FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

FEUP

Hyperbolic Tree Visualization on Mobile Devices

Luis António Alves Ferreira

Report of Project

Master in Informatics and Computing Engineering

Supervisor: Ademar Aguiar (PhD.)

29th June, 2009

Hyperbolic Tree Visualization on Mobile Devices

Luis António Alves Ferreira

Report of Project
Master in Informatics and Computing Engineering

Approved in oral examination by the committee:

Chair: Ana Paula Cunha da Rocha (Auxiliar Professor)

External Examiner: Carlos Miguel Ferraz Baquero Moreno (Auxiliar Professor)
Internal Examiner: Ademar Manuel Teixeira de Aguiar (Auxiliar Professor)

29th June, 2009

Abstract

The use of mobile devices such as *smartphones* is increasing exponentially as they become cheaper and more flexible to use. The communication technologies are now widespread throughout the developed countries. People don't need to be seated in front of a desk to work anymore. The desktop paradigm is rapidly becoming a thing of the past. Nowadays with services like *Flickr* or *Facebook* people do not need a computer to store their pictures or talk with friends. The information is spread all over the internet. The reality is that people are now more connected in every possible meaning.

The content available on the internet is reaching dimensions that wouldn't be imagined a few years ago. Twenty years ago, during the analog era, people were used to take a couple of photos from their vacations. Today they and their friends take hundreds of photos. However, despite the increasing number of content the human capacity remains the same. People still have limits regarding the number of memories they can process and keep.

The goal of the project is to focus on new ways of enhancing the browsing experience by filtering and revealing implicit relations between content. The working domain is media content which is available on the internet. The medium are mobile devices. The development of new visualization and interaction methods on this kind of small computers must be the next step. More often, people have a busy life which means they don't want to waste time dealing with technology. They expect to find the information they want as fast as possible in the most simple way possible.

The developed solution is a visualization software module that can be integrated within a mobile application as a browsing tool. It uses the hyperbolic tree visualization technique as the main starting point. It is aimed for hierarchical information such as graphs and is capable of giving both context and focus views. It employes a fish-eye distortion over a set of nodes, showing nodes in different sizes depending on their selected level of detail. This technique is known for an optimal ratio between content displayed and screen area usage. This is specially useful on mobile devices which have very limited screen sizes.

The platform of choice for the prototype development was the iPhone. The visualization module was integrated with two applications for the iPhone. *TinDI* is used to upload photos and videos to a private storage service. *KleinerExpresso* is a pure visualization tool of services such as *Flickr*, *Facebook*, *Picasa* and *Twitter*. In both applications the goal is to prove that the developed visualization is useful and innovative in the mobile market.

The prototypes showed are early examples of what might become a commercial product. The priority was the visualization and graphics development. They were not taken under exhaustive usability or performance tests. The overall feedback from empirical test

was mainly regarding interaction issues, meaning that additional work needs to be done in this fields.

The concept of using the hyperbolic tree to explore and visualize media content on mobile devices leads to other techniques that also apply distortion to content mapping. This work laid out a new way to design and develop visualizations and hopefully many more will follow the same principle.

keywords: information visualization, mobile, media, network, browsing, hyperbolic tree, interaction

Resumo

A utilização de dispositivos móveis, tal como *smartphones*, está a aumentar exponencialmente à medida que estes diminuem de preço e se tornam mais fáceis de utilizar. As tecnologias de comunicação estão agora espalhadas nos países desenvolvidos, o que significa que as pessoas não precisam de estar sentadas em frente a uma secretária para trabalhar. O paradigma do *desktop* está rapidamente a transformar-se numa coisa do passado. Hoje em dia, com serviços de partilha de fotos como o *flickr* ou *facebook*, não é preciso um computador para a aceder à informação ou comunicar. Na realidade as pessoas estão hoje mais ligadas do que no passado.

O conteúdo disponível na internet está a atingir dimensões que nunca se imaginaram há alguns anos atrás. Há 20 anos atrás, durante a era do analógico, as pessoas estavam habituadas a tirar apenas algumas fotos durante as férias. Hoje é possível tirar centenas de fotos. Por consequência, a quantidade de conteúdo está a aumentar. No entanto, os humanos continuam iguais. As pessoas continuam a ter limites em relação ao número de memórias que conseguem manter e processar.

O objectivo do projecto aqui descrito é desenvolver novas técnicas que melhorem a experiência de navegação através de filtragem ou revelando relações implícitas entre conteúdo. O domínio de trabalho é o conteúdo multimedia disponível na internet sendo que o meio de utilização são os dispositivos móveis. O desenvolvimento de novos métodos de visualização e interacção para este tipo de micro computadores é o próximo passo. Cada vez mais as pessoas têm uma vida ocupada e como consequência não querem desperdiçar tempo com detalhes de tecnologia. Elas esperam encontrar a informação que pretendem de forma rápida e de forma simples.

A solução apresentada é um módulo de software de visualização que pode ser integrado em aplicações móveis como ferramenta de navegação. A técnica de visualização hiperbólica é usada como principal ponto de partida. Esta é orientada para informação hierárquica tal como grafos, permitindo vistas ao mesmo tempo sobre o detalhe e contexto do conteúdo. Isto é possível usando distorções do tipo *fisheye* sobre o conjunto de nós. A vantagem da utilização desta técnica é o rácio óptimo entre conteúdo visualizado em relação à área de ecrã utilizada. Esta característica é especialmente útil em dispositivos móveis que têm tamanhos de ecrã muito reduzidos.

A plataform de desenvolvimento escolhida para desenvolver o protótipo foi o *iPhone*. O módulo de visualização foi integrado em duas aplicações para o *iPhone*. A aplicação *TinDI* é usada para enviar fotografias e videos tirados no dispositivo para um serviço de alojamento privado. A aplicação *KleinerExpresso* é uma ferramenta de visualização pura para serviços como o *Flickr*, *Facebook*, *Picasa* e *Twitter*. Em ambas o objectivo é provar que a visualização implementada é útil e inovadora no mercado móvel.

Os protótipos mostrados são exemplos numa fase inicial de desenvolvimento e poderão

a tornarem-se em produtos comerciais. A prioridade de desenvolvimento foi a visualização e o desenvolvimento gráfico. Os protótipos não foram sujeitos a testes de usabilidade ou performance exaustivos. O feedback obtido através de testes empíricos realizados por especialistas em interfaces gráficas indica que é necessário explorar melhor a área de interacção com a árvore hiperbólica.

O conceito de utilizar a árvore hiperbólica para explorar e visualizar conteúdo em dispositivos móveis leva à utilização de outras técnicas que também usam distorção. Este trabalho definiu um conjuto de preceitos fundamentais para o desenho e desenvolvimento de visualização dinâmica de informação no contexto móvel.

palavras-chave: visualização de informação, móvel, multimedia, grafos, navegação, árvore hiperbólica, interacção.

Acknowledgements

In completing this project, there are many people to whom I am extremely grateful. I am a privileged person because I have them in my life.

Thanks to...

> Fraunhofer Portugal amazing team who welcomed me, gave support to all my activities and allowed me the freedom to do this work. Fraunhofer has been an incredible space to learn, play and explore.
>
> Luis Teixeira who oriented me. He was a very comprehensive person, always keeping me in the right direction.
>
> Elizabeth Carvalho who gave the moto for the thesis project. Her expertise was a precious help during the first month of work. She gave me important insights about information visualization and mobile development.
>
> My parents who never complained about my decisions. They give me the emotional equilibrium I need to perform my work.
>
> My sister who has an amazing capacity to think sideways and interesting views about the human and philosophical side of technology. Thanks for everything.

I would like to dedicate this work to...

> My cousin André, a true Warrior.
>
> My grandfather, who always believed me.

"Vision driven by art survives beyond our lifespan"

— Hiroshi Ishii

Contents

1 Introduction **1**
1.1 Domain . . . 1
1.2 Project Context . . . 2
1.3 Summary of Contributions . . . 2
1.4 Thesis Overview . . . 3

2 Background **5**
2.1 The User . . . 5
2.1.1 Memory . . . 6
2.1.1.1 Attention . . . 7
2.1.1.2 Long-Term Memory . . . 8
2.1.2 Perceptual Processing . . . 8
2.1.2.1 Parallel Processing . . . 8
2.1.2.2 Pattern Perception . . . 9
2.1.2.3 Sequential Goal-Directed Processing . . . 9
2.1.3 Interaction Design . . . 9
2.1.3.1 Where does it fits in? . . . 10
2.1.3.2 The Process . . . 11
2.2 Mobile Devices . . . 12
2.2.1 Visualizing Information on Mobile Devices . . . 12
2.2.1.1 Mobile versus Desktop Visualization . . . 12
2.2.1.2 Mobile Context . . . 13
2.2.1.3 Mobile Visualization Design . . . 13
2.2.1.4 Content Presentation . . . 14
2.2.1.5 User Input and Interaction . . . 14
2.2.2 Everywhere Computing . . . 14
2.2.3 Market Overview . . . 15
2.3 Information Visualization . . . 16
2.3.1 Data Analysis . . . 17
2.3.1.1 Entities . . . 18
2.3.1.2 Relationships . . . 18
2.3.2 Hyperbolic Tree . . . 18
2.3.2.1 Hyperbolic Space . . . 19
2.3.2.2 Mapping and Representation . . . 19
2.3.2.3 Change of focus . . . 20
2.4 Summary . . . 21

3 Related Work **23**
3.1 Exploring Networks . . . 23
3.1.1 Graphic Design . . . 24
3.1.1.1 Data Visualization of a Social Network . . . 24
3.1.1.2 ClusterBall . . . 24
3.1.2 Internet Mashups . . . 25
3.1.2.1 Media Networks . . . 25
3.1.2.2 Interactive Arcs . . . 25
3.2 Hierarchical Information Visualization . . . 27
3.2.1 Non-Distortion based Visualizations . . . 27
3.2.2 Generalized Distortion-based Visualizations . . . 27
3.2.3 Hyperbolic Tree Visualizations . . . 29
3.3 Summary . . . 29

4 Methodology **31**
4.1 Problem Analysis . . . 31
4.2 Problem Specification . . . 32
4.2.1 Requirements . . . 33
4.2.1.1 Usage Scenarios . . . 33
4.2.1.2 Functional Requirements . . . 33
4.2.1.3 Non-functional requirements . . . 34
4.2.2 Visualization Process . . . 34
4.2.3 Interface Design . . . 35
4.3 Technological Decisions . . . 35
4.3.1 iPhone . . . 36
4.3.2 OpenGL ES . . . 36
4.3.3 Data Storage . . . 36
4.4 Design and Implementation of the Visualization Module . . . 36
4.4.1 Overview . . . 37
4.4.2 Services . . . 38
4.4.3 Data Views . . . 39
4.4.4 Content Management . . . 40
4.4.5 Hyperbolic Tree Visualization . . . 40
4.4.5.1 Layout . . . 40
4.4.5.2 Picking . . . 41
4.4.5.3 Change of focus . . . 41
4.5 Applications . . . 42
4.5.1 Kleiner Expresso . . . 42
4.5.2 TinDI . . . 42
4.6 Summary . . . 43

5 Discussion and Analysis **45**
5.1 Evaluation . . . 45
5.1.1 Information Visualization Process . . . 45
5.1.1.1 Criteria . . . 45
5.1.1.2 Techniques . . . 46
5.1.1.3 Metrics . . . 46

5.1.1.4 Process 46
5.1.2 Empirical Results 47
5.2 Successes, Challenges and Pitfalls 48
5.3 Summary 49

6 Conclusion **51**
6.1 Future Work 51
6.1.1 Implementation 51
6.1.2 Design 52
6.2 The Future of Mobile Devices 53
6.3 Summary 53

References **57**

A Thesis Proposal: Initial Presentation **59**

B Technological Experiments **61**
B.1 Whiterose 61
B.2 Gunpowder 61

C Visualization Module Implementation Details **63**
C.1 Class Diagram 63

D Suplementary Sketches **65**

CONTENTS

List of Figures

2.1 Stroop Effect 7
2.2 Interaction design context [Mog06] 10
2.3 Interaction design process [Mog06] 11
2.4 Context-aware computing scenarios [BNSW94] 15
2.5 The 7 steps of the information visualization process 17
2.6 Changing the focus on the hyperbolic tree [LRP95] 20

3.1 One of Lombardi's "narrative structures", and a detail of the piece 23
3.2 Felix Heinen's social network graphic visualizations 24
3.3 ClusterBall: Humans, Medicine, History 25
3.4 Tuneglue and FLickrGraph 26
3.5 Examples of interactive arcs: del.icio.us.discover and Digg Arc 26
3.6 RadialMobile screenshot 28
3.7 Perspective Wall, ConeTree and TreeMap 28

4.1 Problem overview 33
4.2 Visualization steps involved in the problem specification 34
4.3 Initial interface mockup 35
4.4 Architecture overview of the visualization module 37
4.5 Main architecture module interaction 37
4.6 Class diagram of the service related classes 38
4.7 Class diagram of the data view related classes 39
4.8 Example of a Data View for liked content 40
4.9 Class diagram of the content management related classes 40
4.10 Screenshots of the kleiner expresso application 42
4.11 Screenshots of the TinDI application 42

B.1 Screenshot of the whiterose opengl ES experiment 61
B.2 Screenshot of the gunpowder core data experiment 61

C.1 Complete class diagram of the visualization module 63

D.1 Original sketch for the initial solution 65
D.2 Sketch for the data management solution 66

LIST OF FIGURES

List of Tables

2.1 Comparison of mobile devices [YC06] 16

LIST OF TABLES

Definitions and Abbreviations

CPU Central Processing Unit

FOVEA Part of the eye located in the center of the retina

GPU Graphics Processing Unit

H2 Two-dimensional Hyperbolic Space

HCI Human-Computer Interaction

HTTP Hypertext Transport Protocol

ICT Information and Communication Technologies

MVC Model View Controller

PDA Personal Digital Assistant

UFOV Useful Field of View

UI User Interface

URL Universal Resource Locator

InfoVis Information Visualization

Smartphone A smartphone is a mobile phone offering advanced capabilities, often with PC-like functionality

Chapter 1

Introduction

Mobility is the kind of word we hear everyday and read everywhere. It has rapidly changed from a trend in the past to our everyday reality. Mobility encompasses a broad range of words and concepts. In the scope this thesis it means ubiquitous computing. The growing need for connectedness in our lives fostered the development of mobile devices. On the other hand, according to Metcalfe's law [Met95] the increasing number of people connected led to the exponential growth of possible relationships within the network. Consequently, the content users can access from the tip of their fingers increased in the same proportion.

Nowadays, we, the users have huge amounts of content that belong to us in a certain way, but we live a stressful life and so we expect technology to bring the meaningful content close to us. It is a reality that we want to access our content whenever we want, but it is also true that mobile devices have a completely different set of characteristics compared to typical desktop or laptop computers. This means we can't just import the old User Interface (UI) paradigms and apply them to mobile devices. We need to find new ways that take advantage of this specific type of small scale factor computer.

The use of Hyperbolic Trees to visualize and browse content is not new, but has been underrated during the last couple of years. Combining it with other techniques it will be possible to view simultaneously local detail and global context of the information being analyzed. Following that idea, the main focus of this thesis is to enhance the browsing experience of the user on the "move".

1.1 Domain

In Information Visualization (infovis), every problem is unique. For a specific problem there is a specific data set, constraints and usage scenarios that should be taken into account in order to develop the visualization solution that best suits it. This particular thesis focus on media content stored and uploaded through services such as Flickr [Fli] and

Facebook [Fac]. The people that own the content and their "friends" are also part of the scope.

The difficulty in selecting a specific problem comes from the fact that there is no strong solution for two different problems at the same time. There are different kinds of content to be seen using mobile devices. However, when it comes to life the things that matter people are life memories – photos, videos and their friends. This is the kind of domain with more potential to be explored. There is a strong network of relations to explore between content published in this kind of services with a social component.

1.2 Project Context

The project has been developed at Fraunhofer AICOS (Assistive Information and Communication Solutions) in Portugal. Fraunhofer is a German research organization with several research institutes each focusing on different fields of *applied* science. The core purpose of AICOS is to democratize access to Information and Communication Technologies (ICT) through the development of market-oriented, commercially successful, applied R&D assistive solutions.

The main target of the project is a group of consumers who actively use information technologies in their everyday lives. It is part of a set of projects oriented around mobile platforms and content, that try to reinvent the current usage for existing services.

1.3 Summary of Contributions

This chapter exposes the need for an infovis module designed taking into account mobile devices specific needs and being able to handle dynamically changing information in a flexible manner. The work described in this thesis makes two primary contributions in this area.

Life Streaming - Bring personal media content to a mobile application. Specifically it streamlines all the content, despite the source, and depict hidden connections exploring metadata attributes.

Dynamic Mobile Visualization - The implementation and redesign of the hyperbolic tree visualization to meet mobile devices specific characteristics. The tree is fed in a dynamic way, changing according to filters and whenever new content is available.

1.4 Thesis Overview

The thesis document is organized in a way that give the unfamiliar reader an insight about the infovis research field, related works, the methodology followed to design and implement the software and finally a discussion and conclusions about the development process. The structure is the following:

Chapter 2 - Background is a theoretical overview that explains concepts about interaction design, visualization and interface design for mobile devices and existing visualization techniques.

Chapter 3 - Related Work is about projects and research done either in the infovis or mobile device field that explore hyperbolic trees, media content or networks build upon available information on the internet.

Chapter 4 - Methodology describes the main development steps that took the initial problem to the final applications for mobile devices.

Chapter 5 - Analysis and Discussion of the evaluation steps, needed to verify the usefulness and usability issues of the developed software.

Chapter 6 - Conclusion of the project, goals and expectations versus final result comparison.

INTRODUCTION

Chapter 2

Background

During the last century *vision* has been heavily discussed among the scientific and philosophical community. Despite the current thinking of vision as always the same and identical, it should be distinguished among different modalities and conceptions, among different visual practices and visual regimes [Sha03]. When compared to other senses, vision is a recent development of the human specie, as proposed by Freud:

> *[. . .] the priority of vision was a relatively late development in humans, one that had to do with the assumption of an erect posture that led to the abjection of smell and touch.* [FG89]

On this basis, there's a common belief that the advantage gained in the instrumental manipulation of the environment is at least matched by the alienation from our own bodies and their immediate circumstances to an environment based upon visual abstractions. Our vision is coupled to a series of complex systems such as memory, cultural consciousness, thinking and finally our eye system. Seamlessly, visualization is a complex field with several subjects to be taken into account.

This chapter is focused on a series of subjects that are directly related with information visualization. They give the reader an insight about the concepts referred along the thesis.

2.1 The User

The user is the central piece of interaction and visualization design. In order to design better interfaces and ways of mediating the computer-human relation, researchers must understand how the user's memory, attention and perception works. This section describes briefly some parts of the human systems that are important for the development of the project.

2.1.1 Memory

Active cognition has two main driving forces that work side by side: attention and memory. Despite the fact that attention is the motor, memory provides the framework [War04]. There are three types of memory:

Iconic memory is a very brief image store, holding what is on the retina until it is replaced by something else or until several hundred milliseconds have passed [Spe60]

Long-term memory is the information that we retain from everyday experience, perhaps for a lifetime. It happens as a result of active processing that integrates new information into existing knowledge [CL72].

Visual working memory holds the visual objects of immediate attention. These can be either external or mental images. In computer science terms, this is a register that holds information for the current operations of visual cognition [CL72].

From these types of memory, *Visual working memory* is the most critical cognitive resource for visual thinking. Some of its key properties are [Irw92, Ren00, Mar82]:

- visual working memory is separate from verbal working memory;
- capacity is limited to a small number of simple visual objects and patterns, varying from three to five;
- positions of objects are stored in a egocentric map. Only three to five are linked to specific objects;
- attention controls what visual information is held and stored;
- the time to change attention is about 100 milliseconds;
- the semantic meaning of an object or scene (related more to verbal working memory) can be activated in about 100 milliseconds;
- for items to be processed into long-term memory, deeper semantic coding is needed.

Visual working memory can be roughly defined as the visual information retained from one fixation to the next. Besides position, abstract shape, color and texture information is also retained. The number of objects retained depend on the task and the kind of pattern. One of the consequences of the very small amount of information held in visual working memory is a phenomenon known as *change blindness* [Ren00]. Because we remember so little, it is possible to make large changes in a display between one view and the next and the user generally will not notice, unless the change is to something they have recently attended to it.

To many people, the extreme limitation on the capacity of visual working memory seems quite incredible. How is it possible that we have a rich and detailed representation of the world with such a shallow internal representation. We perceive the world to be rich and detailed, not because we have an internal detailed model, but simply because whenever we wish to see detail we can get it. The detail is obtained by focusing attention on some aspect of visual image at the current fixation or by moving our eyes to see the detail on some other part of the visual field [O'R92].

2.1.1.1 Attention

Although we are blind to many changes in our environment, some visual events are more likely to cause us to change attention that others. The degree of blindness changes according to how big or small is the visual event. Generally, patterns larger than one degree of visual angle appearing near the point of fixation are always noticed.

PURPLE YELLOW RED BLACK BLUE RED GREEN

Figure 2.1: Stroop Effect

The selectivity of attention works efficiently. Even though we may wish to focus on one aspect of a display, other information is also processed, to a high level. For example, based on the Stroop effect test [Str33] in a set of words printed in different colors if the words themselves are color names that do not match the ink colors, subjects name the colors more slowly than if the colors match the words (illustrated in figure 2.1). This means that the words are processed automatically; we cannot entirely ignore them even when we want to. More generally, it is an indication that all highly learned symbols will automatically invoke verbal-propositional information that has become associated with them.

As we work in front of a computer, there is an increasing need to attract the user's attention to a specific problem or issue. These *user interrupts* should have four basic visual requirements in order to be efficient:

- a signal should be easily perceived, even if it is outside of the are of immediate focal attention;
- if the user wishes to ignore the signal and attend to another task, the signal should continue to act as a reminder;
- the signal should not be so irritating that it makes the computer unpleasant to use;
- it should be possible to endow the signal with various levels of urgency.

2.1.1.2 Long-Term Memory

The long-term memory is more related with the verbal memory. Even thought it is not directly related with visual processing, it's understanding is important to perceive how visualizations can function as memory aids by rapidly activating structured non-visual information.

Long-term memory contains the information that we build up over lifetime. We tend to associate long-term memory with events we can consciously recall. However, long-term memory also includes motor skills, such as finger movements involved in typing and the perceptual skills, integral to our visual systems, that enable us to rapidly words and thousands of visual objects. Non-visual information that is not closely associated with concepts currently in verbal working memory can take minutes, hours or days to retrieve from long-term memory.

2.1.2 Perceptual Processing

The human visual perception has three main stages

1. Parallel Processing - to extract low-level properties of the visual scene
2. Pattern Perception - active processes of pattern perception pull out structures and segment the visual scene with different colors, texture and motion patterns
3. Sequential goal-directed processing

2.1.2.1 Parallel Processing

Visual information is first processed by large arrays of neurons in the eye and in the primary visual cortex at the back of the brain. Individual neurons are selectively tuned to certain kinds of information, such as the orientation of the edges or the color of a patch of light. In this stage processing, billions of neurons work in parallel, extracting features from every part of the visual field simultaneously. This parallel processing methods proceeds whether we like it or not, and it is largely independent of what we choose to attend to. It is also rapid. If we want people to understand information quickly, we should present it in such a way that it could easily be detected by these large, fast computational systems in the brain. The main important of this stage are:

- Rapid parallel processing;
- Extraction of features, orientation, color, texture, and movement patterns;
- Transitory nature of information, which is briefly held in an iconic store;
- Bottom-up, data-driven model of processing;

2.1.2.2 Pattern Perception

At the second stage, rapid active processes divide the visual field into regions and simple patterns, such as continuous contours, regions of the same color, and regions of the same texture. Patterns of motion are also extremely important, although the use of motion as an information code is relatively neglected in visualization. The pattern-finding stage of visual processing is extremely flexible, influenced both by the massive amount of information available from the previous stage and by the top-down action of attention driven by visual queries.

The most important characteristics of this stage are:

- Slow serial processing;
- Involvement of both working memory and long-term memory;
- More emphasis on arbitrary aspects of symbols;
- In a state of flux, a combination of bottom-up feature processing and top-down attentional mechanisms;
- Different pathways for object recognition and visually guided motion.

2.1.2.3 Sequential Goal-Directed Processing

At the highest level of perception are the objects held in visual working memory by demands of active attention. In order to use an external visualization, we construct a sequence of visual queries that are answered through visual search strategies. At this level, only a few objects can be held at a time; they are constructed from the available patterns providing answers to the visual queries. For example, if we use a roadmap to look for a route, the visual query will trigger a search for the connected red contours (representing the roads) between two visual symbols (e.g. two connected xcities).

2.1.3 Interaction Design

The concept of *Design* differs from *Interaction Design*, as the first is a broader field and the later is focused on the user experience. From the several definitions found doing the research, here are one example for each:

Design. Design is all around us – either we control it – or it controls us [Oli89].

Interaction Design. The Design of the subjective and qualitative aspects of everything that is both digital and interactive, creating designs that are useful, desirable, and accessible [Mog06].

2.1.3.1 Where does it fits in?

The development disciplines related with interaction design are categorized according to 3 characteristics:

- Human or Technical
- Subjective or Objective
- Physical or Digital

If we discard the last item about physical or digital, we can group the disciplines into 4 kinds:

Design Disciplines: Industrial design, graphic design, web design and interaction design.

Human Sciences: Physical ergonomics and Human-Computer Interaction.

Engineering: Mechanical, Production, Hardware and Software engineering.

Technical Sciences: Physical and Computer sciences.

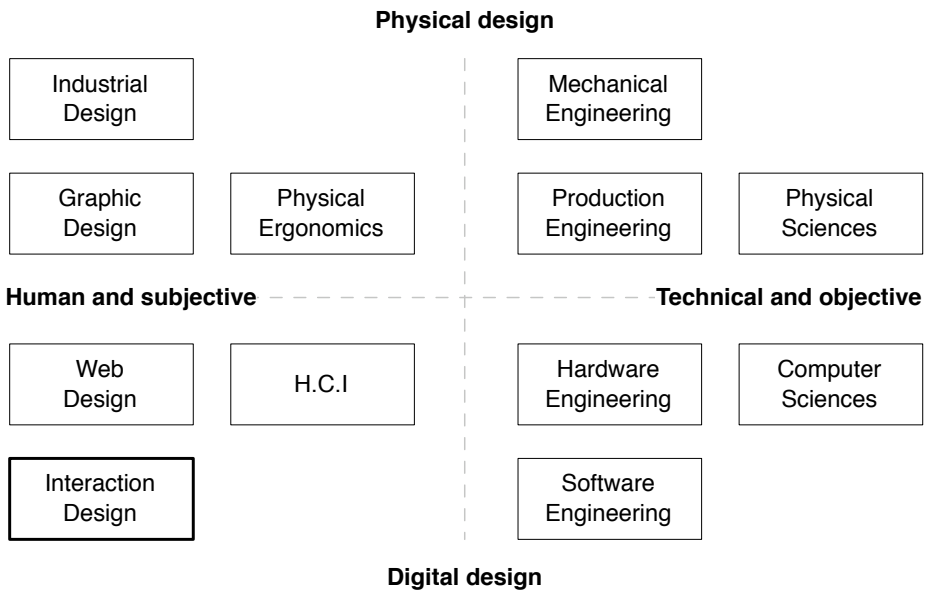


Figure 2.2: Interaction design context [Mog06]

As shown in figure 2.2, interaction design is a discipline in the digital domain with human and subjective qualities. It is closely related with Human-Computer Interaction (HCI) and graphic design.

2.1.3.2 The Process

Every interaction design process starts with an idea or a solution for an interaction problem. The designers decide about the ideal goal for the design. Then, they try to find metaphors, tasks and develop scenarios that map the initial motivation to the end goal. The process is about finding conceptual models with meaning that link the ideas and scenarios altogether. Finally they decide what kind of controls and displays should be used and how to arrange them [Mog06].

Figure 2.3: Interaction design process [Mog06]

The process of designing interactions can be summarized in this four steps (figure 2.3):

1. **Motivation:** Design ought to start from the problems that people are experimenting. The ideas are prompted by errors and faults that designers observe on daily life.

2. **Meaning:** An effective metaphor is the best way of giving meaning to design. It connects the abstract ideas to the real world scenarios.

3. **Models:** This is the conceptual cognitive science of understanding what the user doing a specific task needs to know. It plays an important role in the process. Designers need to have a clear picture of what users are thinking about in order to create the conceptual models.

4. **Mappings:** The user interaction is done via controls on the interface or in separate buttons. This step consists of mapping the visual representation on the display to a set of controls operated by the user.

2.2 Mobile Devices

2.2.1 Visualizing Information on Mobile Devices

Computer users have always relied on visual tools and techniques such as maps, diagrams and charts to better understand problems and solve them in less time. The continuous improvement of computers performance and graphics made it possible to implement a wide range of advanced visualization techniques to several different domains (business, medicine, criminal investigation, science). However, adapting these techniques to mobile devices such as PDAs (Personnal Digital Assistant) and Smartphones poses several technical and usage limitations. In order to harness the power of mobile visualizations an additional effort is needed to understand how to get the most from these tiny devices we carry everyday in our pockets [Chi06].

2.2.1.1 Mobile versus Desktop Visualization

Compared to mobile desktop computers, mobile devices have a set of restrictions that limit the development of applications [Chi06]:

- displays are smaller, have lower resolution and smaller color range;
- the width/height aspect ratio differs from desktop displays;
- onboard hardware such as CPU, GPU and memory are very limited;
- input devices such as tiny keyboards, micro-joysticks and scroll balls are inadequate from complex user tasks;
- connectivity is slower, affecting interactivity when a significant quantity of data is retrieved/stored remotely;
- form-factor, performance and input devices vary greatly;
- development tools and graphic libraries tend to be very limited;
- multitasking is usually not support among mobile applications;

Due to mobile devices characteristics, applications developed for desktop computers don't scale to mobile devices. Despite the current development on mobile devices, some of these limitations are unlikely to disappear as the devices must remain compact in size to be practical [YC06].

2.2.1.2 Mobile Context

Besides the existing limitation on current devices, the mobility context introduces further complications. The physical environment of a mobile device is extremely variable. A mobile device can be used in different lighting conditions, that range from extreme bright environment to complete darkness. These changes affect the color and graphics perception [Chi04].

The usage scenarios of mobile devices must also be addressed in a different way. When porting a desktop application to the mobile context, mobile devices usage must be taken into account. For example, a CAD system to design engines would face several performance and display limitations on a mobile device. In this case, the mobile application should be seen as a companion for the desktop application. Instead of porting the complete set of features, an acceptable example would be a schematic viewer or browser that could be used for maintenance operations.

It is also important to refer the external circumstances or activities that interfere with mobile users, making it difficult to focus attention on the device. A person has fewer cognitive and perceptional resources when driving a car or walking on the sidewalk compared to sitting in front of a desk or at home [Chi04]. Usually, in these cases the mobile task is secondary and done concurrently with other important tasks such as driving. This factor is very important while developing and testing mobile applications, as it can put the user in a life-threatening or dangerous situation.

2.2.1.3 Mobile Visualization Design

Mobile visualization design should address a set of steps in order to avoid situations such as misleading the user, the user getting lost when navigating using a map or presenting great looking and impressive graphics that are difficult to read or interpret.

The main steps are [Chi04]:

Mapping: the way information is visually encoded is crucial. Visualization turns numbers, objects, relations and string into graphics with several features – colors, lines, lengths, curvatures, animations. An effective mapping is consistently applied throughout the application and easy to interpret.

Selection: Due to display size limitation and mobile context already referred it is important to show only the essential information. Insufficient information will lead to suboptimal or wrong decision while excess of information may confuse the user or make it difficult to reason.

Presentation: Even with an efficient mapping it is important to think about how information is laid out on the display. Special attention must be taken to the useful screen area and if it is enough to display all the required information.

Interactivity: The way users interact with their information is crucial in order to engage his attention. The current devices, with touch capabilities, make it easier to develop new techniques that are very intuitive.

Human Factors: Users must be able to quickly recognize and easily interpret a visualization. The application interface should take into account human perceptual and cognitive capabilities.

Evaluation: Usability testing is the most crucial. This is the best way of doing user evaluation to test the application in *quasi-real* situations. Mobile application evaluation should be done with real devices (phone emulators can lead to unreliable results) and variable such as distraction should be measured.

2.2.1.4 Content Presentation

Content presentation on mobile devices depends on content type and problem domain. Content type ranges from text, picture, videos, maps, physical objects and abstract data. Depending on the problem, this content can be visualized in 2D or 3D mode, map like, with lists or any other custom technique.

2.2.1.5 User Input and Interaction

The existing mobile devices provide specific sensors and ways of interacting with the interface [AKK06]. The following list provide a briefing over the existing technologies:

Pen - An accessory similar to a pen is used as picking device on touch screen displays.

Multi-touch - The display is capable of detecting more than one simultaneous touch on the display surface.

Accelerometer - It provides de acceleration of the device, usually on tree axes *XYZ*, and can be used to determine the device orientation.

2.2.2 Everywhere Computing

The term "Everywhere Computing" is just a placeholder for a broad range of concepts that make possible the integration of computing power into everyday objects that are always connected or know where we are. This trend is being fostered thanks to new mobile technologies that make *Ubiquitous* and *Context-Aware* computing possible.

Ubiquitous computing [Pos09] is about small and networked devices distributed throughout everyday life (e.g. watch, clothing and domestic equipment such as the fridge). These devices are networked in order to be "aware" of their contents or context. A common scenario is the refrigerator that adjusts the temperature according from the reading sent

from the clothes sensors, or the watch that warns the user about spoiled food in the fridge. Besides the technological breakthrough, another challenge of this trend is related with interface and interaction design. It is important to keep the usage of this systems simple enough to make them "invisible" from the user's point of view.

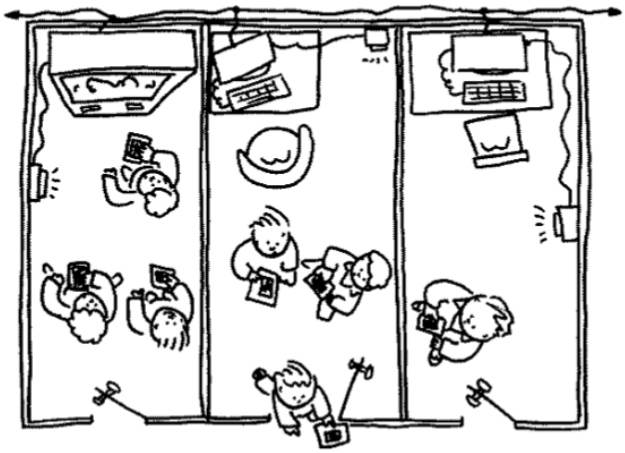


Figure 2.4: Context-aware computing scenarios [BNSW94]

One challenge of mobile distributed computing is to exploit the changing environment with a new class of applications that are aware of the context in which they are run. Such context-aware software [BNSW94] adapts according to the location of use, the collection of nearby people, hosts, and accessible devices, as well as to changes to such things over time. A system with these capabilities can examine the computing environment and react to changes to the environment. Three important aspects of context are: where you are, who you are with, and what resources are nearby (figure **??**). Context encompasses more than just the user's location, because other things of interest are also mobile and changing. Context includes lighting, noise level, network connectivity, communication costs, communication bandwidth, and even the social situation.

The growing number of mobile users also fostered the reinvention of certain services available on the internet [Pos09]. Webservices such as *Flickr*, *Facebook* and *Twitter* are now used everywhere and enable people to be "closer" to their friends. The content in this systems has also interesting properties that make possible to create new ways of visualize and combine the information into a single stream of content. This kind is called *lifestreaming applications* [Pos09].

2.2.3 Market Overview

A mobile device is any digital device that is small, easy to carry and designed to be used everywhere. The table 2.1 presents information about different categories of mobile devices.

Table 2.1: Comparison of mobile devices [YC06]

Features	PDA	Smartphone	Cellular Phone	Notebook	Consoles
Screen Size	240x320	320x480	160x160	800x480	480x272
Processing Power (vs. Desktop)	low	low	very low	medium	low
Memory	128MB	128MB	32MB	1GB	32MB
Networking	WI-FI, Bluetooth	3G, EDGE, WI-FI, Bluetooth	3G, Bluetooth	WI-FI, Bluetooth	WI-FI, Bluetooth
Input	pen, touch screen, buttons	pen, touch screen, buttons, keyboard, multi-touch	keyboard, buttons	keyboard	buttons, touch screen
Typical Usage	personal organization	communication, personal organization, entertainment	communication	travel laptop	entertainment
Popular Models	Palm TX, HP iPaq, iPod touch	iPhone, Palm Pre, HTC Dream	Nokia, Motorola	Asus Eee PC	Sony PSP, Nintento DS

2.3 Information Visualization

Visualization as a sub-field of science, statistics, and graphics has only been recognized as its own entity since the mid-to late-80s [Fry08]. The depth of the seminal work is in line with that of a young field, but finds its strength in background drawn from years of statistics and graphic design. However, there's a clear difference between visualization and information visualization [Fry08].

Visualization is concerned with non-abstract data sets, for example imagery from weather data or an animation describing the movement of a fluid. For this kind of data, representations or physical analogues already exist.

Information Visualization is concerned with making an abstract set of information visible, usually in circumstances where no metaphor exists in the physical world.

This thesis is focused on information visualization. The process of showing information has several steps (figure 2.5) that include data acquisition, parsing, filtering, representing and allowing interaction.

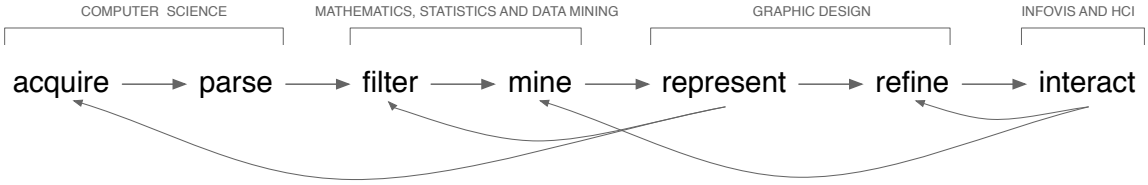


Figure 2.5: The 7 steps of the information visualization process

1. **Acquire** - the matter of obtaining the data, whether from a file on a disk or from a source over network.

2. **Parse** - providing some structure around what the data means, ordering it into categories.

3. **Filter** - removing all but the data of interest.

4. **Mine** - the application of methods from statistics or data mining, as a way to discern patterns or place the data in mathematical context.

5. **Represent** - determination of a simple representation, whether the data takes one of many shapes such as a bar graph, list or tree.

6. **Refine** - improvements to the basic representation to make it clearer and more visually engaging.

7. **Interact** - the addition of methods for manipulating the data or controlling what features are visible.

Part of the problem with the individual approaches of dealing with data is that the separation of the fields leads to each person solving an isolated part of the problem. Along the path towards a solution, something is lost at each transition.

2.3.1 Data Analysis

The classification of data is an important issue. It is closely related to the classification of knowledge. The following classification is informal and was brought here in order to give background to the methodology and related work chapter.

There are two fundamental forms of data: *data values* and *data structures* [Ber99]. A similar idea is to divide data into entities and relationships (often called *relations* along the thesis). Entities are the objects we wish to visualize; relation define the structures and patterns that relate entities to one another. Sometimes the relationships are provided explicitly; sometimes discovering relationships is the very purpose of visualization. We also can talk about the attributes of an entity or relationship. Thus, for example, an apple

can have color as one of its attributes. The concepts of entity, relationship, and attribute have a long history in database design and have been adopted more recently in systems modeling. However, we shall extend these concepts beyond the kinds of data that are traditionally stored in a relational database. In visualization it is necessary to deal wit entities that are more complex and we are also interested in seeing complex structure relationships – data structures – not captured by the entity relationship model.

2.3.1.1 Entities

Entities are the objects of interest. In the context of the problem an entity can be anything that needs to be represented as such. For example, People and Photos can be entities. Seamlessly a group of entities can also be considered an entity if it is convenient for the problem. For example, a group of photos – a collection – can be an entity.

2.3.1.2 Relationships

The structures that relate entities are called relationships. The type of relationship can be either physical or structural. The physical relationship defines how different physical parts related to each other (e.g. how different car parts are assembled together). The structural relationship is based on concepts (e.g a collection of digital photos contains one or more photos). Relationships can also be causal or temporal. A causal relationship causes another event and a temporal one defines an interval between two events.

2.3.2 Hyperbolic Tree

In the last few years, information visualization research has explored the application of interactive graphics and animation technology. The goal is to visualize and facilitate the comprehension about larger information sets that would otherwise be impossible to analyze or explore. One recurring theme has been the power of focus+content techniques, in which detailed views of particular parts of an information set are blended in some way with a view of the overall structure of the set. The hyperbolic tree, also known as hyperbolic browser, is such an example. It replaces the conventional approach of laying a tree out on a Eucledian plan by doing *layout* on the hyperbolic plane and then *mapping* to the unit disk. The *change of focus* is handled by performing an isometric transformation of the hyperbolic plane, moving the laid out tree in the process.

The focus of this section is explain the general concepts around hyperbolic space, tree visualization [LRP95] and change of focus, used in the implementation of the project (Section 4).

2.3.2.1 Hyperbolic Space

The hyperbolic visualization uses a different geometry, called non-euclidean geometry. This geometry differs from its euclidean counterpart because the fifth postulate, "parallel axiom", doesn't hold [WR02]. It means that within a two-dimensional plane, for any given line l and a point A, which is not on l, there is exactly one line through A that does not intersect l. In hyperbolic geometry, by contrast, there are infinitely many lines through A not intersecting l.

In order to represent the hyperbolic space, there are several models that define a real space which satisfies the axioms of a hyperbolic geometry:

Klein model - uses the interior of a circle for the hyperbolic plane and chords of the circle as lines.

Poincaré disc model - uses the interior of a circle, but lines are represent by arcs of circles that are orthogonal to the boundary circle.

Lorentz model - employs a 2-dimensional hyperboloid of revolution embedded in specific 3-dimensional space.

For the purpose of the thesis, only the Poincaré disc model used to implement the hyperbolic tree is explained in detail.

2.3.2.2 Mapping and Representation

The poincare projection is used as a model to map the hyperbolic space (H2) to the real space in order to draw the hyperbolic tree on the display. This model has been chosen due to a set of characteristics [WR02]:

Display compatibility: The infinite large area of the H2 is mapped entirely into a circle, the Poincaré disk (PD). This infinity representation

Circle rim: is close to infinity. All remote points are close to the rim, without touching it.

Focus+Context: The *focus* can be moved to each location in H2, like a FOVEA. The zooming factor is 0.5 in the center and falls (exponentially) off with distance to the FOVEA. Therefore, the context appears very natural. As more remote things are, the less spatial representation is assigned in the current display.

Lines become circles: All H2 lines appear as *circle* arc segments or (centered) straight lines in (PD). There extensions cross the PD-rim always perpendicular on both ends.

Conformal mapping: Angles (and therefore form) relations are preserved in (PD), area and length relations are not.

Parallel axiom: While the Euclidean space accommodates exactly one parallel to a line through a given point (not laying on this line), the H2 offers infinitely many. In (PD) a line is an arc who's ends are perpendicular to the circle rim. It is easy to draw many non-crossing circles through any given isolated point.

Exponential large space: There exist two kinds of parallels": *(i) asymptotic parallels*, circles which touch at the rim in the same INFINITE point and *(ii) ultra parallels*, circle arcs which do not intersect within (PD). One can antecipate, there is much more space in ∞ "usual" – sometimes the H2 is also called "more intensive infinite" that the $\Re^2$. This extra space is desired for finding good accommodation for our data.

2.3.2.3 Change of focus

For changing focus in (PD) we need a translation operation. In general, the transformations which do not change form and metric in H2 are, similar to $\Re^2$ the translations, rotation, and the reflection. In the Poincare' model H2-lines remain H2-lines, their corresponding (PD) circle arcs must remain (generalized) circle arcs in (PD).

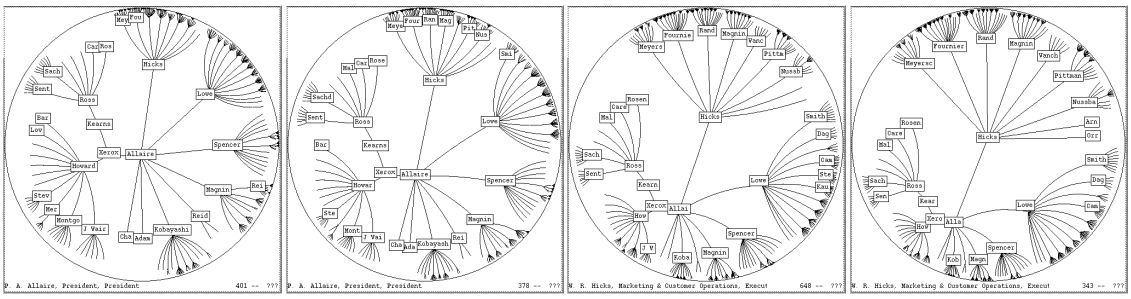


Figure 2.6: Changing the focus on the hyperbolic tree [LRP95]

The transformations that respect these properties (also called "circle auto-morphy" transformations) are here the *Möbius* transformations T(z) which can be formulated using complex numbers z, a, b:

$$T_{a,b}(z) = (az + \bar{b})/(bz + \bar{a})$$

By describing the Poincaré disk (PD) as the complex unit circle, the isometric transformations for a point $z \in (PD)$ can be rewritten in the more comprehensible form:

$$z' = T(z; c, \Theta) = \frac{\Theta z + c}{\bar{c}\Theta z + 1}, |\Theta| = 1, |c| < 1$$

Here the complex number Θ describes the pure rotation of (PD) around the origin 0. The following translation by c maps the origin to c and -c becomes the new center 0 (if Θ = 1). The inverse transformation is

$$T^{-1}(z;c,\Theta) = T(z;-\bar{\Theta}c,\bar{\Theta})$$

Two successive transformations in (PD) are computed faster by evaluating the concatenated transformation:

$$T(T(z;c_1,\Theta_1);c_2,\Theta_2) = T(z,c,\Theta)$$

with

$$c = \frac{\Theta_2 c_1 + c_2}{\Theta_2 c_1 \bar{c}_2 + 1}$$

and

$$\Theta = \frac{\Theta_1\Theta_2 + \Theta_1 \bar{c}_1 c_2}{\Theta_2 c_1 \bar{c}_2 + 1}$$

2.4 Summary

The development of user interfaces and interaction systems must be careful planned and designed. The user memory, attention and perception are key concepts that must be well understood in order to develop efficient interaction technologies. The discipline of interaction design is highly human and subjective, which means there are no strict or objective rules that lead to a successful solution. The process of development is iterative and experimental as it needs constantly user feedback.

Mobile devices are the new trend in the computing industry. They are small, easy to carry and connected to the world over the internet. However, they differ significantly from their desktop counterparts due to processing power and input limitations. Their applications should be focused on a small set of features typically used on mobility contexts.

Information visualization is the process of transforming raw data into a meaningful visualization. The use of it into mobile devices requires special attention regarding the visualization methods and the interaction techniques. The hyperbolic tree visualization technique offers a way of optimizing the space used to visualize graph-like content.

BACKGROUND

Chapter 3

Related Work

The effective visualization of information can make a wide range of applications more intuitive and productive. This chapter is focused on previous works and research related to interface design, hierarchical information visualization and mobile applications. The topics related to mobile applications and its limitations when compared to the desktop counterpart are also explored.

3.1 Exploring Networks

As more and more aspects of our lives become connected in the webbed environments of our culture, the information that we generate and consume is collected into massive databases [RK08]. The exploration of that large amount of data to build a visual narrative can be considered an "ancient art". *Mark Lombardi* [Fry08] created highly structured hand-drawn graphs of inter-relating parts called "narrative structures" about recent events. First, he compiles large amounts of information about a specific event (e.g. collapse of a large international bank) and then he draws the whole story depicting the political, social and economic map of interactions [Lom03].

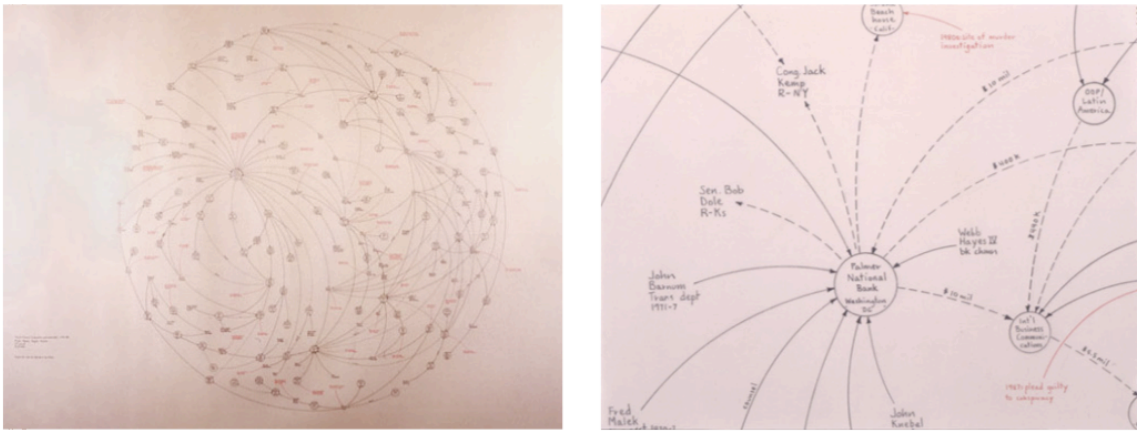


Figure 3.1: One of Lombardi's "narrative structures", and a detail of the piece

3.1.1 Graphic Design

3.1.1.1 Data Visualization of a Social Network

Location plays an important role in the online world of web communities. By forcing members of social networks to log their post code, a community provider knows what is the penetration of its membership is a specific urban area. On the other side, information such as age, gender and educational background about each individual is also public. Felix Heinen created a set of visualizations that focus on this characteristics based on the MySpace [MyS] social network [Hei07].

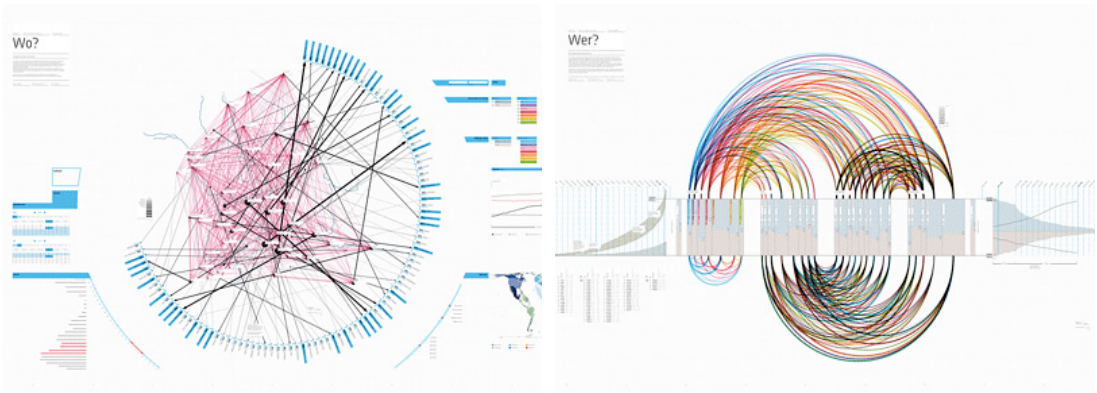


Figure 3.2: Felix Heinen's social network graphic visualizations

As shown in Figure 3.2, he explores in two different visualizations the concept of *where* and *who* by establishing relationships between members that share a demographic or geographic attribute. This graphics give providers a valuable snapshot of product, marketing and strategic customer information [RK08].

3.1.1.2 ClusterBall

Language can be seen as a semiological net, exposed through the internet by a web of volunteer workers. Wikipedia [Wik] is becoming the default reference to knowledge and understanding. Using its contents, Chris Harrison designed a visualization called "ClusterBall" [Har]. The visualization shows the connection between the concepts and the hierarchy in which they have been placed by *Wikipedians*.

ClusterBall starts with a unique entry on Wikipedia [Wik] and explores up to two levels of distance. Centered in the graph is a parent node. Pages that are linked from this parent node are rendered inside the ball. Finally, pages that are linked to the latter (secondary) nodes are rendered on the outer ring. Links between category pages are illustrated by edges, which are color coded to represent their depth from the parent node. Nodes are clustered such that edge lengths are minimized (Figure 3.3).

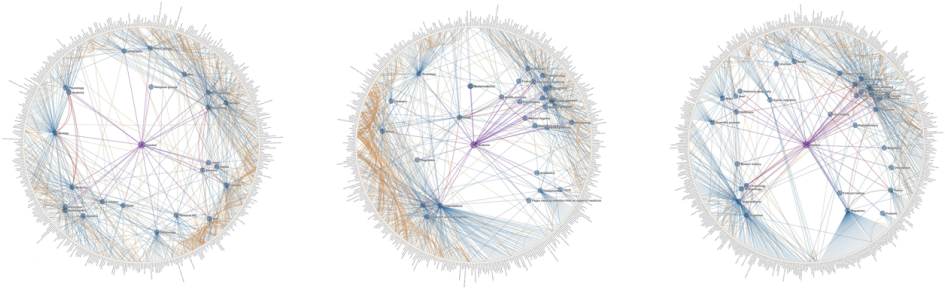


Figure 3.3: ClusterBall: Humans, Medicine, History

3.1.2 Internet Mashups

During the last years with the advent of Web 2.0 services and their public APIs, some projects were created based on the services content. Their main goal was hacking the existing information, either by finding new ways of visualizing it or by combining several services within one interface. The focus of this section are the internet mashups that explore new concepts of visualizing and interacting with the information.

3.1.2.1 Media Networks

The rise of internet services such as *Flickr*, *Amazon* or *Last.fm* where users have became a central piece either by producing content or by helping organizing, led to the creation of networks with high intrinsic value. Projects such as *Tuneglue* [ony] or *FlickrGraph* [Wes] allow the exploration of relationships between content available on the networks.

Tuneglue uses data from both *Amazon* (for user purchases) and *Last.fm* (for user listening history). It creates a springy network of artists based on similarity. Users provide a seed, starting artist, and can explore the resulting map by expanding any artist node. Similarly, *FlickrGraph* works by starting with a *Flickr* user and then exploring user's contacts by expanding the user node.

3.1.2.2 Interactive Arcs

As shown in Section 3.1.1, there is a trend about using arcs to establish relationships between concepts that are in some way connected. Recently, this technique has been applied to interactive applications that not only represent but also let the user explore and interact. *Digg Arc* [Dig] and *del.icio.us.discover* [Tim06] are two examples that use arcs in that way.

Ian Timourian developed *del.icio.us.discover* [Tim06]. The project aims to deliver implementable algorithms to recommend "links-of-interest" that might otherwise be missed

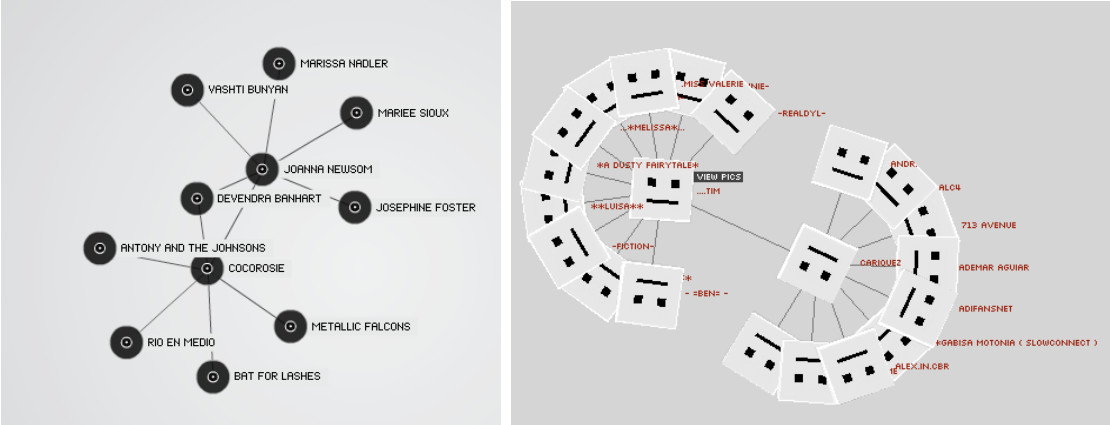


Figure 3.4: Tuneglue and FLickrGraph

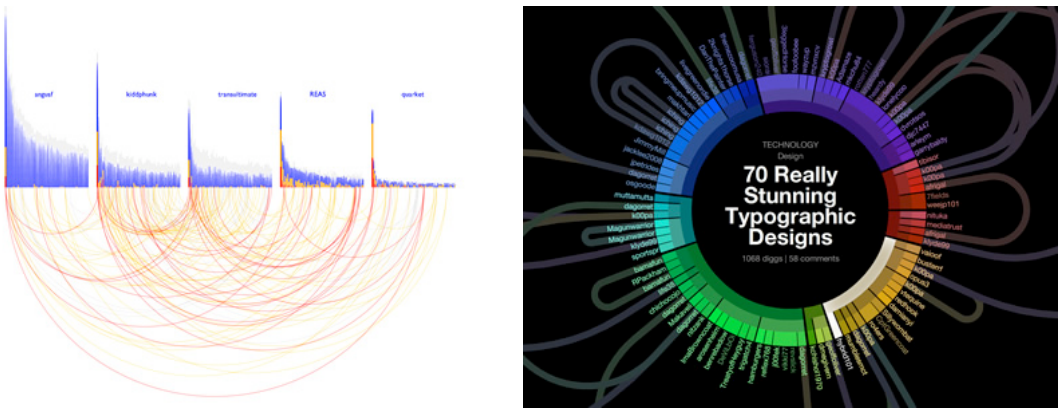


Figure 3.5: Examples of interactive arcs: del.icio.us.discover and Digg Arc

by a user currently utilizing only popularity-based link aggregators, and ideas for their further usages.

Digg Arc is part of a set of interactive visualizations from Digg Labs [Dig]. This visualization tool is useful for observing the level of activity within Digg. It displays stories, topics, and containers wrapped around a sphere. Arcs trail people as they Digg stories across topics. Stories with more Diggs make thicker arcs.

3.2 Hierarchical Information Visualization

Many hierarchies, such as organization charts or directory structures, are too large to display entirely on a computer screen. The conventional display approach maps all the hierarchy into a region that is larger than the display and then uses scrolling to move around the region. This approach has the problem that the user can not see the relationship of the visible portion of the tree to the entire structure (without auxiliary views). It would be useful to be able to see the entire hierarchy while focusing on any particular part so that the relationship of parts to the whole can be seen and so that focus can be moved to other parts in a smooth and continuous way.

Over the years there were several attempts to represent hierarchies using other than the typical 2D graph with node and edges. The main goal of the researchers is to display large amounts of data that are in someway inter-related more effectively [XBAA03]. In order to achieve this there are several techniques developed over the years that focus on data per area ratio improvement [BSW02, XBAA03, YC06, Fur86], node and edge cost visualization [RM05], that employ distortion techniques to distinguish between nodes [LRP95, Fur86, WR02] and navigation/searching focused visualizations [Cou07, HZ07]. These **focus+context** display techniques are the focus of this section.

3.2.1 Non-Distortion based Visualizations

The simplest technique to visualize a tree on a 2d surface optimizing the amount of space available is by using a radial layout [Bat99]. *RadialMobile* [YC06] applies this technique to mobile devices. It uses circles centered in the origin of the screen to distribute nodes according to their level in the hierarchy. The root node is in the center while its descendants are drawn at an n distance from the center. The distance n corresponds to the node's level in the hierarchy.

3.2.2 Generalized Distortion-based Visualizations

Many of distortion-based visualizations such as the document lens [Fur86] and the perspective wall [XBAA03] can be applied to browsing trees laid out using conventional 2d layout techniques. The problem is that there is no satisfactory conventional 2d layout of

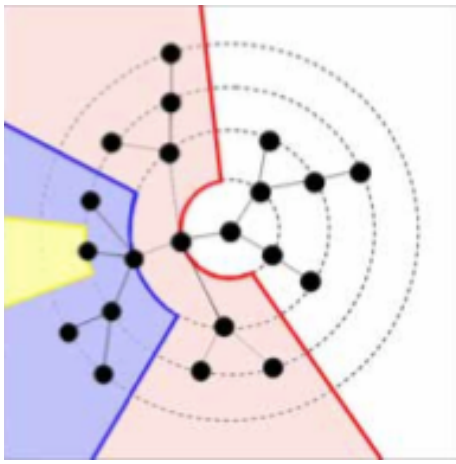


Figure 3.6: RadialMobile screenshot

a large tree, because of its exponential growth. If leaf nodes are to be given adequate spacing, then nodes near the root must be placed very far apart, obscuring the high level tree structure, and leaving no space to display the context of the entire tree at the same time.

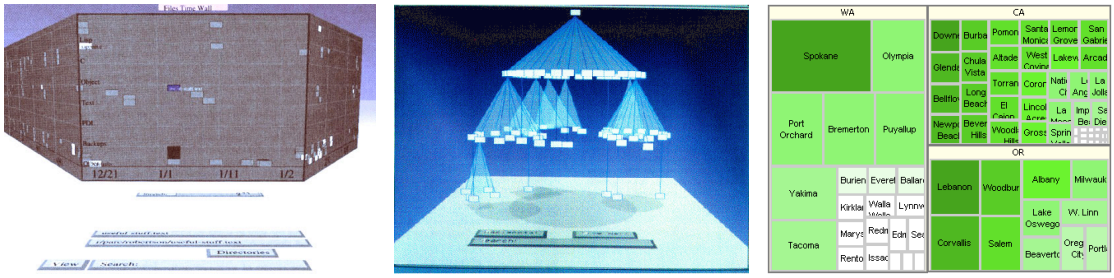


Figure 3.7: Perspective Wall, ConeTree and TreeMap

The Cone Tree [WR02] modifies the above approach by embedding the tree in a three dimensional space (figure 3.7). The embedding of the tree has joints that can be rotated to bring different parts of the tree into focus. This requires currently expensive 3D animation support. Furthermore, trees with more than approximately 1000 nodes are difficult to manipulate. The hyperbolic browser is two dimensional and has relatively modest computational needs, making it potentially useful on a broad variety of platforms. Another novel tree browsing technique is treemaps [BSW02] (figure 3.7) which allocates the entire space of a display area to the nodes of the tree by dividing the space of a node among itself and its descendants according to properties of the node. The space allocated to each node is then filled according to the same or other properties of the node. This technique utilizes space efficiently and can be used to look for values and patterns amongst a large collection of values which agglomerate hierarchically, however it tends to obscure the hierarchical structure of the values and provides no way of focusing on one

part of a hierarchy without losing the context.

Some conventional hierarchy browsers prune or filter the tree to allow selective display of portions of the tree that the user has indicated. This still has the problem that the context of the interesting portion of the tree is not displayed. Furnas [Fur86] introduced a technique whereby nodes in the tree are assigned an interest level based on distance from a focus node (or its ancestors). Degree of interest can then be used to selectively display the nodes of interest and their local context. Though this technique is quite powerful, it still does not provide a solution to the problem of displaying the entire tree.

3.2.3 Hyperbolic Tree Visualizations

The use of hyperbolic plane and trees to visualize large amounts of data has been used in several fields and it is implemented on commercial and open-source [Bel, Tou].

In crime analysis, law enforcement officials have to process a large amount of criminal data and figure out their relationships. It is important to identify different associations among criminal entities. *COPLINK* [XBAA03] is a Criminal Relationship Visualizer that uses a database to interactively retrieve and visualize criminal relationships. The project aims to construct an efficient, user-friendly information analysis and sharing system by integrating data from various sources and enabling more efficient crime analysis.

3.3 Summary

The representation of networked information either by exploring intrinsic relationships or by representing an existing hierarchy is a trend. Services such as *facebook* or *flickr* provide loads of rich data about people demography and geography that can be explored to provide new insight about a specific culture.

The visualization of hierarchical information has been a topic of interest during the last years. The use of distortion-based techniques has interesting effects on the way users perceive the information. The main reason is the easiness of keeping a global context around a detail view and the capability of showing more information using the same amount of space when compared to non-distortion based techniques.

Chapter 4

Methodology

The goal of the methodology chapter is to explain the processes and steps that led from the initial thesis proposal (Appendix A) to the developed visualization software. The main phases described are the problem analysis, problem specification, technological decisions, visualization software design, implementation details and the mobile applications that used the visualization module.

4.1 Problem Analysis

The initial problem proposed was the development of

> *[...] a software module that is capable of reading different kinds of data (photos, video, text, tags, etc) and create an hyperbolic view of it. This module should be used on a mobile device such as a PDA*

There are several issues related with the project proposal that should be addressed in order to formulate the problem correctly:

- the developed software should be modular in order to be integrated easily with other existing applications;
- it should support different kinds of content;
- the module must be developed for mobile devices, taking into account its characteristics.

The problem itself poses several technical and non-technical challenges:

Content. The data used to generate the hyperbolic view can be either local or remote data.

Hierarchy. The data gathered from the module may not have an explicit hierarchy defined. In this case, an algorithm is needed to organize and establish relationships between content.

Architecture. It is impossible to visualize without a data set previously structured. Does it make sense to separate the visualization from a hypothetical data gatherer?

Device. There are several mobile platforms available for development. Should the module support different platforms or should the prototype be focused on a single mobile platform?

Target. Who are the final users of the application?

Context. What is the most probable usage scenario for the module?

Performance. Are the current mobile devices capable of showing media content and support advanced graphics?

4.2 Problem Specification

The problem described in section 4.1 has several possible variations and thus allows different solutions. In order to focus the software development, a set of constraints was introduced. The revised project statement is:

> *A software module that is capable of reading personal media content (photos, video and text) stored on remote storage services such as Flickr, Facebook, Picasa or any custom service with a public API. The software module uses content metadata to merge and build a hierarchy in order create a hyperbolic view. This module should be used on a mobile device such as a PDA.*

The target user audience for this type of module varies according to the application that integrates it. As described in section 4.5, there is a wide range of applications for this visualization module that use different data sources and consequently usage environment or contexts. Figure 4.1 shows an overview of the problem and identifies several entities involved:

Flickr, Vimeo and Facebook: These are examples of web services that may be used as data sources of the visualization module. The content available can be video, photos, text, friends and has associated metadata (e.g. date, tags, title, geo-location).

Visualization Module: This is the piece of software responsible for data gathering, processing and visualization.

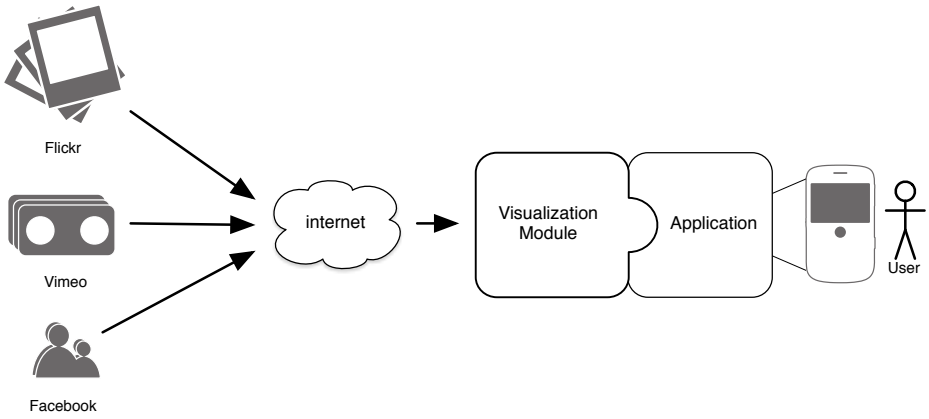


Figure 4.1: Problem overview

Application: It is an end-user application for the mobile device that uses the visualization module as part of its features.

User: The user of the application, that is registered in the web services described.

4.2.1 Requirements

4.2.1.1 Usage Scenarios

Scenario 1. The user is registered on several media content storage services (such as *flickr* and *picasa*) and social networks (such as *facebook*) with content uploaded by himself and related friends and contacts who also uploaded content. The main goal of the user is to browse through all the content while being able to define time slices and hyperbolic views. These views organize content according to content metadata such as tags, title, comments, friends proximity and also content attributes such as brightness, color and shape.

Scenario 2. The user wants to explore public content stored on services such as flickr and generate hyperbolic views based on time slices and establish an hierarchy based on content metadata or attributes.

Scenario 3. The same context as Scenario 1, but the user wants to use his current geo-location to establish a proximity relationship between content and generate an hyperbolic view with the focus on the content entity closer to his position.

4.2.1.2 Functional Requirements

- Support the hyperbolic visualization of data from social networks and media storage web services.

- Support a strategy for establishing relations among different kinds of content.
- Provide filtering mechanisms over the represented data.

4.2.1.3 Non-functional requirements

The non-functional requirements are:

- The architecture should support different remote storage services with a public API or any kind of remote access to content that may be combined together.
- The visualization views should be independent from the underlying data structure.
- The software module should be easily integrated within any mobile application as a mean of visualizing information.

4.2.2 Visualization Process

The visualization process as described in section 2.3 has several steps involved to transform the data gathered into an interactive visualization.

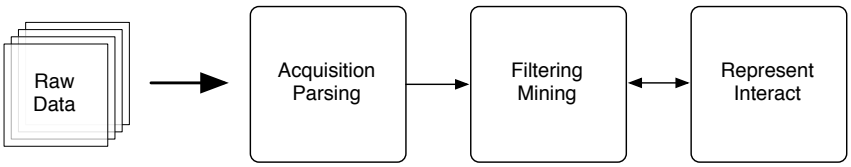


Figure 4.2: Visualization steps involved in the problem specification

Each of the visualization steps has specific technical challenges that need to be taken into account before the solution design (section 4.4). These steps, outlined in figure 4.2, are:

Analysis: the process of **acquiring** and **parsing** the data structures received from the remote web services. It needs to convert the data to an internal representation of the software module.

Organization: reduces the dimensionality of the resulting data set by **filtering** and **mining** according to the views selected.

Visualization: a graphical user interface **representing** a visualization of the selected content. It allows **interaction** over the data to select different views or filter by metadata attributes such as *date*. The visualization and organization steps work in parallel. The user actions regarding data manipulation are received by the organization step that filters and selects new data being visualized.

4.2.3 Interface Design

The interface design is the exercise of organizing the requirements (section 4.2.1) into a set of features. The hyperbolic tree view can be enriched with other visualization and interaction techniques. As mentioned in section 4.2.1, there is a need to create an hierarchy of the gathered data in order to establish a relationship between content and then represent it using an hyperbolic tree. This relationship can be established using tags, title or any other attribute such as the relationship with people. The date is an attribute that can be used to create a timeline view of the content and is an efficient way of filtering content by allowing *time slicing*.

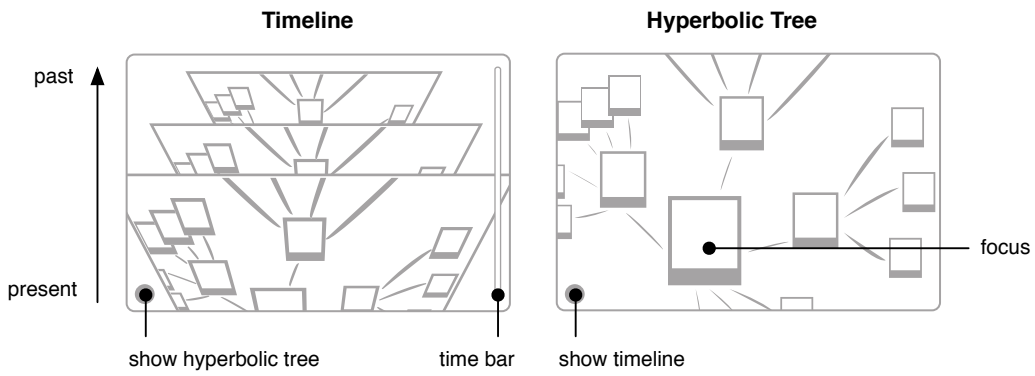


Figure 4.3: Initial interface mockup

The main features, represented visually in figure 4.3, of the visualization module are:

- Support the hyperbolic visualization of data from social networks and media storage web services
- Layout the content in a timeline view and allow the user to define a time slice that is applied to contents on the hyperbolic view
- Support a strategy for establish relations among different kinds of content
- Allow pinch/zoom over the hyperbolic view

4.3 Technological Decisions

As described in section 2.2 there are several mobile devices available in the market with differences regarding the development tools and applications ecosystem. As the software module described is graphics intensive, require advanced user interaction inputs and highly dependent on the connectedness of the device, the technological decisions had these factors into account. Due to time constraints the software module was implemented in only one platform, despite the title of the thesis.

The main technological decision were:

iPhone: The iPhone is a smartphone with several characteristics that make it ideal for this application. It has a multi-touch display providing a more flexible way of designing interactions that make content easier to manipulate.

Graphics Library: OpenGL ES Library

Data Storage: SQLite

4.3.1 iPhone

As shown in Section 2.2.3, smartphones are the category of mobile devices with more features compared to its size. The iPhone was chosen because:

- Wifi and 3G
- Multitouch display
- GPS and Accelerometer
- Popularity
- Development tools and SDK

4.3.2 OpenGL ES

The current version of OpenGL ES supported on the iPhone 3G is 1.1. It provides 24MB of memory and is hardware accelerated. When compared to other graphic libraries (e.g. CoreAnimation), it requires an additional effort to implement basic drawing routines but gives several improvements and gains over flexibility and performance.

An experiment with OpenGL ES 1.1 can be found on Appendix B.1

4.3.3 Data Storage

The data is stored on a SQLite database. The development framework provides mechanisms to access directly to the database, meaning a major improvement over other storage files such as XML or text file. The experiments with data storage can be found on Appendix B.2.

4.4 Design and Implementation of the Visualization Module

The visualization module is the main goal of this thesis.

4.4.1 Overview

As described in section 4.2.2 about the visualization process, there are three main steps that successively transform raw data from the internet into a organized set of content represented visually.

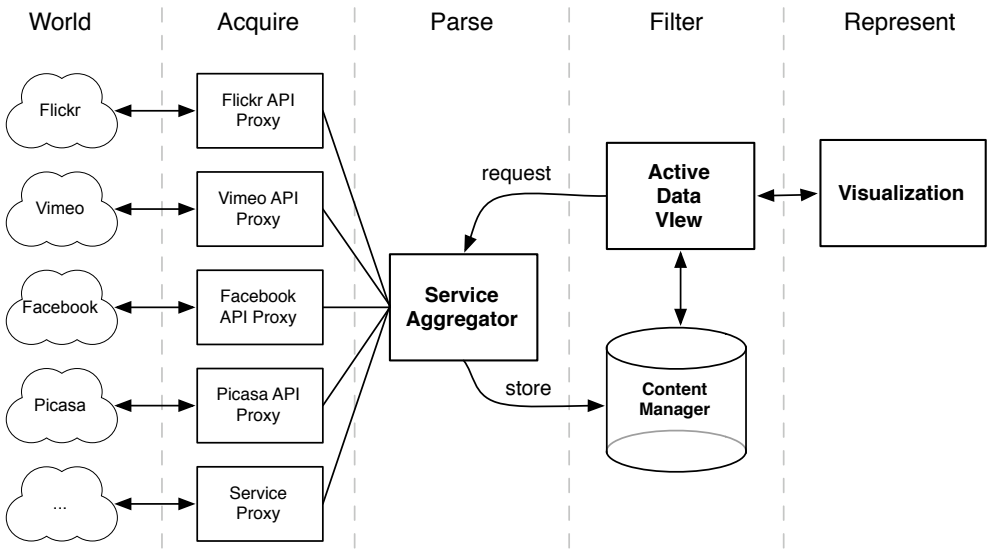


Figure 4.4: Architecture overview of the visualization module

The module was designed to be totally independent from any application it might be integrated with. As shown in figure 4.4, given a specific set of data sources, is capable of gathering the data, filtering it according to the visualization views and converting all the data into a generic data structure. This generic data structure (section 4.4.4) is cached and has mappings that allow itself to be used in the different visualization views (section 4.4.5).

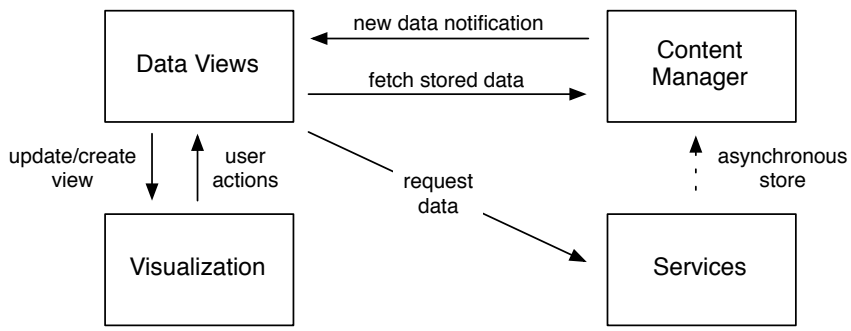


Figure 4.5: Main architecture module interaction

The architecture follows the Model View Controller (MVC) pattern. The *Data Views* is a set of controllers responsible for exploring the "world". The data retrieved is stored in a *Content Manager*. The *Data Views* feeds the visualization with content and requests more data according to the visualization needs. The figure 4.5 shows the 4 different modules:

Content Manager. The content manager is responsible for keeping the data stored in the generic data format. It is also responsible for caching and merging operations.

Services The service aggregator keeps track of all remote services configured and active in the application. When asked for data it forwards these requests to each service and merges the responses before sending back to the content manager. It also defines an interface that each service must implement in order to receive and respond to its requests.

Data Views. The data views are the ones that implement the data acquisition and filtering strategy. When selected they are responsible for updating the graphic view, fetching new data available in the content manager and asking the service aggregator for more data.

Visualization The views implemented are the Hyperbolic Tree and the Timeline. The views are tightly coupled with the data views selected.

A detailed view of the dependencies between the classes is on appendix C.1 class diagram.

4.4.2 Services

The services software module is responsible for all the communication needed with the remote web services. It includes an *Aggregator* that receives requests from a *Data View*, forwards the requests to all its active *services* and stores asynchronously the data received through the *Content Manager*.

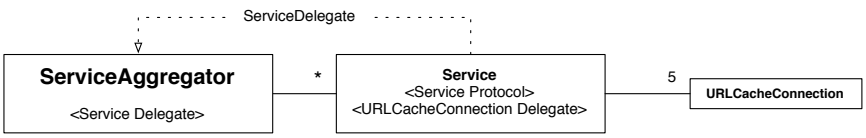


Figure 4.6: Class diagram of the service related classes

As shown in figure 4.6, the classes that manage the *web services* are:

Service Aggregator: The service aggregator is responsible for forwarding the requests to all the active services and then merge all the responses in order to store the data. It

implements the *Adapter* pattern to convert the data format received from the services into a normalized structure, internal to the *Content Manager*. It also implements the *Service Delegate* Protocol to receive in asynchronous mode the responses from the services.

Service: The service implements the *Service Protocol*. This way it is possible to implement new services without subclassing. It follows the *Proxy* pattern as it provides a way of accessing a service with data to be fetched, either through a public API or direct access to the storage space (e.g. Webdav).

URLCacheConnection: This class is responsible for fetching data over HTTP on a specific URL (Universal Resource Locator). It has means of caching the results for a URL. Due to memory limitations of the underlying framework, each *Service* starts only up to 5 simultaneous requests.

4.4.3 Data Views

The *Data View* software module is responsible for implementing the strategy needed to explore the data available remotely and represent it. It has knowledge about both the *web services* and the visualization being displayed currently. If more data is needed it sends a request to the *Service Aggregator*. All the new data is stored directly through the *Content Manager*. In order to know when new data is available, so that it can update the visualization, the *Data View* registers itself as an *Observer* of the *Content Manager*, to be notified every time new data arrives.

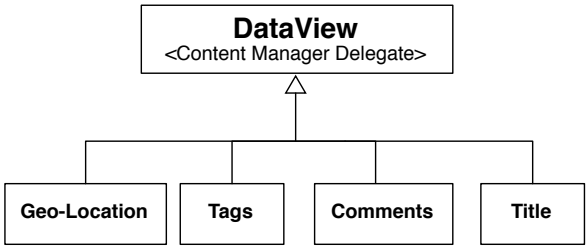


Figure 4.7: Class diagram of the data view related classes

As figure 4.7 shows, there are subclasses for each specific data view. For example, there is a specific *data view* that builds an hyperbolic view based on the liked content of some person in the network.

As shown in figure 4.8 it is possible to use a *Data View* that explores the relations between content and people such as content that people like (sometimes also referred as favorites). In this case those relationships are extracted from the network (already stored

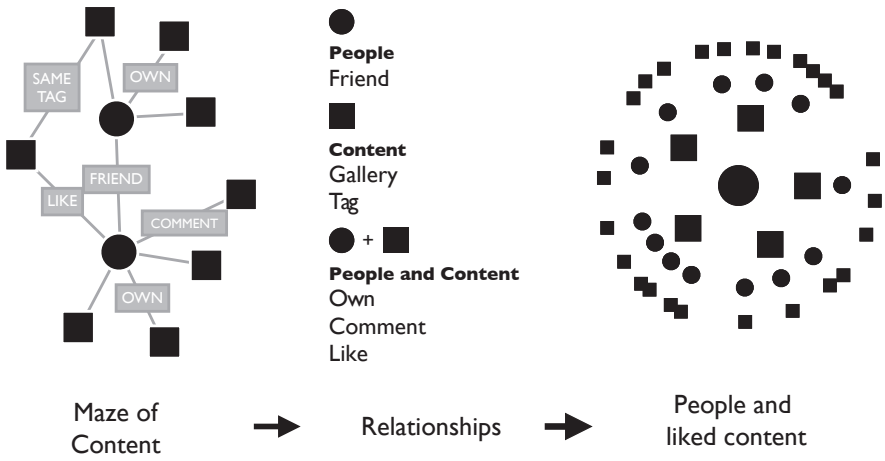


Figure 4.8: Example of a Data View for liked content

locally) and from a single person (e.g. me) the tree expanded following the logic *person – content he likes – people who like that content* recursively until a certain level of depth.

4.4.4 Content Management

The content management is responsible of keeping an internal representation of the data.

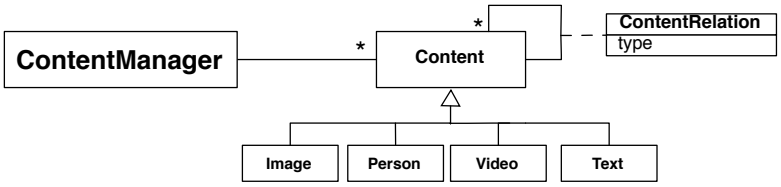


Figure 4.9: Class diagram of the content management related classes

As shown in figure 4.9, the internal structure has two important classes:

Content stores the content data and attributes such as date, title, geo-location, description and tags.

ContentRelation relates any kind of content. It has an attribute *type* that defines the degree of relationship. The defined relationships are between people, content or people and content.

4.4.5 Hyperbolic Tree Visualization

4.4.5.1 Layout

The Hyperbolic tree is represented from a graph structure. In order to calculate the layout position, a root node (initial focus) must be selected. The final positions are calculated in

three steps:

1. Compute tree graph levels for each node;
2. Compute angular widths for each successive subtree
3. Compute final positions for each node

The angular width of each subtree is calculated based on the node's level and parent's angular width. The final position are represented as complex numbers. The algorithm has the following steps:

1. Flag the root node (*focus*)
2. Do breadth first traversal and for each node
   (a) Compute the depth
   (b) Computer the angular width for the descendants subtree
   (c) Compute the position

4.4.5.2 Picking

The picking is used to determine the node selected when the user touches the screen. Instead of using the internal's opengl mechanism (querying the scene buffer), the internal structure for the nodes is used. The sequence is:

1. Retrieve the position of each touch (more than one touch)
2. Iterate over all nodes and compare their position against the touch positions
3. Pick touched nodes
4. start animation

4.4.5.3 Change of focus

To change the focus of an hyperbolic tree only a möbius transformation is needed. However, in order to animate the change a set of successive transformations is applied. The sequence is:

1. Calculate versor from focus to touched node
2. Scale versor according to animation delta
3. Apply successive möbius transformations using scaled versor to each node position

4.5 Applications

4.5.1 Kleiner Expresso

The *KleinerExpresso* application was thought and developed exclusively as a showcase of the visualization module features. The main purpose is a need in the mobile industry, but so far it is an early prototype used for testing only. It is a life streaming application for *flickr* that explores someone's public gallery and his contact's content.

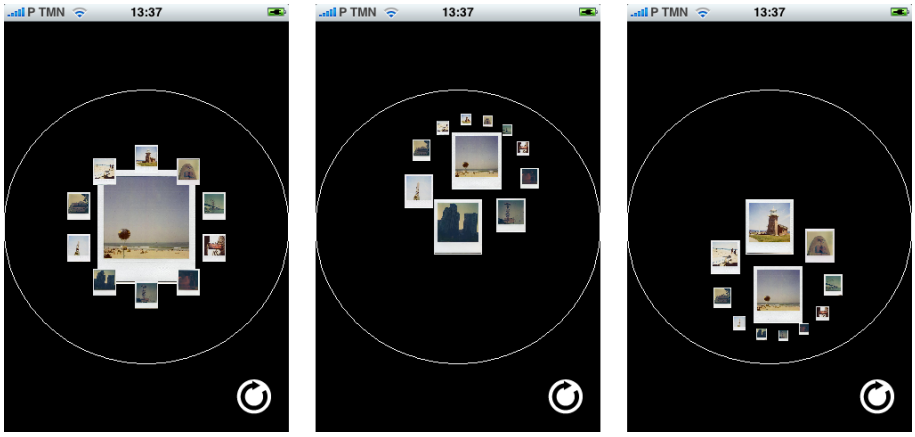


Figure 4.10: Screenshots of the kleiner expresso application

4.5.2 TinDI

The *TinDI* application is a port of an existing Android application. It works with t-mobile's mediencenter, a storage service for photos, videos, music and other kind of files. Besides visualization, the application also supports uploading your own photos either on a local library or using the phone's camera.

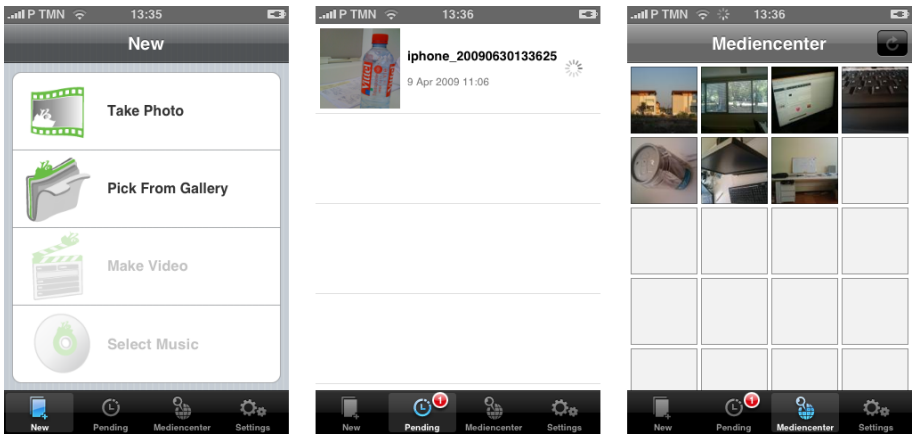


Figure 4.11: Screenshots of the TinDI application

Usage of mediencenter to show the it with an hyperbolic view. In this case the mediencenter is the only data source.

4.6 Summary

The project's implementation has two parts: a visualization module and mobile applications that use that visualization module. In order to integrate the visualization module with different applications, the module must be flexible enough to allow different kinds of content and data sources. The visualization module uses remote data sources such as *flickr* or *facebook* to provide content. It is supposed to be used as a lightweight application for showing your photos and other media content to friends in a intuitive way.

The technologies used for the first prototype are based on graphics power of the mobile device and easiness of development. The platform of choice is iPhone because it offers the main networking devices, hardware accelerated graphics for OpenGL ES and a broad development framework with support for networking, interface design and graphics.

The implementation of the software module was decomposed in 4 parts: services, data views, content management and graphic views. The services implement software proxies responsible to access public APIs and gather content from web services. The Data views are controllers responsible to make content requests according to the selected view on a hyperbolic tree. The content management is where the content retrieved from the web is stored and cached for later use. The graphic view include the hyperbolic tree and a custom timeline view for date filtering and slicing.

Two application where developed that use the visualization module. *KleinerExpresso* is a prototype application designed exclusively to showcase the module features. *TinDI* is an application that allows the user to upload content and view the content on a private web service.

METHODOLOGY

Chapter 5

Discussion and Analysis

This chapter describes the evaluation processes for the solution, the successes and the shortcomings of the project and a comparison of the developed applications.

5.1 Evaluation

The evaluation of a prototype is an important step of any software project. In this project it is particularly important to evaluate the UI with a group of potential users. This is the way to obtain results about the system's usability and overall usefulness. Even though the prototype was not tested inexhaustibly, the formal process is described in this chapter as a guide for future work.

5.1.1 Information Visualization Process

The techniques and criteria described in this section are focused on the specific characteristics of the information visualization process.

5.1.1.1 Criteria

The criteria [Pla04] that can be addressed in an evaluation process are:

Functionality: It consists on comparing the user requirements with the features provided by the system.

Effectiveness: It is focused on the visualization purposes and determining if it provides new insights compared to other previous solutions.

Efficiency: Analyses if the visualization helps the users achieve a better performance with their tasks using it.

Usability: Tests how the users interact with the system and of the information provided is clear and intelligible.

Usefulness: Analyses the visualization within its usage context. It determines if the visualization is useful and how users may benefit from it.

5.1.1.2 Techniques

The evaluation process has several possible techniques that can be grouped into two main categories, analytics and empiric evaluations [TZ05].

Analytic evaluation: based on formal analysis models and conducted by experts.

- **Heuristic:** experts act as less experienced users and describe the potential problems they foresee for such users;
- **Cognitive walkthroughs:** experts walk through a specific task using a prototype.

Empirical evaluation: realized through experiments with user test subjects.

- **Quantitative studies:** consist of an analysis to determinate hypotheses tested through direct measurements. Usually done within controlled experiments, where an evaluator can manipulate a number of factors associated with interface design and study their effects on various aspects of user performance.
- **Qualitative studies:** involves the analysis of qualitative data, which may be obtained through questionnaires, interviews and observing users using the system, to understand and explain social phenomena. Usually done with a focus group of selected individuals.

5.1.1.3 Metrics

The metrics used for HCI evaluation are not appropriate for assessing information visualization systems. Some of the HCI metrics used in this field are:

- time required to learn the system;
- time required to achieve a goal;
- error rates;
- retention of the use the interface over the time.

5.1.1.4 Process

The developed visualization module aims to demonstrate that:

- The use of an hyperbolic tree to display content on a mobile device is useful.

- Manipulating and interacting with media content using the techniques described in Chapter 4 is effective by helping users to discover forgotten or unfamiliar friends and/or content.
- The interface and the visualizations provided are well integrated and make the content easy to navigate.

In order to test each of these items, a small user study should be designed to touches upon each of them. It starts with a focus group of 5 users unfamiliar with the visualization and application goals. It takes about 30 minutes. During this period the visualization is explained and demoed briefly, the user has the chance to play with the visualization and execute specific tasks. The results are evaluated through a qualitative questionnaire about the user experience.

Each 30-minutes evaluation session follows this structure:

1. (5 min) Introduction to the visualization module with an active *flickr* account. The explanation is complemented with a brief demo where the main interaction techniques are shown.
2. (10 min) The subject tries to use the visualization freely in order to understand the main action-effect behaviors of the application
3. (10 min) The user is asked to execute specific tasks:
   (a) View the photos of a specific friend's contact
   (b) View the photos of all his contacts uploaded during the last week
   (c) View the photos that are related to the tag "Polaroid" taken in Portugal during the last week up to one level of contacts
4. (5 min) Fill the questionnaire.

5.1.2 Empirical Results

Due to time restrictions the visualization module was not subject to the qualitative evaluation described in section 5.1.1.4. However, the applications described in section 4.5 were tested and analyzed by a UI Consultant from Apple, at the World Wide Developers Conference 2009 [App09]. The feedback can be summarized:

Usability: The interface is the content. In order to make it easy how to manipulate content, help dialog should be shown on the first the applications are launched.

Effectiveness: The visualization technique is not new. However, the way it is being used to display content is very effective from the navigation point of view. The focus is not obvious. The content should have labels containing a title, tags or description.

Tips:

1. Analyze the tab most used by the user and every time the application is launched automatically open that tab.
2. Users prefer motion and moving rather than tap or touch. The content nodes should move and be mapped on a sphere like object. Associate "springs" to content relations giving an idea that the position is not static.

5.2 Successes, Challenges and Pitfalls

Throughout the research and development process there were some successes were the hypothesis was successful proved and achieved. However several technical challenges that made impossible the fulfillment of certain goals. This section provides a glimpse about the main issues to be taken into account in the field of infovis research.

- *aesthetics* The aesthetics of a visualization are an important issue. Users are more attracted to beautifully designed graphics.
- *help* Help dialogs are a useful way of giving hints to the user about things he can do.
- *hyperbolic navigation* The user is rapidly engaged with the navigation flow and easily learns how to use it
- *Time slicing* The time filtering is frustrating as the user needs to change views
- *network* In order to achieve the best performance, the creation of queues is important.
- *scalability* The current prototype doesn't deal with large data sets (more than 100 entities). It is an important future improvement.
- *animation* the change of focus animation is important to give the user a notion of what is moving and changing
- *portability* A port to another mobile platforms is planned. The OpenGL code is cross-platform compatible. However, the supporting code for networking and interface design is specific.
- *memory* Memory is an important constraint on mobile devices. It is important to manage memory efficiently in order to avoid leaks and performance drops.

5.3 Summary

The evaluation of the prototype is focused on the usability and interaction aspects. Despite the fact that it was not possible to test it with a focus group of subjects, some important hints were given by the UI consultants. It is important to give users the chance to move and manipulate the content rather than tapping, because it gives them a bigger sense of ownership and power.

The development of mobile applications poses several challenges. The most important aspects of mobile development is memory constraints and efficient interface design.

DISCUSSION AND ANALYSIS

Chapter 6

Conclusion

This thesis has laid out a new way to design and develop visualization and interaction techniques for mobile devices. However, it is important to mention some of the limitations found during the project. The latest mobile platforms offer a great deal between performance and interaction capabilities. In order display information and media content the quantity of content displayed must be limited. The network and graphic performance is still slow compared to desktop computers. On the other side, information visualization is a broad field that includes subjects from different research fields. In order to achieve interesting and complete solutions a multidisciplinary approach is required. Both the visualization, interaction, data analysis and gathering need the same effort and play a vital roll. A well implemented and intuitive visualization without an organized and efficient data set doesn't work. The same principle applies the other way around.

6.1 Future Work

How can the hyperbolic tree visualizer be a better tool for navigating and exploring the new dimensions of media content? The results presented in this thesis were about the early prototypes of the software. Despite some planned goals only the one with high priority were implemented. Here, some of the planned improvements are briefly described.

6.1.1 Implementation

The implementation of the visualization software was mainly focused on the main features and architecture requirements. Essentially, the main goal was implementing a hyperbolic tree visualization that would be used as visualization module for an application. Issues related with performance, network robustness and memory efficiency were planned but will be subject of attention during the future versions.

Network - The communication with external services is an essential part of the module. It is important to keep redundant request at the minimum possible due the memory

devices limitations. On the other side, lots of requests at the same time may cause memory problems or even network stack overflow. That way, it is important to create a centralized mechanism to manage the requests, creating a maximum of simultaneous connections (not only per service) and implementing a way of detecting failures and retrials.

Memory - Mobile devices have serious memory limitations. In this case the OpenGL context is managing several textures simultaneously without any memory reduction mechanism. As most of the nodes shown in the hyperbolic tree have considerable small size compared to the focus node, it is important to create intermediate sizes (stored in the local memory) in order to adjust texture size accordingly.

Caching - Memory and network resources are one of the mobile devices main limitations, as referred throughout the thesis. So far, caching is only applied to the URL requests over HTTP done via the services. It is important to create caching policies regarding the content being displayed and the views built from the last application session.

Android - The current implementation was designed and thought to be implemented on the iPhone platform. It is planned to develop an android variant that support the basic visualization features. However, the effort to port to Android is considerable big. Even tough OpenGL was used, most of the architecture and artifacts available in the frameworks and language used are implemented in a different way on the android.

6.1.2 Design

The design process was full of rich ideas and concepts that could be applied to the existing hyperbolic tree visualization. Despite the current solution, here are some of the design features planned for future versions.

Anamorphic View - The current mobile devices have an internal accelerometer that provides the device orientation regarding *XYZ* axis. That way, it is possible to infer the device's position relatively to the users head. The anamorphic view is a technique used to give a sense of depth on interfaces and drawings by adapting the perspective according the user's position. It could be used to change the mapping a give a sense of 3D.

3D View - Use a 3D View to show multiple kinds of relationships simultaneously (e.g. tags and geo-location).

Hyperbolic Geometry - Most of the advantages offered by the Hyperbolic tree come from the underlying geometry. For content that cannot be related using a graph

(e.g. geographic location), the hyperbolic geometry is a efficient way of showing 2D content.

Physic Engine - Each node is a physic object. The node's position is not static and move with finger touches. The idea is create spring-like relationships that keep node's always in the same position but allow the user to move them with a certain degree of freedom.

Data Source - Redesign the *services* architecture to support the integration of local content (e.g. local library of photos).

6.2 The Future of Mobile Devices

The concept of Mobile Device as suffered a big impact during the last 3 years. This change has been strongly catalyzed due to the recent devices available on the market such as the Android, Blackberry, Palm Pre and the iPhone. The iPhone was the decisive device with a simple interface that empowered the average users. The iPhone was an inflective moment in the computing industry the same way the PC was 25 years ago.

The future versions of mobile devices will focus on performance, memory improvements and new graphic capabilities. However, the big changes are about market evolution, interface design paradigms and usage habits.

6.3 Summary

The initial goal of the project was the development of a software module that used an hyperbolic tree view technique for infovis purposes. This goal was totally achieved. However, it requires some future work regarding the performance and testing. The main focus of future work is improving memory and networking performance. It is also planned to port the existing solution to the Android mobile platform.

The mobile device market is going to create a big impact in the computing and entertainment industry during the next three years.

Conclusion

References

[AKK06] Benjamin B. Bederson Amy K. Karlson. Understanding single-handed mobile device interaction. 2006.

[App09] Apple. World wide developers conference 2009. http://developer.apple.com/wwdc/, 2009. Accessed 20th June 2009.

[Bat99] Battista. 1999.

[Bel] Nicolas Belmonte. Javascript infovis toolkit. http://thejit.org/. Accessed 2nd June 2009.

[Ber99] Jacques Bertin. Graphics and graphic information processing. pages 62–65, 1999.

[BNSW94] Norman Adams Bill N. Schilit and Roy Want. Context-aware computing applications. 1994.

[BSW02] Benjamin B. Bederson, Ben Shneiderman, and Martin Wattenberg. Ordered and quantum treemaps: Making effective use of 2d space to display hierarchies. *ACM Trans. Graph.*, 21(4):833–854, 2002.

[Chi04] Luca Chittaro. Hci aspects of mobile devices and services. *Personal Ubiquitous Comput.*, 8(2):69–70, 2004.

[Chi06] Luca Chittaro. Visualizing information on mobile devices. *Computer*, 39(3):40–45, 2006.

[CL72] Fergus I. Craik and Robert S. Lockhart. Levels of processing: A framework for memory research. *Journal of Verbal Learning & Verbal Behavior*, 11(6):671–684, December 1972.

[Cou07] Rémi Coulom. Treemaps for search-tree visualization. 2007.

[Dig] Digg. digg labs. http://labs.digg.com/. Accessed 9th June 2009.

[Fac] Facebook. Welcome to facebook! — facebook. http://www.facebook.com/. Accessed 9th June 2009.

[FG89] Sigmund Freud and Peter Gay. *Civilization and Its Discontents*. W. W. Norton & Company, 1989.

[Fli] Flickr. Welcome to flickr! http://flickr.com/. Accessed 9th June 2009.

[Fry08] Benjamin Fry. *Visualizing Data*. O'Reilly, first edition, 2008.

[Fur86] G. W. Furnas. Generalized fisheye views. *SIGCHI Bull.*, 17(4):16–23, April 1986.

[Har] Chris Harrison. Clusterball: Humans, medicine, history.

[Hei07] Felix Heinen. Data visualization of a social network, 2007.

[HZ07] Jie Hao and Kang Zhang. A mobile interface for hierarchical information visualization and navigation. In *Consumer Electronics, 2007. ISCE 2007. IEEE International Symposium on*, pages 1–7, June 2007.

[Irw92] D. E. Irwin. Memory for position and identity across eye movements. *Journal of experimental psychology. Learning, memory, and cognition*, 18(2):307–317, 1992.

[Lom03] Mark Lombardi. *Mark Lombardi, Global Networks*. Independent Curators International, New York, USA, 2003.

[LRP95] John Lamping, Ramana Rao, and Peter Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 401–408, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[Mar82] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt & Company, June 1982.

[Met95] R. Metcalfe. Metcalfe's law: A network becomes more valuable as it reaches more users. *Infoworld*, (17), 1995.

[Mog06] Bill Moggridge. *Designing Interactions*. The MIT Press, Cambridge, Massachusetts, 2006.

[MyS] MySpace. Myspace. http://www.myspace.com/. Accessed 9th June 2009.

[Oli89] Wally Olins. Corporate identity: Making business strategy visible through design, 1989.

[ony] onyro. Tuneglue°— relationship explorer. http://audiomap.tuneglue.net/. Accessed 9th June 2009.

[O'R92] J. K. O'Regan. Solving the "real" mysteries of visual perception: the world as an outside memory. *Canadian journal of psychology*, 46(3):461–488, September 1992.

[Pla04] Catherine Plaisant. The challenge of information visualization evaluation. 2004.

[Pos09] Stefan Poslad. *Ubiquitous Computing: Smart Devices, Environments and Interactions*. Wiley, 2009.

[Ren00] Ronald A. Rensink. The dynamic representation of scenes. *Visual Cognition*, 7(1 - 3):17–42, January 2000.

[RK08] S. Ehmann-F. van Heerden T. Tissot R. Klanten, N. Bourquin. *Data Flow, Visualizing Information in Graphic Design*. gestalten, 2008.

[RM05] Juergen Rilling and S.P. Mudur. 3d visualization techniques to support slicing-based program comprehension. *Computers & Graphics*, 29(3):311 – 329, 2005.

[Sha03] Gary Shapiro. *Foucault and Nietzsche on Seeing and Saying Archaeologies of Vision*. The University of Chicago Press, first edition, 2003.

[Spe60] George Sperling. The information available in brief visual presentations. *Psychological monographs: General and applied*, 74(11):1–29, 1960.

[Str33] 1897-1973. Stroop, J. Ridley (John Ridley). Studies of interference in serial verbal reactions. 1933.

[Tim06] Ian Timourian. del.icio.us.discover. http://www.mandalabrot.net/delicious/, 2006. Accessed 10th June 2009.

[Tou] LLC TouchGraph. Touchgraph — products: Navigator. http://www.touchgraph.com/navigator.html. Accessed 2nd June 2009.

[TZ05] Petra Neumann-Mark S. Hancock Sheelagh Car pendale Torre Zuk, Lothar Schlesier. Heuristics for information visualization evaluation. 2005.

[War04] Colin Ware. *Information Visualization, Second Edition: Perception for Design (Interactive Technologies)*. Morgan Kaufmann, April 2004.

[Wes] Marcos Weskamp. Flickr graph - marcos weskamp. http://marumushi.com/apps/flickrgraph/. Accessed 9th June 2009.

[Wik] Wikipedia. Wikipedia. http://www.wikipedia.com/. Accessed 9th June 2009.

[WR02] Jörg A. Walter and Helge Ritter. On interactive visualization of high-dimensional data using the hyperbolic plane. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 123–132, New York, NY, USA, 2002. ACM.

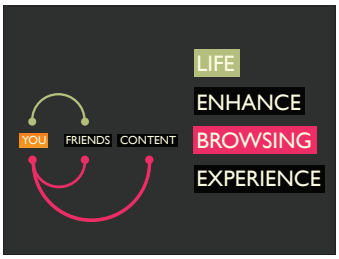
[XBAA03] Yang Xiang, Michael Chau B, Homa Atabakhsh A, and Hsinchun Chen A. Visualizing criminal relationships: comparison of a hyperbolic tree and a hierarchical list, 2003.

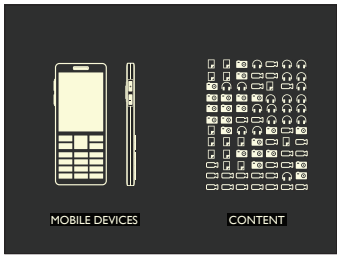
[YC06] Hee Yong Yoo and Suh Hyun Cheon. Visualization by information type on mobile device. In *APVis '06: Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation*, pages 143–146, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc.

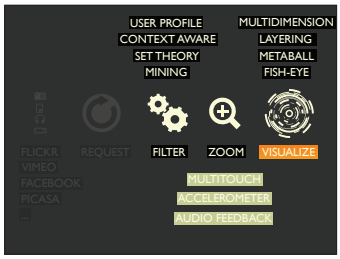
REFERENCES

Appendix A

Thesis Proposal: Initial Presentation







Appendix B

Technological Experiments

B.1 Whiterose

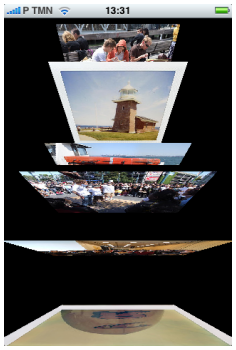


Figure B.1: Screenshot of the whiterose opengl ES experiment

B.2 Gunpowder

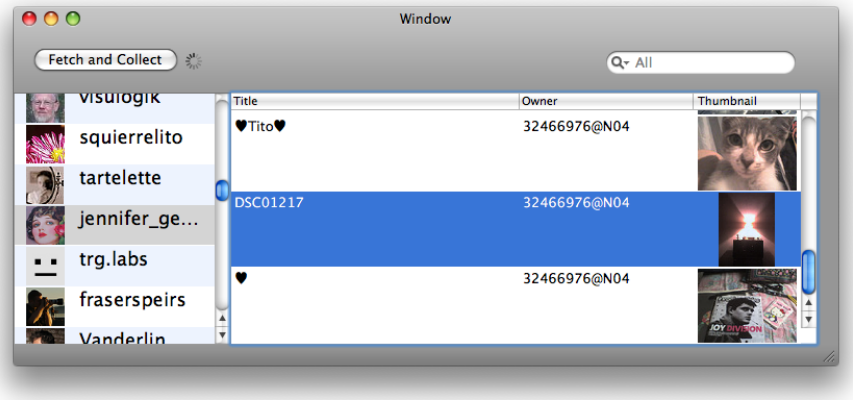


Figure B.2: Screenshot of the gunpowder core data experiment

Appendix C

Visualization Module Implementation Details

C.1 Class Diagram

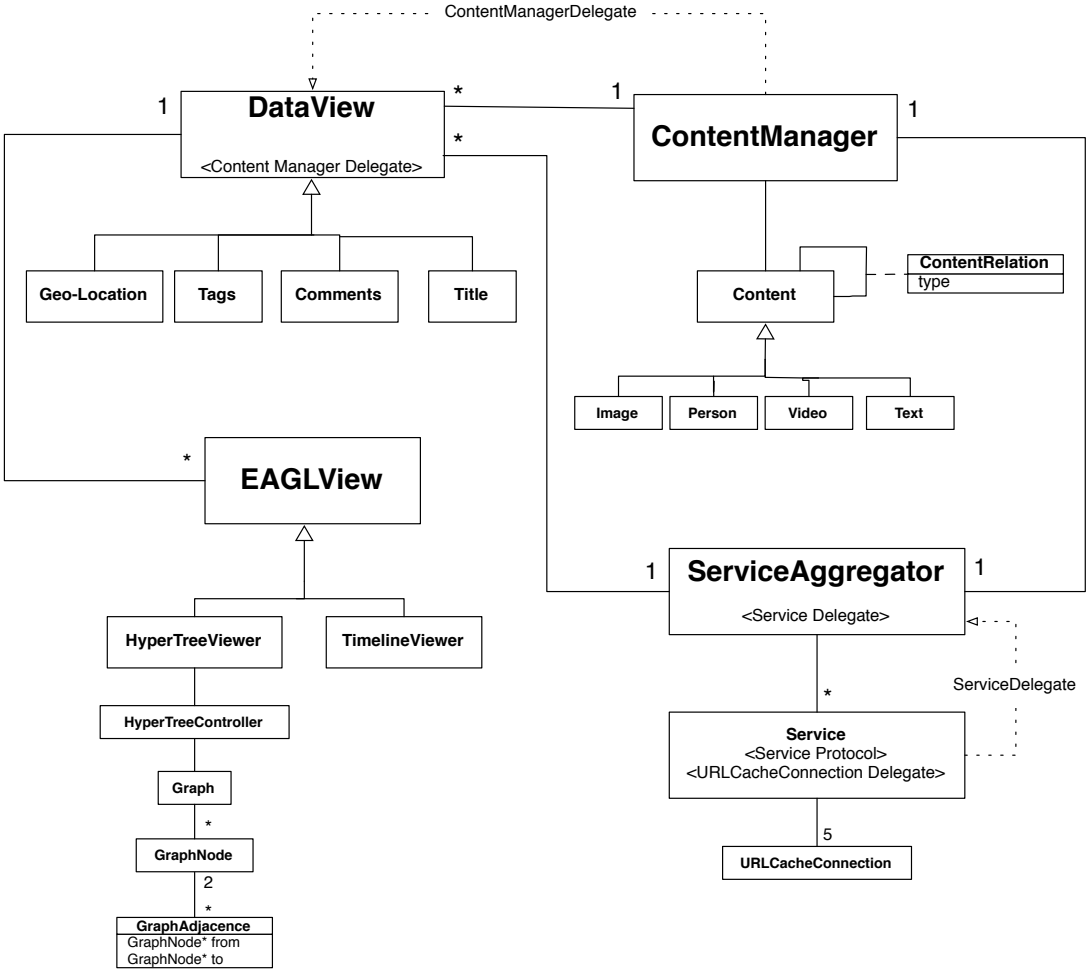


Figure C.1: Complete class diagram of the visualization module

Appendix D

Suplementary Sketches

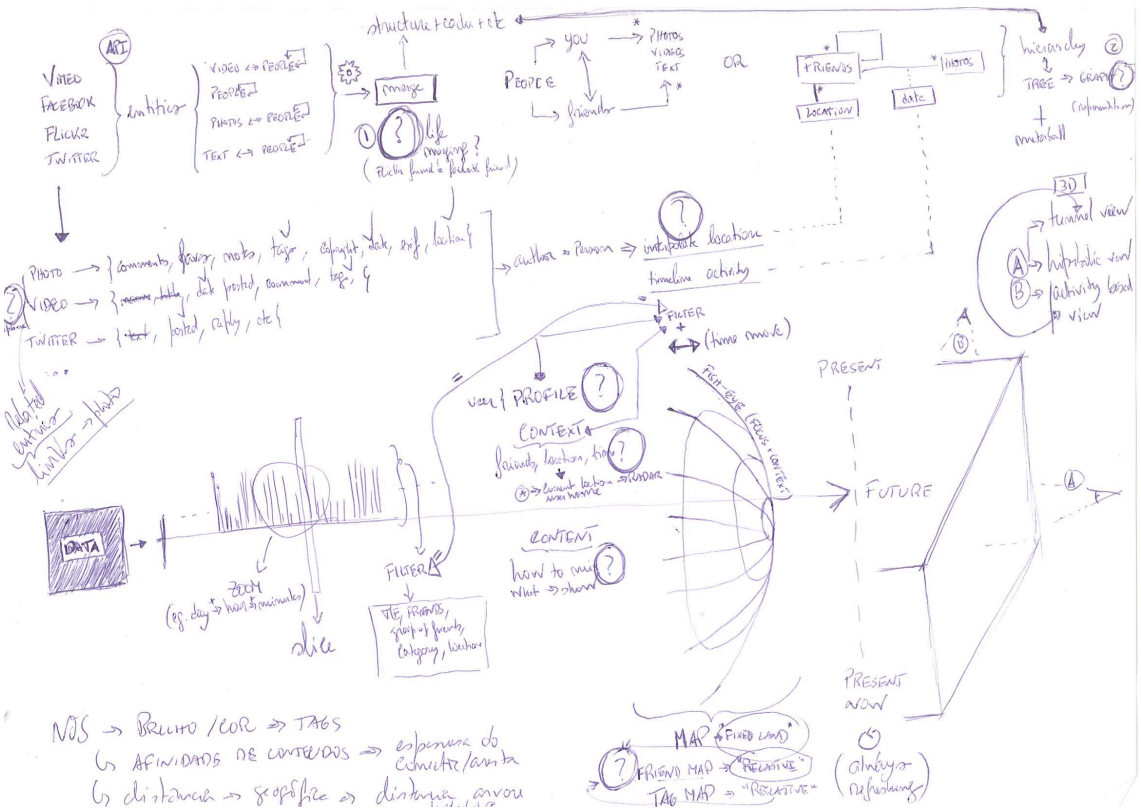


Figure D.1: Original sketch for the initial solution

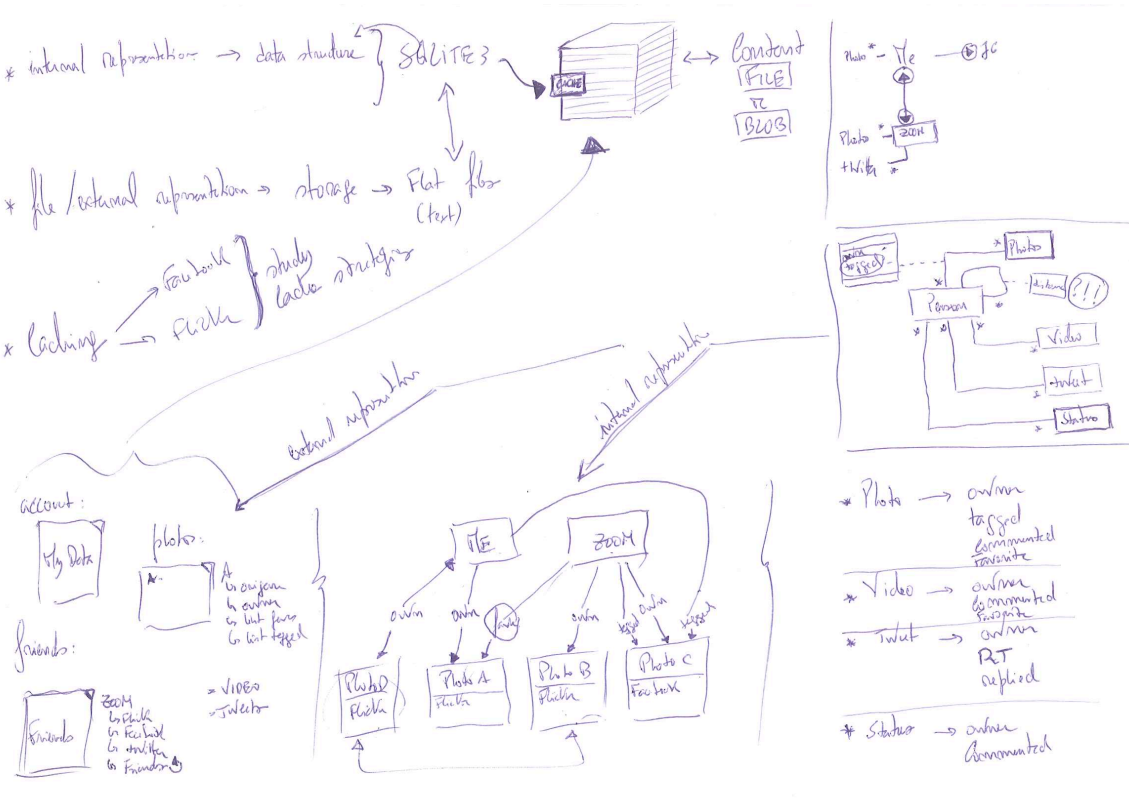


Figure D.2: Sketch for the data management solution