A webtool for e-textiles design

Roope Lehikoinen

School of Electrical Engineering

Bachelor's thesis
Espoo 12.10.2019

Supervisor

Prof. Samuli Aalto

Advisor

Asst. Yu Xiao

Aalto University
School of Electrical
Engineering

Aalto University
School of Electrical Engineering

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the bachelor's thesis

Author Roope Lehikoinen	
Title A webtool for e-textiles design	
Degree programme Automation and information technology	
Major Information Technology	**Code of major** ELEC3015
Teacher in charge Prof. Samuli Aalto	
Advisor Asst. Yu Xiao	
Date 12.10.2019	**Number of pages** 10	**Language** English

Abstract

Keywords E-textiles, smart textiles, weaving, web application, internet application, database, software

Aalto University
School of Electrical Engineering

Aalto-yliopisto, PL 11000, 00076 AALTO
www.aalto.fi
Tekniikan kandidaatintyön tiivistelmä

Tekijä Roope Lehikoinen	
Työn nimi A webtool for e-textiles design	
Koulutusohjelma Automaatio- ja informaatioteknologia	
Pääaine Informaatioteknologia	**Pääaineen koodi** ELEC3015
Vastuuopettaja Prof. Samuli Aalto	
Työn ohjaaja Asst. Yu Xiao	
Päivämäärä 12.10.2019 **Sivumäärä** 10	**Kieli** Englanti

Tiivistelmä

Avainsanat E-tekstiili, älytekstiili, kudonta, web-sovellus, tietokanta, tietokoneohjelma

Contents

Symbols and abbreviations

Abbreviations

TIFF	Tagged image file format
GPU	Graphics processing unit
API	Application programming interface
2D	Two dimensional
3D	Three dimensional
HTML	Hypertext markup language
WebGL	Web graphics library
TC2	Thread controller 2

1 Introduction

Electronic textiles (e-textiles) have been around for two centuries and are still a growing industry. They are wearable pieces of clothing that have conductive materials woven into the fabric. Since the late 1800s people have worn clothing with conductive materials, like for example gold or silver. Only later have designers started using the conductivity to create circuitry out of clothing. Most early uses involved integrating some forms of lights into clothing and jewellery. [1]

The methods for designing e-textiles have stagnated. Current methods of designing e-textiles involve paper sketches complemented by computer-controlled weaving machines. Instructions are entered into the weaving machine by images created using Photoshop or similar software from the paper sketches. The image is then converted into a file readable by the weaving machine's software, after which it is entered into said software. Simple revisions and changes to the design require reediting of the image which is time consuming, unintuitive and prone to small errors.

This thesis is a research and seeks to solve the need for a useful and feature rich tool for designing textiles with the particular emphasis on features that aid the design of e-textiles. The way this is achieved is by designing and creating such an application. Achieving this goal could ease the work of designers and allow them to focus on more important parts of designing. It could also ease the work of new designers and amateurs. The first research topic of this thesis is web-based applications. The research looks at the tools for creating web applications, the pros and cons of these tools and which ones best suit the type of application created in this thesis. The second topic is user experience of computer software. This means to study how humans perceive computer programs and what features are intuitive. This does not include studying the human psyche, but to create a user-friendly program that has enough complexity to aid even in more complex tasks. Lastly the research looks at how the tools used and the user experience can be applied to aid textiles design.

The research adheres to the following restrictions. It includes the basics of weaving and textile design as they are important building blocks for the fundamental features of the application. For example, how the pixel-based textile designing works and how it is translated into the finished fabric. However, extensive research of specific weaving techniques are not included as these are unimportant to the basic features of the application. The application has the basic tools to create weaves but does not expand the user's knowledge of weaving techniques. In short, the goal is not to create a learning tool.

The application is built step by step and the progress is reported in section 3. The first version is a rudimentary template where the objective is to create the foundation, onto which the tool can be expanded. The later versions add functionality and have user experience as the main objective. Ideas for features in the application are discussed with the e-textile designer Emmi Pouta from Aalto university. Ideas for features include pixel based and three dimensional (3D) viewing and editing of the fabric, presets for ease of use and easy conversion between fabrics of different layers. The features are expanded upon in section 4.

2 Background

2.1 Weaving

Fabrics are woven by interlacing two different yarns, called warp and weft, with each other at 90-degree angles as is illustrated in figure 1. Warps represent the vertical yarns and weaves the horizontal yarns. [2] Weaving machines (looms) hold the warps inside small crevices called heddles, that can be lifted or lowered. Older wooden looms, or shaft looms, have varying amounts heddles and often require multiple warps to be threaded through the same heddle, which reduces the available complexity of the fabric. [3, 4] Modern power looms have up to 10000 heddles which allows for greater complexity.

The weaving process can be divided into three parts that are repeated in succession. First is shedding, where some warps are raised and some are lowered. The lifted warps are determined by the person weaving or by a file submitted to a software in modern power looms. After this is picking, where the weft is carried through in between the warps that are lifted and the ones that are down. Lastly is beating-up, where the weft is pushed back against the fabric to make is tight. [3, 4] The pattern that the interlaced warp and weft create is called a weave. The simplest weave, a plain weave, is achieved when for every row, every other warp is lifted. This gives us a sort of chess pattern. [2, 4]

In 1804 Joseph Marie Jacquard invented the Jacquard machine, which can be fitted on to a power loom to simplify the process of weaving. Unlike the older shaft looms where multiple warps were threaded through the same heddles, the Jacquard machine has a heddle for each warp. The first Jacquard machines were controlled by cards with holes punched into them. Each row on the cards was a row in the fabric and the holes marked which warps should be lifted for that row. [5] Modern Jacquard power looms are controlled by programs that read tagged image file format (TIFF) bitmaps with black and white pixels. The pixels tell the weaving machine row by row which warps should be lifted. If a pixel is black, the warp should be lifted and vice versa. These files can be created by photoshop or by industry software. [6]

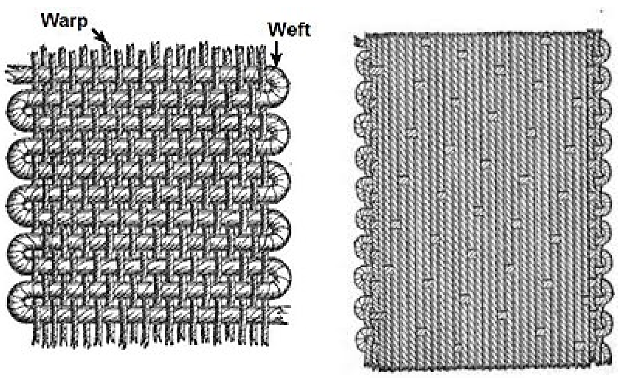

Figure 1: Illustration of basic method of weaving with the warp and the weft being interlaced with each other.

2.2 Multilayered weaving

E-textiles often employ layered weaving to create complexity and protection to the conductive threads. The basis of layered fabrics is that the warps and wefts are divided into layers and stay separate from the other layers unless overlap is needed. For example if a plain weave as a single layer fabric is converted into two layers, as can be seen in figure 2, we have to take every other warp and assign them to the top fabric and every other to the bottom fabric. Then we first look at the points where the top warps meet the top wefts. In the case of a two layered fabric this includes four points. The plain weave is then translated onto these four points. After that the pattern is translated onto the points where the bottom warps meet the bottom wefts. This results in a diagonal line from the bottom left corner to the top right corner. When the loom is working on a top layer, the warps of the bottom layer must be kept down in order to separate them. The same is true for the bottom layer, the warps of the top layers must be lifted. This means black pixels must be added to the points where the top warps meet the bottom wefts. The result is the weave seen in figure 2. In this example a 2x2 weave becomes a 4x4 weave. The same means can be used for three layers and above. Complexity to the weaving arises when the weave employs varying amounts of layers in different parts of the fabric. Designing this type of weave with pen and paper requires unreasonable amounts of resources.

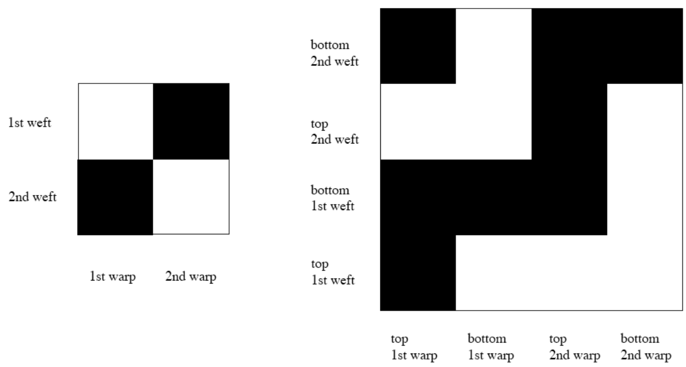

Figure 2: Single layered plain weave is converted into two layers.

2.3 WebGL and three.js

Web graphics library (WebGL) is an application programming interface (API) for JavaScript. It allows for the rendering of two and three dimensional (2D and 3D) objects inside a web browser. [7,8] It was released in 2011 and was soon integrated into Chrome and other web browsers natively. [9] WebGL was built off of open graphics library for embedded systems (OpenGL ES) which is a 3D rendering software. Embedded systems refers to the fact that it was designed for embedded systems like smartphones and tablets. WebGL is the browser based version of OpenGL ES, and as such, allows for hardware-accelerated graphics. [7, 8] This means that WebGL uses the graphics processing unit (GPU) of the computer and allows the removal of dependency on a server's calculation capacity. Over all it allows graphics calculation of greater intensity inside a web browser. WebGL rendering is controlled by JavaScript and shading is controlled by OpenGL shading language (GLSL). GLSL is what allows WebGL to use a computers GPU. On top of WebGL, the applications described in this thesis uses a library called three.js. It is an open source library that provides high level, developer friendly, functions for WebGL. [7,8,10] The application uses WebGL 1.0 instead of WebGL 2.0 since it is more widely supported by web browsers and GPUs. The support for WebGL 2.0 in three.js is also varying.

3 First Version

3.1 Design

The objective of the first version of the application is to be a base onto which more features can be added. It has a Hypertext markup language (HTML) base and a WebGL canvas. Different features are added to the space surrounding the WebGL canvas. Half of the canvas is used for a pixel-based editor. The editor has a two-dimensional grid of squares that can be toggled on and off. The contents of the editor are then be converted into a matrix that is then be displayed on the other half of the canvas. The other half features a 3D model of the fabric as it is determined in the editor. The model on the 3D-viewer is can be moved, rotated and zoomed in on. Likewise, the editor can be moved and zoomed in on. After the user has achieved the desired weave, it can be downloaded as a TIFF file and run on power looms. The power loom used in Aalto University is a digital Jacquard loom Thread Controller 2 (TC2). This application is be optimized to be used with the TC2 loom [6].

3.2 Implementation

The application employs two HTML5 canvases that are converted into WebGL canvases that this paper calls views. The views are pixel view and 3D view They both have their own class that contains their methods and parameters. Three.js is used as a basic tool for creating the Scenes, Renderers and Cameras. A scene is a object that hold the positions of all the objects added to it and is rendered to the screen every frame. Renderer is an object that takes the scene and the camera object and adds them together to create a view of the scene through the camera and renders the view on the screen. Camera is the viewport into the scene. A multidimensional JavaScript array is used to create a matrix to store the pixels. A one represents a black pixel and a zero a white pixel in the TIFF file. The two canvases share the same matrix so if it is edited, the results are shown in both views. User controls provided by three.js called OrbitControls are added to both view with different settings. In the pixel view, rotating is not allowed and panning is bound to the right mouse button. In the 3D view rotating is bound to the right mouse button and panning to the left mouse button. In both views zooming is bound to the mouse wheel.

The pixel view is a two dimensional rectangle with black and white pixels representing the crossings of the warps and wefts. The pixels can be turned on and off by mouse click. The type of camera used for the pixel view is an orthographic camera, where the camera does not have a field of view. The viewing window is the canvas itself which means there is no depth. Orthogonal camera is useful with two dimensional uses where sense of depth would be out of place. Figure 3 illustrates the difference between the orthographic camera and a camera with a field of view i.e. a perspective camera.

The 3D view is a representation of what the resulting fabric looks like. It uses TubeGeometry granted by three.js, which is a geometry object that has a radius and follows a path specified to it. Using segments of TubeGeometries the view builds the warps that arc up or down depending on whether the corresponding pixel is black or white. Up if the pixel is black and down if the pixel is white. After the warps are generated, the wefts are added as straight TubeGeometries perpendicular to the warps. In order to enhance the effect of the 3D tubes, the application adds a HemisphereLight to the scene. The light is placed above the TubeGeometries and provides an even light onto the top surface of the tubes. This helps differentiate the same color tubes from each other.

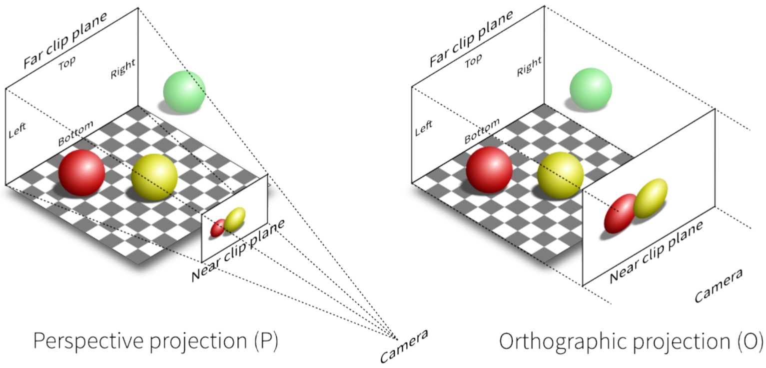

Figure 3: The difference between a perspective camera and an orthographic camera.

4 Second Version

4.1 Design

The second version of the application supports multilayered weaving. The user should be able to specify how many layers the weave is divided into and then interact with the program as before. If the user wants to create a new weave from scratch, the application should have options for configuring a base from which the user can begin designing. Possible options are: how many layers, what type of weave or size of canvas. After entering the desired options the program creates a canvas with the options and the user can begin working.

4.2 Development

5 User test

6 Summary

References

[1] K. Flood, "The original creators: Diana dew," *VICE Media LLC*, 2011. [Online]. Available: https://web.archive.org/web/20111219075111/http://www.thecreatorsproject.com/blog/the-original-creators-diana-dew

[2] J. Shenton and E. Ridsdale, *Woven Textile Design*. London, England: Laurence King Publishing, 2014, ISBN: 9781780675695. [Online]. Available: web.a.ebscohost.com.libproxy.aalto.fi/ehost/ebookviewer/ebook/bmxlYmtfXzkyNjE5NF9fQU41?sid=efe5ee37-0b30-4d6a-b8d8-ff9caa91e5ad@sdc-v-sessmgr01&vid=0&format=EB&lpid=lp_34&rid=0

[3] A. M. Collier, *A Handbook of Textiles*, 3rd ed. Oxford, England: Wheaton, 1980, ISBN: 9780080249742.

[4] W. H. Dooley, *Textiles*. Boston, USA: D.C. Heath and Co, 2007. [Online]. Available: www.gutenberg.org/files/24077/24077-h/24077-h.htm

[5] J. Essinger, *Jacquard's web : how a hand-loom led to the birth of the information age*. Oxford, England: Oxford University Press, 2007. [Online]. Available: https://ebookcentral-proquest-com.libproxy.aalto.fi/lib/aalto-ebooks/detail.action?docID=422876

[6] "Tc2 digital jacquard loom." [Online]. Available: https://www.digitalweaving.no/tc2-loom/

[7] D. Cantor and B. Jones, *WebGL Beginner's Guide*. Packt Publishing, 2012. [Online]. Available: https://www.oreilly.com/library/view/webgl-beginners-guide/9781849691727

[8] T. Parisi, *WebGL: Up and Running*. O'Reilly Media, 2012, ISBN: 9781449323578. [Online]. Available: https://www.oreilly.com/library/view/webgl-beginners-guide/9781849691727

[9] E. Kay and A. Boodman, "A dash of speed, 3d and apps," 2011. [Online]. Available: web.archive.org/web/20150411184902/http://chrome.blogspot.co.uk/2011/02/dash-of-speed-3d-and-apps.html

[10] "Three.js library." [Online]. Available: https://threejs.org/docs/index.html#manual/en/introduction/Creating-a-scene