity of each other to the whole segment. The other one sets a time window for segmenting; all the strokes within that time belong to the ongoing segment drawn.

For this work we decided to use a proximity-based technique due to the fact that the user can probably go back and change a previously defined segment. Modified (EGIS) [17], is a segmentation technique proposed by Sun et. al. EGIS [37] segments pixels of an image by generating a undirected graph in which the vertices represent pixels and the edges represent the dissimilarity between the pixels. MEGIS, adapted to a sketch context, set strokes as vertices and minimum distance between strokes as edges. The process is summarized in the following 5 steps:

1) $Graph\ G(V, E) => V1 = stroke1\ and\ V2 = stroke2; E = \min\{dist(V1, V2)\}$
2) $Int(S) = \max weight\ \{MST(G)\}; MST = Minimum\ Spanning\ Tree$
3) $Tolerance => \ \tau(S) = \frac{k}{|S|}; |S| = \frac{total\ length\ of\ strokes\ in\ S}{maximal\ side\ of\ the\ sketch\ panel}$;

   where k is a constant value that establishes the distance between segments. |S|is changed dynamically depending on the context; when coordinates are not defined, the max side sketch panel is the dimension of the main window, otherwise,
4) $Calculate\ minimal\ internal\ difference\ by$

   $$MInt(S1, S2) = \min\big(Int(S1) + \tau(S1), Int(S2) + \ \tau(S2)\big)$$
5) $Compare\ if\ minimum\ euclidean\ distance\ between\ segments\ is\ less\ than\ MInt.$

   $$Dist(S1, S2) \leq MInt(S1, S2);\ if\ so\ segments\ should\ be\ merged.$$

Two segments can be merged, if the segment is not a closed segment. A segment can be closed if it is already a completed component and it is a Coordinate or a Box. Completed components are shapes that contain all the required elements in the network. Figure 24, shows 4 segments on this graph, each segment is located inside a green rectangle. Coordinates are a completed component and are not merged anymore once completed.

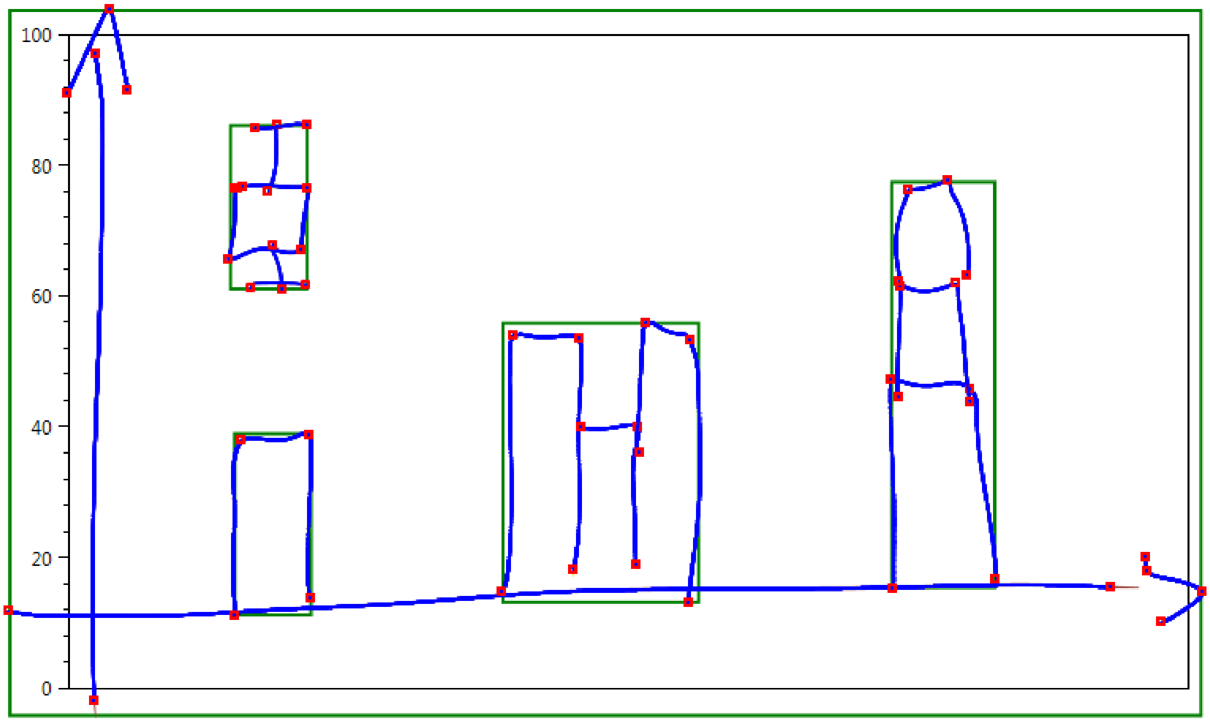


Figure 24 – Components segmented by MEGIS algorithm.

5.2 – Sketch recognition

The recognition phase involves the definition of every shape using a Hierarchical description language, which is largely explained in [18]. A generative approach is taken in order to understand the sketch generated by users. High level components are proposed as hypotheses which will be validated given the lower level hypothesis identified and primitive elements.

5.2.1 – Hierarchical description language

This description language is useful to model the elements of a hierarchical network, specify domain and finally establish interaction between shapes or primitive elements. The following are examples of the definition of Coordinates, stacked bar and 100% stacked bar graph shapes in our system. The complete definition is presented in Appendix A.

Define Coordinates (C)

(**Subshapes** A1, A2: (Arrow x y))

(**Constraints** C1: (perpendicular x.p1 y.p1) C2: (coincident x.p1 y.p1))

Define StackedBar (SB)

(**Subshapes** vector Bar: (B[n]))

(**Constraints** Ci: (OnTop B[i] B[i+1])

Define 100% Stack Bar Graph (SBG)

(**Subshapes** vector StackedBar: (SB[n]))

(**Constraints** C1: (nextTo SB[i] SB[i+1]) C2: (sameHeight SB[i] SB[i+1] … SB[n-1]))

5.2.2 – SketchREAD

SketchREAD [7], is a multi-domain sketch recognition engine which we adapted for the statistical graphs domain. A variation to the algorithm is proposed in this work by using a segmentation algorithm by proximity and reducing the search space to each segment rather than the whole sketch. We believe an improvement can be achieved by doing so. This recognition engine proposes the use of a dynamically constructed Bayesian Network, which generates a probability network with shapes and primitive elements. For each shape, a Bayesian network is generated based on the hierarchical description. A shape has an outcome node and a set of income nodes. Primitive elements have probabilities set based on a previous observation from a pilot study. When a high level hypothesis is completed, depending on the probability (> 60%), a graph is generated.

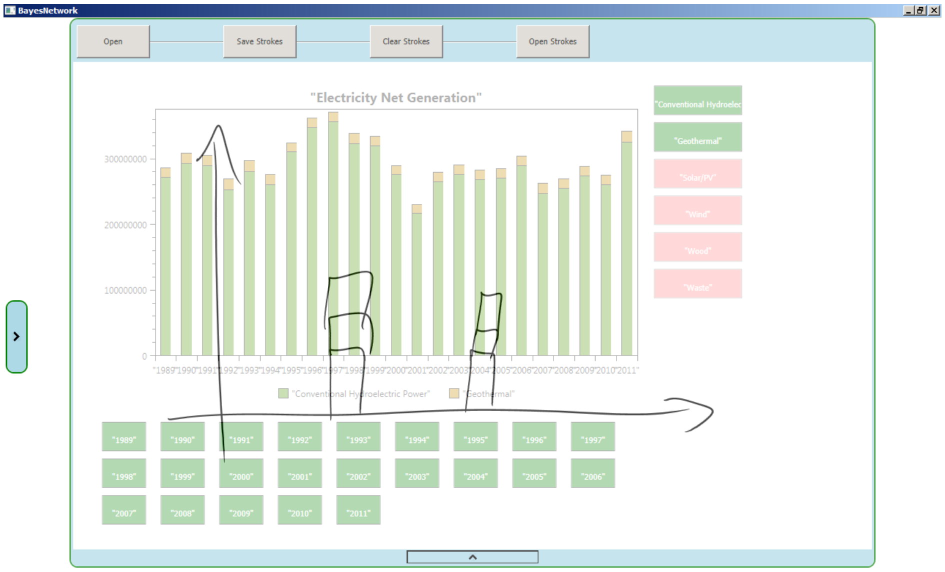


Figure 25 – Sketch of a Stacked Bar Graph to generate probability network

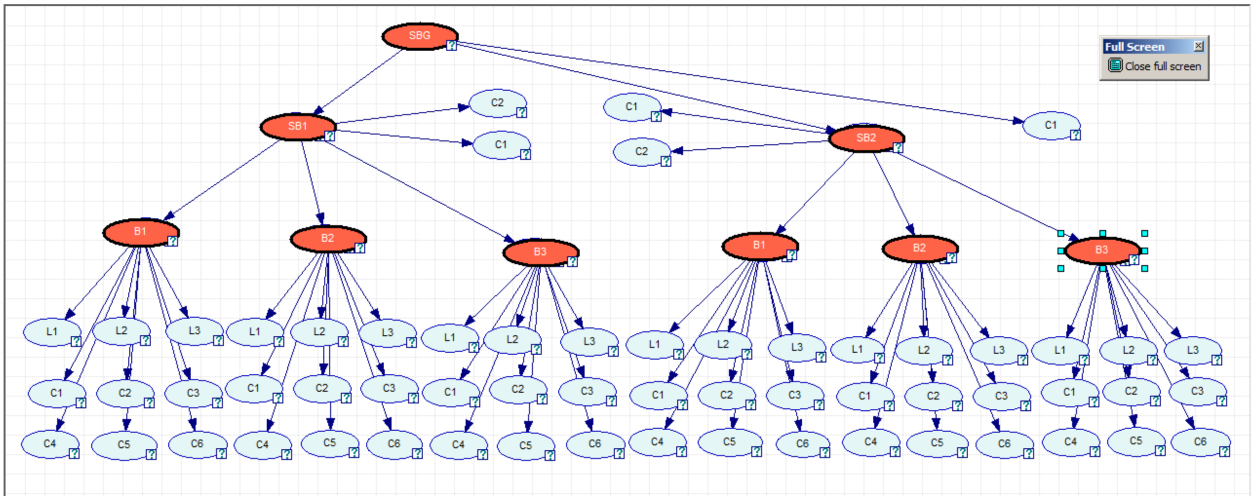


Figure 26 – Bayesian Network generated by the sketch on Figure 25

Figure 26, shows the network generated by the successfully recognized sketch in Figure 25, which is a Stacked Bar Graph. The graph is generated from bottom to top, once a hypothesis in the lower level is complete a new set of hypotheses is generated based on it.

5.2.3 – Bayesian Networks

A Bayes network (BN) is a graphical model represented by a directed acyclic graph (DAG), in which each node has a conditional probabilistic distribution (CPD) [7]. BN has two structures (See Figure 27): diverging structures which we use in our work and converging structure. In a) out-degree is 4 and in b) in-degree is 3. The joint probability in our network is calculated by:

$$P(A, B, C, D, E) \quad = \; P(B/A)\ P(C/A)\ P(D/A)\ P(E/A)\ P(A)$$

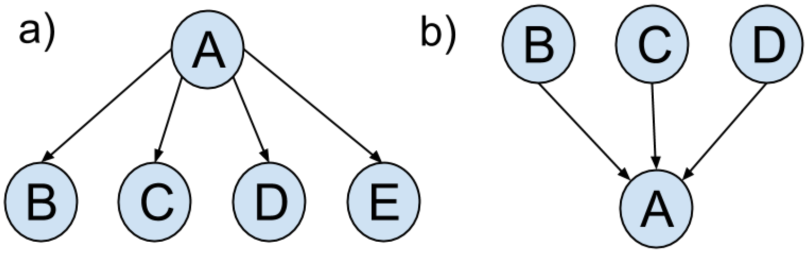


Figure 27 – Bayes Network structure types. a) Diverging structure, b) Converging structure

SMILE.NET is used to model the Bayesian Networks on C#. It has the advantage to implement several graph algorithms and it provides a graphical user interface (GeNIe) to visualize the network as well [39]. In SketChart Shape has as super class Network and each Shape's child class create a node with the parent and children information based on the LADDER description of the component.

5.2.4 – Types of Hypotheses

A Hypothesis is a shape element that is generated based on the state and conditions of the network. For instance, if a set of bars is detected a grouped bar hypothesis would be triggered. In SketChart hypotheses are created based on context. For instance, given an empty canvas, the only hypothesis that can be generated is an arrow in order to generate a coordinate. Once a coordinate is completed a new set of hypotheses is generated for being used for graph generation.

Inside a coordinate, the hypotheses that can be generated are bars, quadrilaterals, grouped bars, but not arrows or coordinates. Constraints are functions to evaluate behavior and characteristics of the different shapes.

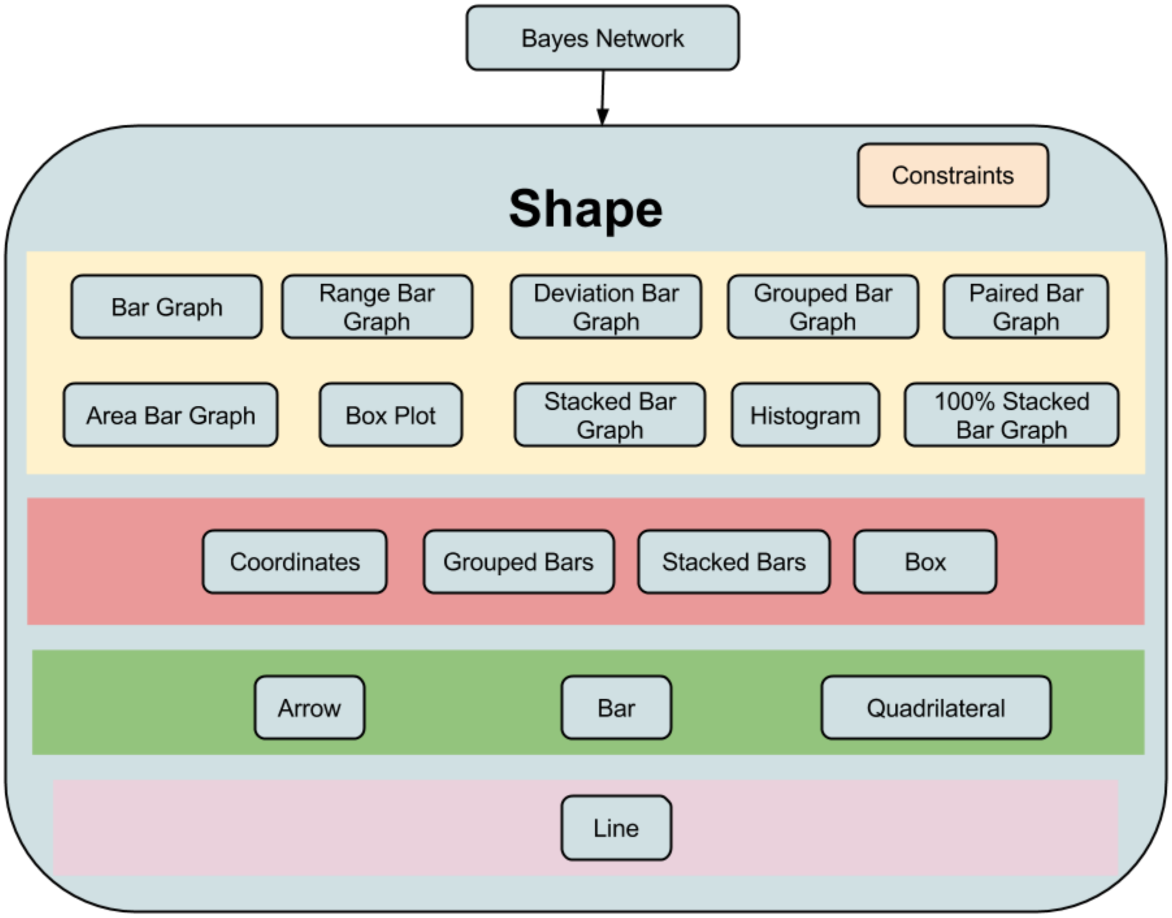


Figure 28 – Possible Hypotheses of SketChart arranged in a hierarchical level

5.2.4.1 – Low-level Hypothesis

Line is our low level Hypothesis as can be seen in Figure 28. In the case of having PieCharts as high level components, we would have Circle at the same level than line. Each one of the elements on top of the low-level hypothesis should have the ability to add a line, or another component.

5.2.4.2 – Mid-level Hypothesis

Hypotheses created between primitive elements and end elements are considered middle level. In Figure 28, second and third levels represent these hypotheses. These hypotheses are composed of primitive elements or other hypotheses at the same level.

5.2.4.3 – High-level Hypothesis

A triggered high level hypothesis will give birth to a new rendered graph, if the probability of the node is the greatest from all the other high level hypotheses. In Figure 28, the 10 possible graphs to be recognized are in the yellow background.

5.2.5 – Hypothesis Evaluation

Each hypothesis has two methods to evaluate in its structure: AddLine and AddComponent. In order to have a complete hypothesis the elements should be completed and the probability should be great than 0.6 which was estimated empirically. Candidate probabilities are first chosen by the greatest value. If so, the hypothesis would be added to the list of completed hypotheses.

5.3 – Gesture recognition

The gestures the user can perform on the system are recognized using a template matching algorithm defined in [38]. This multi-stroke recognizer performs classification by comparing the strokes drawn with a set of training strokes stored on disk. The comparison is given by measuring a score value based on the Euclidean distance between correspondent points. The advantage of $N is that is fast for a small set of samples, it is translate, rotation and scale invariant, and it is easy to implement.

The number of gestures defined for this work is 14. The gestures considered are the numbers from 0-9, '+', '>' and '<'. The number of templates for each gesture is 6 and it was gathered from right-handed people. Gestures are performed on the graph area.

5.4 – Data Model

A file can be uploaded and it is information loaded in memory. The file should be a csv file with headers on top and on the left. This data will be modeled in the application following the object model.

5.5 – Object Model

An object model is passed to the drawing function of all high-level hypotheses. It is composed by a list of generic elements created to handle data coming from the csv. In addition, two dictionaries are defined to hash the string of the header and sub-header with a position and a possible state. All interaction on charts, updates the object model but not the Data model.

5.6 – Chart Renderer

A graph library is used to plot the statistical graph on the canvas. OxyPlot [19] is an open source library to represent data on colorful 2D visualizations.

5.6.1 – OxyPlot

Oxyplot is a cross-platform plotting library which is easy to integrate and use. It has a variety of different charts available in the gallery. SketChart graph rendering is performed by OxyPlot. Interaction over the rendered graph is not possible natively but with positional information a set of gestures can make the graph interactive.

CHAPTER 6 – INFORMAL USER STUDY

A user study was conducted with 7 participants to measure the usability of the application and get some feedback to improve the user interface. The average age of the participants was around 26, 23 being the youngest and 31 the oldest. The majority of the users were male (4). Their majors were engineering (5), accounting (1), and journalism (1). Participants have previously used a tool to generate statistical graphs and visualizations. First, an initial exploration of the system was performed. Sketch and gesture interaction was explained and they had 5 minutes to interact with SketChart. Next, a pre-questionnaire was given to gather some demographical information and previous skills with stylus and statistics. A questionnaire with 10 questions was given to the participant, one question per graph and all questions had to be answered by interacting with the application. Once they were finished with the experiment, a post-questionnaire was given, in order to receive feedback about their experience, and how they compare SketChart with other traditional tools. A copy of the IRB letter and all questionnaires are presented in Appendix B and C respectively.

6.1 – Results

The results show a general acceptance of the system and 6 participants agree that the system can improve time to generate statistical graphs according to the requirements given on the questionnaire. Figure 29 shows that on average the recognition system has a positive evaluation, even though some participants had a tough time while drawing Boxplots. The users consider that the feedback given by the interface is good enough, but they suggested that somehow the system should show a text with the name of the graph recognized. In addition to this, they suggested to show a list with 3 possible graphs to be plotted before rendering the graph on the canvas. Gestures were evaluated as positive, however, due to the fact that training samples were done by a right handed person, a left handed participant had trouble performing "+". An additional gesture that some people had trouble with was the grouping gesture by placing a circle on top of sketch buttons. This can be improved in both cases. Graph visualization aspects in general are positive but an issue arose when users performed the exercises. The issue is related to the size of the rendered graph. Initially, the application is implemented to support more than one graph on the canvas, each graph is created according to the size of the sketch generated. A possible solution is to support multi-touch interaction to scale, translate and rotate the graph.

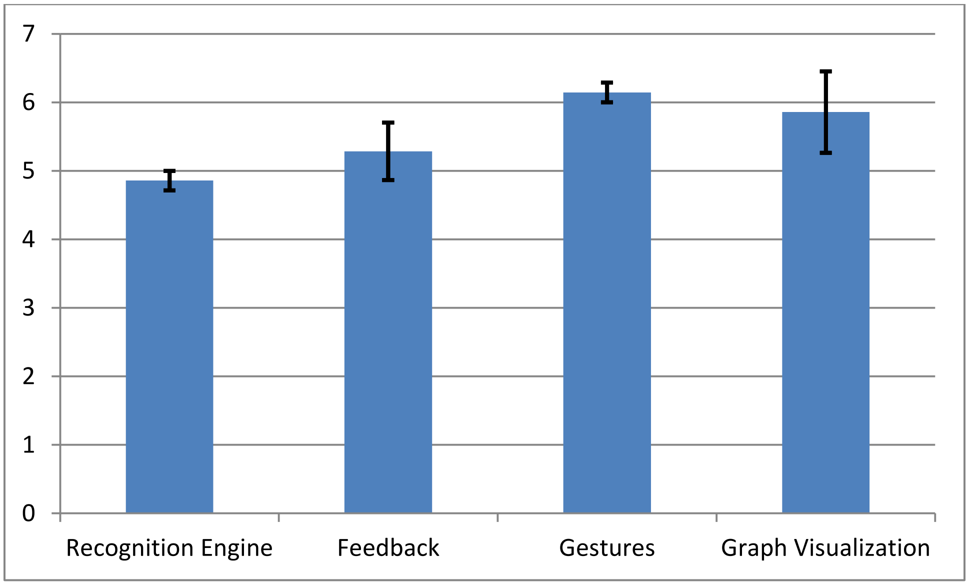


Figure 29 – Average of the aspects rate, error bars are 95% CI.

In order to improve the system, participants suggested, making clearer labels on the graph when several bars are placed on it. Another recommendation involves the recognition engine, a previous step before rendering the graph, showing potential plots for users to select. There was one particular case of a user that sketched everything by one stroke, he had some problems especially with box plots. Grids were also suggested to visualize numbers easily. Finally, range values should be shown on right side as well as in left side, in order to visualize values of elements on the right side easily.

Some comments about the user interface compared with a typical GUI are:

“It is easier to describe the type of graph that I wanted by sketching it instead of selecting from a menu. Also, it is easier to change the graph to show different parts of the data.”

“Sketch based system allow more freedom to create graphs.”

“Smarter and easier to visualize”

“It is more personal and interactive”

“More efficient, more organized and more intelligent”

“It is more comfortable, slightly intuitive, faster and seems to be more powerful. Manipulation of data is way easier since the number of clicks is reduced.”

According to Figure 30, in terms of difficulty, erasing was the hardest to perform because of the scribble gesture. Overall interaction seems to be easy for most of the participants.

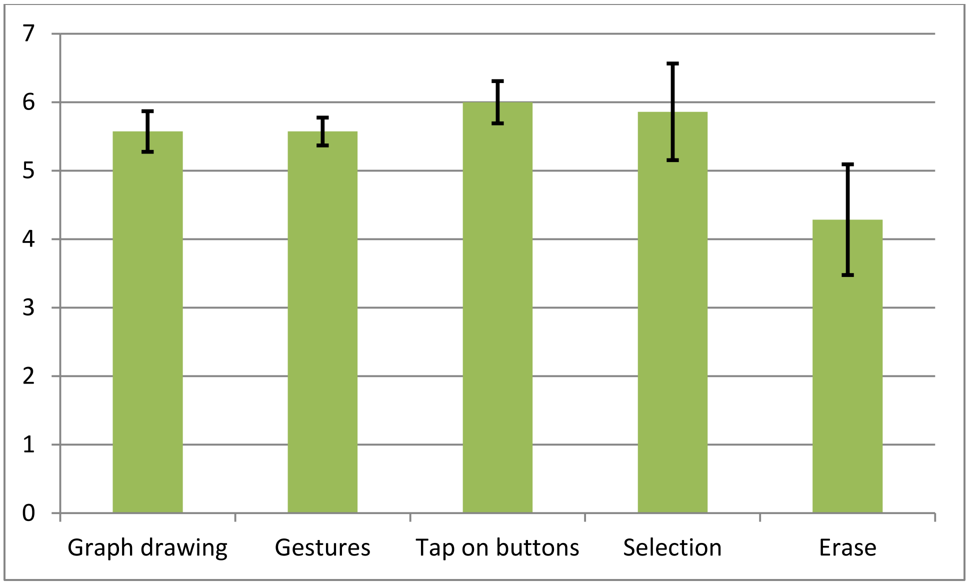


Figure 30 – Ease of use of the interaction techniques, error bars are 95% CI.

Participants suggested the following functionality to the system:

1. In the same way that two series or categories can be summed up, add subtraction to it.
2. Create a gesture for “num1 < x < num2”.
3. Comparing additional files generating a graph for each one.
4. Grids on the graph.
5. Zoom in, Zoom out.

Additional graphs for the next release:

1. Pie Chart
2. Error bar
3. Fitting line

4. Scattered graph
5. Curves

From the 7 participants, 3 would change the way they actually generate graphs, 2 definitely won't change it and 2 maybe.

Analytical Results:

	Total Right	Total Wrong	Total	Accuracy
Coordinates X,Y	122	19	141	86.52
Bargraph	10	1	11	90.91
Grouped Bargraph	10	6	16	62.50
Stacked Bargraph	9	2	11	81.82
100% Stacked Bargraph	12	3	15	80.00
Range Bargraph	9	7	16	56.25
Box plot	10	10	20	50.00
Paired bargraph	8	4	12	66.67
Area Bargraph	8	0	8	100.00
Histogram	8	1	9	88.89
Deviation Bargraph	10	2	12	83.33
Filtering	24	5	29	82.76
+ followed by circle	46	13	59	77.97
Gesture "+"	56	41	97	57.73
Gesture "X"	11	8	19	57.89
Tap	1706	32	1738	98.16
Circle gesture	100	205	305	32.79
Multiple selection	831	3	834	99.64
Erase	21	54	75	28.00

Table 1 – Accuracy for each element on the user interface.

Table 1 shows the accuracy obtained for each one of the elements in the user interface. It can be seen that more work is needed for box plot, range bar graph, grouped and paired bar graph which are the lowest ranked from the statistical chart recognition engine. Gestures need to be improved overall such as erase, circle for selection, “+” and “x” which where ambiguous due to users handwriting. However filtering and multiple selections were the easiest to perform by study participants. Finally, Table 2 shows the general accuracy for the 10 bar based graph recognition engine. Table 3 shows the performance of each participant per question. The overall correct question rate was 84.73%. User study questions are in Appendix C.

Total Right	Total Wrong	Total	Accuracy Graphs
94	36	130	72.31

Table 2 – Accuracy for the bar based recognition engine.

Participant														
	1		2		3		4		5		6		7	
Q.	R	W	R	W	R	W	R	W	R	W	R	W	R	W
1	3	0	3	0	3	0	2	1	3	0	2	1	3	0
2	3	0	3	0	3	0	3	0	3	0	3	0	3	0
3	3	0	3	0	2	1	3	0	3	0	3	0	3	0
4	3	0	3	0	3	0	1	2	3	0	3	0	2	1
5	3	0	2	1	3	0	2	1	3	0	2	1	3	0
6	3	0	2	1	1	2	3	0	3	0	3	0	1	2
7	3	0	2	1	3	0	3	0	3	0	3	0	3	0
8	1	1	2	0	2	0	2	0	2	0	0	2	2	0
9	3	0	3	0	3	0	3	0	2	1	3	0	3	0
10	0	3	0	3	3	0	2	1	2	1	1	2	1	2

Table 3 – Questions answered for the participants. R = right; W = wrong; Q = questions

CHAPTER 7 – FUTURE WORK AND CONCLUSIONS

7.1 – Future work

The recognition engine needs to be improved; one alternative to try would be the use of HMMs to generate a model that can be explored in all directions. More graphs needs to be added to the system. The architecture allows integrating more graphs just by defining another shape class and the primitive elements associated to it. Feedback elements need to be added as well, such as popping out potential graphs from the sketch already generated.

Additionally, exploring the application capabilities on different platforms is worth considering. Another feature to implement is a prediction function, calculated based on the partial hypothesis which could be the most probable next recognized element according to previous experience.

Finally, multi-touch gestures are important, in order to achieve a more fluid interaction. Pinch and zoom are the most common interactions in current market applications and most of the users were expecting this as well.

The ability to generate more than one graph at the same time, and interact with the data from different graphs on the same interface is definitely a great idea. By having two or more graphs and exchanging data between them can make a big difference over current tools.

7.2 – Conclusions

We have presented SketChart, a pen-based tool that allows people to generate graphs, interact, and manipulate data directly on the graph by using a natural user interface. SketChart can be a useful tool for brainstorming and presentations due to the nature of real time interaction. Users are free to modify the graph, without worrying about modifying the data.

A user study was performed to get a general idea of the usability of the system and get some feedback for future improvements. The results have shown a relative acceptance of the prototype and have left clear the weaknesses and strengths of it. People feel comfortable using it and found most of the interaction techniques easy to perform.

Finally, users' feedback to the application has shown that there is still a lot to do, that is very valuable to be explored. A new user study to compare how well SketChart performs against a known application such as Excel is essential.

APPENDIX A – LADDER DEFINITION FOR EACH ELEMENT ON THE PROTOTYPE

Define Arrow(A)

(**Subshapes** L1, L2,L3: (Line shaft, head1, head2))

(**Constraints** C1: (coincident shaft.p1 head1.p1) C2:(coincident shaft.p1 head2.p2) C3:(equal-length head1 head2) C4:(shorter head1 shaft) C5:(acute-angle head1 shatf) C6:(acute-angle head2 shaft))

Define Quadrilateral(Q)

(**Subshapes** (vectorLine L[4]))

(**Constraints** C1:(coincident L[i].p2 L[i+1].p1) C2:(coincident L[4].p2 L[1].p1))

Define Bar(B)

(**Subshapes** (vectorLine L[3]))

(**Constraints** Ci:(coincident L[i].p2 L[i+1].p1) Ci+2:(perpendicular L[i].p2 L[i+1].p1) Ci+4:(equal-length L[i] L[i+2]) Ci+5:(parallel L[i] L[i+2]))

Define Box(Bx)

(**Subshapes** Q, vectorLine L[5]: (Quadrilateral, mean, upper_value, lower_value, upper_limit, lower_limit))

(**Constraints** C1:(inside L[0] Q) C2:(perpendicular L[1].p2 L[3].p1) C3:(perpendicular L[2].p2 L[4].p1) C4:(touches L[1] Q) C5:(touches L[2] Q) C6:(aligned L[1] L[2]))

Define GroupedBar(GB)

(**Subshapes** (vectorBar B[n]))

(**Constraints** Ci:(nextTo B[i] B[i+1]))

Define Coordinates (C)

(**Subshapes** A1, A2: (Arrow x y))

(**Constraints** C1: (perpendicular x.p1 y.p1) C2: (coincident x.p1 y.p1))

Define StackedBar (SB)

(**Subshapes** vector Bar: (B[n]))

(**Constraints** Ci: (OnTop B[i] B[i+1])

Define 100% Stack Bar Graph (TSBG)

(**Subshapes** vector StackedBar: (SB[n]))

(**Constraints** C1: (nextTo SB[i] SB[i+1]) C2: (sameHeight SB[i] SB[i+1] … SB[n-1]))

Define Histogram (H)

(**Subshapes** GroupedBar: (GB))

Define Stack Bar Graph (SBG)

(**Subshapes** vector StackedBar: (SB[n]))

(**Constraints** Ci: (nextTo SB[i] SB[i+1]))

Define Box Plot (BP)

(**Subshapes** vector Box: (Bx[n]))

(**Constraints** Ci: (adjacentTo Bx[i] Bx[i+1]))

Define Area Bar Graph (ABG)

(**Subshapes** vector Bar: (B[n]))

(**Constraints** Ci: (adjacentTo B[i] B[i+1]) Ci+1: (differentWidth B[i] B[i+1] … B[n-1]))

Define Bar Graph (BG)

(**Subshapes** vector Bar: (B[n]))

(**Constraints** Ci: (adjacentTo B[i] B[i+1]))

Define Range Bar Graph (RBG)

(**Subshapes** vector Quadrilateral: (Q[n]))

(**Constraints** Ci: (adjacentTo Q[i] Q[i+1]))

Define Deviation Bar Graph (DBG)

(**Subshapes** vector Bar: (B[n]))

(**Constraints** C1: (sameLine B[i] B[i+1]) C2: (!verifyPaired B[i] B[i+1] … B[n-1]))

Define Paired Bar Graph (PBG)

(**Subshapes** vector Bar: (B[n]))

(**Constraints** C1: (sameLine B[i] B[i+1]) C2: (verifyPaired B[i] B[i+1] … B[n-1]))

Define Group Bar Graph (GBG)

(**Subshapes** vector GroupedBar: (GB[n]))

(**Constraints** Ci: (adjacentTo SB[i] SB[i+1]))

APPENDIX B – UCF IRB LETTER

University of Central Florida Institutional Review Board
Office of Research & Commercialization
12201 Research Parkway, Suite 501
Orlando, Florida 32826-3246
Telephone: 407-823-2901 or 407-882-2276
www.research.ucf.edu/compliance/irb.html

Approval of Human Research

From: **UCF Institutional Review Board #1**
FWA00000351, IRB00001138

To: **Andres Vargas Gonzalez**

Date: **July 01, 2014**

Dear Researcher:

On 7/1/2014, the IRB approved the following human participant research until 06/30/2015 inclusive:

Type of Review:	UCF Initial Review Submission Form
Project Title:	SketChart: A Pen-Based Tool for Chart Generation and Interaction
Investigator:	Andres Vargas Gonzalez
IRB Number:	SBE-14-10398
Funding Agency:	
Grant Title:	N/A
Research ID:	N/A

The scientific merit of the research was considered during the IRB review. The Continuing Review Application must be submitted 30days prior to the expiration date for studies that were previously expedited, and 60 days prior to the expiration date for research that was previously reviewed at a convened meeting. Do not make changes to the study (i.e., protocol, methodology, consent form, personnel, site, etc.) before obtaining IRB approval. A Modification Form **cannot** be used to extend the approval period of a study. All forms may be completed and submitted online at https://iris.research.ucf.edu .

If continuing review approval is not granted before the expiration date of 06/30/2015, approval of this research expires on that date. When you have completed your research, please submit a Study Closure request in iRIS so that IRB records will be accurate.

Use of the approved, stamped consent document(s) is required. The new form supersedes all previous versions, which are now invalid for further use. Only approved investigators (or other approved key study personnel) may solicit consent for research participation. Participants or their representatives must receive a copy of the consent form(s).

All data, including signed consent forms if applicable, must be retained and secured per protocol for a minimum of five years (six if HIPAA applies) past the completion of this research. Any links to the identification of participants should be maintained and secured per protocol. Additional requirements may be imposed by your funding agency, your department, or other entities. Access to data is limited to authorized individuals listed as key study personnel.

In the conduct of this research, you are responsible to follow the requirements of the Investigator Manual.

On behalf of Sophia Dziegielewski, Ph.D., L.C.S.W., UCF IRB Chair, this letter is signed by:

Joanne Muratori

IRB Coordinator

APPENDIX C – QUESTIONAIRES

Participant No:

Pre-Questionnaire

1. What is your age?

2. What is your gender?

   F M

3. What is your major?

4. What hand do you normally favor (writing, sports, ..)?

   5. Left Right

6. Please rate your ability to use statistical graphs to visualize data.

   [1=Not Competent, 5=Very Confident]
   1 2 3 4 5

7. Have you ever used a computer with pen/stylus input before?

   Yes No

8. If you answered “Yes” to the previous question, please describe the task(s) for which you used it.

9. Have you ever used any basic statistical graph tool to visualize data such as Excel?

   Yes No

If you answered ‘Yes’ to the previous question, please list the tool or tools you have used previously.

User Study

SketChart: A Pen-Based Tool for Chart Generation and Interaction

The next 10 questions are not meant to measure your ability in statistical graphs, instead is for measuring the usability of a tool for generating graphs using a sketch based application. We want to evaluate the interaction methods proposed and come up with new if necessary.

1. Taking "Electricity Net Generation.csv" information. Draw a **Bar Graph** and indicate:
    a. In which years the "Conventional Hydroelectric Power" was over 300000000?

    b. Is Solar Electricity generation in years 2004 – 2005 greater than 2006 – 2007?

    c. By simple inspection which of the ways to generate electricity has become the most popular.

2. With the same dataset. Draw a **Grouped Bar Graph** and indicate:
    a. From the six categories presented, which is the category with the most Kw/H produced?

    b. From the graph remove years 1989 – 2010. Taking 2011 year, if you put together the energy generation of "Geothermal", "Solar/PV", "Wind", "Wood", "Waste". Is the sum greater than the energy generated by "Conventional Hydroelectric Power"?

    c. From the graph by selecting "Wind" and "Wood" information, in which years "Wind" electricity generation is greater than "Wood".

3. Now draw a **Stacked Bar Graph** and indicate:
    a. In which year the overall electricity net generation is the highest and in which year the lowest?

    b. In which years the overall electricity generation is between 300000000 and 400000000.

    c. Compare overall energy generation on years 1989, 1990 and 1991 together with overall generation in years 2005 and 2006 together. Which interval generated more electricity?

4. Using the same dataset draw a **100% Stacked Bar Graph** and indicate:
    a. By unselecting “Conventional Hydroelectric Power”. Please indicate with arrows if the energy generation increase, decrease or keep constant from 1989 to 2011.
    “Geothermal” “Solar/PV” “Wind”

    “Wood” “Waste”

    b. By summing up the overall generation of 1989-1990 and 2010-2011. Approximately what is the increase/decrease percentage for “Conventional Hydroelectric Power”?

    c. By summing up the overall generation of “Wind”, “Wood” and “Waste”. Approximately what is the increase/decrease percentage of these three compared with “Hydroelectric Power”?

5. For the next question please load the file “Population by level of education.csv”, draw a **Range Bar Graph** and indicate:
    a. Jot down which is the least range value for female and the year it occurs. Repeat exercise for men.

    b. The year in which the population by level of education is the minimum for males, and the minimum for females. Is the same year?

    c. The year in which the population by level of education is the maximum for males, and the maximum for females. Is the same year?

6. With the same dataset, draw a **Box Plot** and indicate:
    a. The female population average from 1970 to 2011 is approximately in which range?

    b. For Tertiary education level male and female. In which year the population is almost the same, which means that the standard deviation is close to 0 and the confidence intervals are close to the average.

    c. In the graph, show the 3 years in which the average population is the least.

7. For the next set of questions please draw a **Paired Bar Graph**, and indicate:
    a. Is the data symmetric for male and female for each one of the education levels?

    b. Taking primary level of education for male and female. Show in the graph the years in which the populations almost not change.

    c. Join male and female categories and show them in the graph.

8. With the same data, draw an **Area Bar Graph** and indicate:
    a. Select one year and indicate which categories are wider male or female? This will indicate the greatest cardinality in each category.

    b. With the same year join categories in a way that you get the opposite result from previous literal.

9. Please load the file "SAT scores.csv", draw a **Histogram** and indicate:
    a. The maximum frequency for the Math scores with 10 bins.

    b. Number of students with more than 550 points.

    c. Change the number of bins to 5 in the graph.

10. For the next question load the file "Cumulative Trade Deficit/Surplus of Countries.csv" draw a **Deviation Bar Graph** and indicate:
    a. The ratio between the difference between the highest and the lowest of the surplus countries to the difference between Bangladesh and Oman is approximately?

    b. The ratio of the deficit of the first five deficit countries to the overall deficit of all the deficit countries is nearly equal to?

    c. The net total deficit/surplus is equal to?

Participant No.

Post-Questionnaire

This questionnaire asks you to answer questions about the usability, affordability and ease of use of the system used for the study.

1. Using a Sketch-based graph system improve the time to generate a graph according to the requirements given.
   [1=Strongly Disagree, 7=Strongly Agree]
   1 2 3 4 5 6 7

2. How would you rate each one of the following aspects of the system
   [1=Negative, 4=Indifferent, 7=Positive]

Recognition Engine:	1	2	3	4	5	6	7
Feedback:	1	2	3	4	5	6	7
Gestures:	1	2	3	4	5	6	7
Graph visualization:	1	2	3	4	5	6	7

3. In your opinion, how would you improve any of the aspects listed above?

4. In your own words, describe how different is to generate a graph and interact with it in a sketch-based system rather than in a typical graphical user interface (button, keyboard and mouse).

5. From the following interactions indicate the difficulty level represented for each one: [1=Very difficult, 7=Very easy]

Graph drawing:	1	2	3	4	5	6	7
Gestures:	1	2	3	4	5	6	7
Tap on buttons:	1	2	3	4	5	6	7
Selection:	1	2	3	4	5	6	7
Erase:	1	2	3	4	5	6	7

6. List any additional features that you feel might be important in a basic statistical graph system.

7. In your opinion, for a next release of the system which other statistical graphs would you like to be added?

8. In your opinion, would you change the way you are actually generating graphs in a computer? And why?

LIST OF REFERENCES

[1] Priti Shah and James Hoeffner. Review of Graph Comprehension Research: Implications for Instruction. In Educational Psychology Review, Vol. 14, No. 1, March 2002.

[2] William S. Cleveland and Robert McGill. Graphical Perception and Graphical Methods for Analyzing Scientific Data. SCIENCE, Vol 229, August 1985

[3] Jeffrey Browne, Bongshin Lee, Sheelagh Carpendale, Nathalie Riche, and Timothy Sherwood. 2011. Data analysis on interactive whiteboards through sketch-based interaction. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (ITS '11). ACM, New York, NY, USA, 154-157. DOI=10.1145/2076354.2076383 http://doi.acm.org/10.1145/2076354.2076383

[4] Dominikus Baur, Bongshin Lee, and Sheelagh Carpendale. 2012. TouchWave: kinetic multi-touch manipulation for hierarchical stacked graphs. In *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces* (ITS '12). ACM, New York, NY, USA, 255-264. DOI=10.1145/2396636.2396675 http://doi.acm.org/10.1145/2396636.2396675

[5] Lee B, Kazi R H, Smith G. SketchStory: Telling more engaging stories with data through freeform sketching. *IEEE Transactions on Visualization and Computer Graphics*, 2013, to be appeared.

[6] Walney, J., Lee, B., Johns, P., Riche, N.H., Carpendale, S. *Understanding Pen and Touch Interaction for Data Exploration on Interactive Whiteboards.* IEEE Transactions on Visualization and Computer Graphics, 2012, Vol. 18, No.12, pp. 2779 - 2788 http://research.microsoft.com/en-us/um/redmond/groups/cue/publications/TVCG2012-SketchInsigh

[7] Christine Alvarado and Randall Davis. 2004. SketchREAD: a multi-domain sketch recognition engine. In *Proceedings of the 17th annual ACM symposium on User interface software and technology* (UIST '04). ACM, New York, NY, USA, 23-32. DOI=10.1145/1029632.1029637 http://doi.acm.org/10.1145/1029632.1029637

[8] Joseph J. LaViola, Jr. and Robert C. Zeleznik. 2007. MathPad2: a system for the creation and exploration of mathematical sketches. In *ACM SIGGRAPH 2007 courses* (SIGGRAPH '07). ACM, New York, NY, USA, , Article 46 . DOI=10.1145/1281500.1281557 http://doi.acm.org/10.1145/1281500.1281557

[9] Tom Y. Ouyang and Randall Davis. 2011. ChemInk: a natural real-time recognition system for chemical drawings. In *Proceedings of the 16th international conference on Intelligent user interfaces* (IUI '11). ACM, New York, NY, USA, 267-276. DOI=10.1145/1943403.1943444 http://doi.acm.org/10.1145/1943403.1943444

[10] Salman Cheema and Joseph LaViola. 2012. PhysicsBook: a sketch-based interface for animating physics diagrams. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces* (IUI '12). ACM, New York, NY, USA, 51-60. DOI=10.1145/2166966.2166977 http://doi.acm.org/10.1145/2166966.2166977

[11] Marius Brade, Florian Schneider, Angelika Salmen, and Rainer Groh. 2013. OntoSketch: Towards Digital Sketching as a Tool for Creating and Extending Ontologies for Non-Experts. In *Proceedings of the 13th International Conference on Knowledge Management and Knowledge Technologies* (i-Know '13), Stefanie Lindstaedt and Michael Granitzer (Eds.). ACM, New York, NY, USA, , Article 9 , 8 pages. DOI=2494216 http://doi.acm.org/2494216

[12] Paul Schmieder, Beryl Plimmer, and Gillian Dobbie. 2009. Sketching ER diagrams. In *Proceedings of the Tenth Australasian Conference on User Interfaces - Volume 93* (AUIC '09), Paul Calder and Gerald Weber (Eds.), Vol. 93. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 53-60.

[13] Ronald Chung, Petrut Mirica, and Beryl Plimmer. 2005. InkKit: a generic design tool for the tablet PC. In *Proceedings of the 6th ACM SIGCHI New Zealand chapter's international conference on Computer-human interaction: making CHI natural* (CHINZ '05). ACM, New York, NY, USA, 29-30. DOI=10.1145/1073943.1073950 http://doi.acm.org/10.1145/1073943.1073950

[14] Fornés, Alicia and Escalera, Sergio and Lladós, Josep and Sánchez, Gemma and Mas, Joan. 2008. Hand Drawn Symbol Recognition by Blurred Shape Model Descriptor and a Multiclass Classifier. In Graphics Recognition. Recent Advances and New Opportunities. Springer Berlin Heidelberg, DOI= 10.1007/978-3-540-88188-9_4 http://dx.doi.org/10.1007/978-3-540-88188-9_4

[15] Robert L. Harris. 1999. Information Graphics: A Comprehensive Illustrated Reference. Oxford University Press, 1999

[16] Yiyan Xiong and Joseph J. LaViola, Jr.. 2009. Revisiting ShortStraw: improving corner finding in sketch-based interfaces. In *Proceedings of the 6th Eurographics Symposium on Sketch-Based Interfaces and Modeling* (SBIM '09), Dieter Fellner and Stephen Spencer (Eds.). ACM, New York, NY, USA, 101-108. DOI=10.1145/1572741.1572759 http://doi.acm.org/10.1145/1572741.1572759

[17] Zhenbang Sun, Changhu Wang, Liqing Zhang, and Lei Zhang. 2012. Free hand-drawn sketch segmentation. In *Proceedings of the 12th European conference on Computer Vision - Volume Part I* (ECCV'12), Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid (Eds.), Vol. Part I. Springer-Verlag, Berlin, Heidelberg, 626-639. DOI=10.1007/978-3-642-33718-5_45 http://dx.doi.org/10.1007/978-3-642-33718-5_45

[18] Tracy Hammond and Randall Davis. 2007. LADDER, a sketching language for user interface developers. In *ACM SIGGRAPH 2007 courses* (SIGGRAPH '07). ACM, New York, NY, USA, , Article 35 . DOI=10.1145/1281500.1281546 http://doi.acm.org/10.1145/1281500.1281546

[19] https://oxyplot.codeplex.com/ , 2014

[20] Ankerst M, Keim DA, Kriegel HP. 1996. Circle segments: a technique for visually exploring large multidimensional data sets. IEEE Symposium on Visualization ’96, Hot Topics, (San Francisco, CA, USA, 1996).

[21] Daniel A. Keim, Ming C Hao, Umesh Dayal, and Meichun Hsu. 2002. Pixel bar charts: a visualization technique for very large multi-attribute data sets. *Information Visualization* 1, 1 (March 2002), 20-34. DOI=10.1057/palgrave/ivs/9500003 http://dx.doi.org/10.1057/palgrave/ivs/9500003

[22] Danyel Fisher, Igor Popov, Steven M. Drucker, and mc schraefel, Trust Me, I'm Partially Right: Incremental Visualization Lets Analysts Explore Large Datasets Faster, in *Proceedings of the 2012 Conference on Human Factors in Computing Systems (CHI 2012)*, ACM Conference on Human Factors in Computing Systems, 5 May 2012

[23] http://www-958.ibm.com/software/data/cognos/manyeyes/visualizations, 2014

[24] http://www.huffingtonpost.com/2012/04/25/drawing-ability-psychology_n_1451748.html, 2012

[25] Kurzweil, Ray (2012), *How to Create a Mind: The Secret of Human Thought Revealed*, New York: Viking Books, ISBN 978-0-670-02529-9

[26] Mathias Eitz, James Hays, and Marc Alexa. 2012. How do humans sketch objects?. *ACM Trans. Graph.* 31, 4, Article 44 (July 2012), 10 pages. DOI=10.1145/2185520.2185540 http://doi.acm.org/10.1145/2185520.2185540

[27] http://www.icharts.net/. 2014.

[28] http://charts.hohli.com. 2014.

[29] http://www.chartgo.com/index.jsp. 2014.

[30] http://www.tableausoftware.com. 2014

[31] Bongshin Lee, Petra Isenberg, Nathalie Henry Riche, Sheelagh Carpendale: Beyond Mouse and Keyboard: Expanding Design Considerations for Information Visualization Interactions. IEEE Trans. Vis. Comput. Graph. 18(12): 2689-2698 (2012)

[32] Leo Ferres, Gitte Lindgaard, and Livia Sumegi. 2010. Evaluating a tool for improving accessibility to charts and graphs. In *Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility* (ASSETS '10). ACM, New York, NY, USA, 83-90. DOI=10.1145/1878803.1878820 http://doi.acm.org/10.1145/1878803.1878820

[33] Jeffrey Heer and Ben Shneiderman. 2012. Interactive dynamics for visual analysis. *Commun. ACM* 55, 4 (April 2012), 45-54. DOI=10.1145/2133806.2133821 http://doi.acm.org/10.1145/2133806.2133821

[34] LaViola, J. "Mathematical Sketching: A New Approach to Creating and Exploring Dynamic Illustrations", Ph.D. Dissertation, Brown University, Department of Computer Science, May 2005.

[35] A. Wolin, B. Eoff, and T. Hammond. 2008. ShortStraw: a simple and effective corner finder for polylines. In *Proceedings of the Fifth Eurographics conference on Sketch-Based Interfaces and Modeling* (SBM'08), Christine Alvarado and Marie-Paule Cani

(Eds.). Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 33-40. DOI=10.2312/SBM/SBM08/033-040 http://dx.doi.org/10.2312/SBM/SBM08/033-040

[36] Tevfik Metin Sezgin and Randall Davis. 2007. Temporal sketch recognition in interspersed drawings. In *Proceedings of the 4th Eurographics workshop on Sketch-based interfaces and modeling* (SBIM '07). ACM, New York, NY, USA, 15-22. DOI=10.1145/1384429.1384436 http://doi.acm.org/10.1145/1384429.1384436

[37] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. 2004. Efficient Graph-Based Image Segmentation. *Int. J. Comput. Vision* 59, 2 (September 2004), 167-181. DOI=10.1023/B:VISI.0000022288.19776.77 http://dx.doi.org/10.1023/B:VISI.0000022288.19776.77

[38] Anthony, L. and Wobbrock, J.O. (2010). A lightweight multistroke recognizer for user interface prototypes. Proceedings of Graphics Interface (GI '10). Ottawa, Ontario (May 31-June 2, 2010). Toronto, Ontario: Canadian Information Processing Society, pp. 245-252.

[39] http://genie.sis.pitt.edu/, 2014