Ulla Sederlöf

Art Meets Technology – Creating Gigapixel Panoramas for Historical Sites

Louhisaari Conservation Documentation

Helsinki Metropolia University of Applied Sciences

Master of Engineering

Information Technology

Media Production Ecosystems

Master’s Thesis

28 March 2016

Helsinki Metropolia University of Applied Sciences

Author(s) Title Number of Pages Date	Ulla Sederlöf Art meets technology – Creating gigapixel Panoramas for historical sites Louhisaari Conservation Documentation 86 pages + 1 appendices 28 March 2016
Degree	Master of Engineering
Degree Programme	Information Technology
Specialisation option	Media Production Ecosystems
Instructor	Harri Airaksinen, Principal Lecturer

This thesis explores the possibility of using high-quality gigapixel sized, multiresolution panoramas as a way to study, access, present and share high-resolution documentary image material created during the Louhisaari Conservation Project.

The ceiling paintings in the Louhisaari Manor banquet hall have been subject to natural aging. To record and study the condition and the damage, the entire ceiling was photographed with digital macrophotography. Viewing and comparing these high-resolution images proved challenging: requiring a powerful computer with dedicated software, the possibility to download/transfer these files and most importantly – how to use all of these mentioned. To solve this problem this study focuses on finding out how this documentation material could be presented with a specially designed application utilizing panoramas and multiresolution technology, created with standard web technologies, making the material accessible for the common user.

Panorama application creation from the first steps of photography into building a fully functional application is a highly complicated process and understanding how these steps and the decision made along the way relate to each other’s makes the process more tangible. The approach in this study discusses the current steps, equipment, technologies, concepts and approaches in capturing, stitching and rendering these multiresolution panoramas and the app. It focuses on retaining the highest quality and accuracy of the generated panoramas.

The technical implementation is a HTML5 / JavaScript / WebGL based client-side application designed for both desktop and mobile/tablet devices. A 360˙ multiresolution panorama of the hall is paired with information and additional high-resolution zoomable flat panoramas of 10 different allegorical figures and emblems depicted on the paintings. One of the essential components of the application is the comparison tool, specially designed to allow users to study different photography methods used during the documentation.

The results of this study show how multiresolution technology and panoramas can be used to present gigapixel sized, high-resolution documentary material, making the image data accessible for anyone to use, still retaining high-quality and great user experience.

Keywords	360˙, gigapixel-, multiresolution- and flat panoramas, WebGL

Tekijä Otsikko Sivumäärä Aika	Ulla Sederlöf Art Meets Technology – Creating Gigapixel Panoramas for Historical Sites Louhisaari Conservation Documentation 86 sivua + 1 liite 28.3.2016
Tutkinto	Master of Engineering
Koulutusohjelma	Information Technology
Sunntautumisvaihtoehto	Media Production Ecosystems
Ohjaaja	Yliopettaja, Harri Airaksinen

YAMK-insinöörityön tavoitteena oli selvittää voiko laadukkaita korkearesoluutioisia moniresoluutio panoraamoja käyttää Louhisaaren konservointiprojektin kuvamateriaalin jakeluun, tutkintaan ja vertailuun.

Louhisaaren päärakennuksen juhlasalin kattomaalaukset konservoitiin kesän 2015 aikana. Ennen konservointia maalausten kunto tutkittiin ja dokumentoitiin laajasti eri erikoiskuvausmenetelmin. Tutkimuksen aikana generoidun dokumentaatiomateriaalin katselu ja vertailu osoittautui haasteelliseksi, sillä kuvien koko ja laatu asetti niiden katselemiselle vaatimuksia: hyvän kuvankatseluohjelman, hyvän tietoliikenneyhteyden niiden siirtelyyn sekä ymmärrystä sekä taitoa käyttää ja katsella kuvia. Tämän ongelman ratkaisemiseksi tässä tutkimuksessa keskityttiin selvittämään voisiko panoraamoja ja erityisesti multiresoluutiotekniikkaa sekä yleisiä selaintekniikoita käyttää ja hyödyntää niin että kuvat olisivat helposti kaikkien saatavilla, käytettävissä sekä katseltavissa.

Panoraamoja hyödyntävän selainsovelluksen rakentaminen, kuvauksista valmiiseen sovellukseen, on monitahoinen ja monimutkainen prosessi. Prosessin yksittäisten askelien ymmärtäminen sekä sen aikana tehtyjen päätöksien vaikutus kokonaiskuvaan on hyvin keskeistä prosessin onnistumiselle. Lähestymistapana tässä tutkimuksessa tarkasteltiin panoraamojen generoimisen eri vaiheita kuten valokuvausta, kuvien liittämistä ja katselua sekä erilaisia laitteita, teknisiä vaihtoehtoja, käsitteitä sekä lähestymistapoja. Tutkimusprojektin keskeisenä teemana oli säilyttää panoraamojen korkealaatuisuus, tarkkuus sekä optimoida generoidut panoraamat käytettäväksi selainpohjaisissa applikaatioissa.

Tekninen toteutus on HTML5/JavaScript/WebGL -pohjainen sovellus joka mukautuu erilaisille päälaitteille. Sovellus on rakennettu juhlasalista generoidun 360˙ panoraaman ympärille sekä sisältää kattomaamaalauksissa esiintyvistä hahmoista generoituja tasopanoraamoja, joiden avulla käyttäjä voi tutkia dokumentointiin käytettyjä erikoiskuvausmenetelmiä.

Opinnäytetyön tulokset osoittavat että multiresoluutiotekniikkaa voidaan erinomaisesti hyödyntää gigatavujen kokoisien panoraamoja koostamiseen, katseluun sekä vertailuun sekä konservointiprojektin aikana tuotetun korkeresoluutioisen materiaalin jakeluun.

Avainsanat	360˙ panoraama, multiresoluutio, tasopanoraama, WebGL

Contents

1 Introduction

Recently, we have entered an era where the human kind produces more data and information than it is capable of storing, managing or utilizing. In the past three years, we have generated more data than ever before in the history of humanity. [1] Most of the people in the world have access to mobile phones, computers, sensors, tablets and other technology and are constantly producing data. And the rate, at which we are producing it, is ever growing.

Before the digital age, acquiring and storing data was far more tedious, time consuming and expensive. For example, photography has been a chemical process, where images have been captured to a photosensitive film or photographic plates. To take a shot, one had to carefully plan it in advance; there were no means of erasing and retaking the shots as we have in digital photography.

Digital media has made the process of acquiring data much more simple and the process of taking and sharing photographs is no exception. Although, the technique involving traditional analogue photography does not differ from digital photography that much, there is one fundamental difference: The significance of physical media to store the image data. In traditional analog photography, the physical medium is everything, but in digital media this can be discarded, as digital cameras use image sensors that convert optical images into electronic signals. These signals and the resulting image data are in turn captured on rewritable storage cards and other medium.

As technology develops and the capabilities of digital devices such as sensors skyrocket people have access to digitized information that was previously unavailable. The ability to capture extremely detailed high-resolution images changes our relationship to images and the rich information they contain. High-resolution digital photography combined with digital imaging techniques, such as IR photography provide us new ways of exploring and studying objects, such as paintings. High-resolution imagery is now widely adopted as a technical study method for art conservation and restoration. [2]

While panoramic photography has been around for a while, the capability of producing high quality, detailed images has also brought new insight into creating high fidelity panoramas. Ultra high-resolution panoramic views can be composed of hundreds of high-resolution images stitched together resulting in gigapixel size image mosaics.

How are our devices capable of displaying gigapixel size imagery online? How can we process, store and share this and other captured data? Scientists and computer engineers have coined a new term for the phenomenon: “big data”. [3] It is basically a phrase to describe large volumes of date too big to process with traditional techniques. The challenges we face with high-resolution images go along with the characteristics of big data.

This study focuses on the key issues and the workflow in creating high-quality multi-resolution panoramas. It explores the possibilities of using these generated panoramas as way to study, access and share the high-resolution documentary image material created during the Louhisaari Conservation Project. The reference implementation is a HTML5 / JavaScript based client-side application designed for both Desktop and Mobile/Tablet devices.

This study does not explain the different tools and techniques of panoramic photography, nor explains in depth how the front-end web application was built.

2 Research Problem Background: Project Louhisaari

Louhisaari is a 17th century Baroque manor and the childhood home of C. G. E. Mannerheim. The main building, Figure 1, dates from 1655 and is one of the rare examples of palatial architecture in Finland. The festive floor and the service floor are in 17th-century style and furnished to match. [4] One of main attractions of the main building is the banquet hall on the 3rd floor. The ceiling of the banquet hall is filled with painted decorations.

Figure 1. Louhisaari Manor, main building

These painting have been subject to natural aging for hundreds of years and to prevent further deterioration conservation of the paintings was needed. In autumn 2013 an Interdisciplinary Research Project by Senate Properties, National Board of Antiquities, the National Museum of Finland and Helsinki Metropolia University of Applied Sciences was launched. The first part of the project aimed to study the materials and techniques used in the ceiling paintings, as well as to define the damage incurred to them over the centuries. At the time of writing, spring 2016, the project has been finished and the ceiling has undergone conservation. The festive hall will be opened for the public in the beginning of the summer 2016.

2.1 Introduction and History

Louhisaari manor has been the home for known historical families such as the Fleming- and Mannerheim -families, since the 15th century. The ceiling of the banquet hall, located on the 3rd floor is filled with painted decorations, as seen in Figure 2. A Swedish portraitist Jochim Langh carried out these decorations in the 1660s, soon after the building of the manor was finished. [5] In his work, he used expensive, rare colors and pigments and the end result was skilled with beautiful vivid, saturated colors.

Figure 2. Louhisaari Manor – 3rd floor banquet hall, furniture removed

As time has passed, these fragile paintings have started to fade and chip and to prevent further deterioration a conservation project was planned. Before the conservation was commenced, the condition of the paintings, materials and techniques used were thoroughly examined and documented.

The history of painting techniques is by nature a multidisciplinary area of study, combining research in science, conservation, and art history as well as specific expertise in paintings. [6] The joint Research and Conservation project started fall 2013 and the paintings were conserved during the summer 2015. The project consisted of four main parts: an assessment of the current condition of the paintings, technical documentation, conservation treatment and the deployment of the technical solution to present the documentation material.

Before conservation treatment, the over 100m² large ceiling was thoroughly photographed to record the condition and the visible damage. Lecturers Mika Seppälä and Heidi Söderholm from Metropolia UAS, with the help from students from the digital communication and conservation studies documented the paintings with digital macrophotography. To record and study the condition and the damage as accurately as possible the entire ceiling was photographed with digital macrophotography in visible, ultraviolet and infrared ranges of the electromagnetic spectrum and other special lighting setups.

The Louhisaari Application project began in May 2014 with a meeting with Mika Seppälä. He had started the scientific documentation process earlier and was now looking for a solution for publishing his documentation content. He was hoping to find a new and innovative way to view the content so that the material would be easily accessed and one or multiple high-resolution images could be compared digitally and in real-time. I addition he was interested on studying how 360˚ panoramas could be used as documentation material.

For an area as large as the ceiling in Louhisaari, documenting the paintings in a tiled manner, small areas in a single capture at a time, proved to be time consuming. During the documentation phase he had been developing these methods to be as cost effective as possible, still retaining the high quality of the images. One of these promising methods was panorama photography.

He was also facing the fact, that these high-resolution, vivid, detailed files were huge. Most of the researchers working with the files had problems even opening them in their desktop computers. So how could the publication of the documentation material be made better? And how could both the researchers and the public have easy access to view and utilize these images? Could panoramas maybe act as a starting point when searching for a specific detail or figure in the paintings? And could a panorama be used to access high-resolution images of the figures paired with additional data?

A few panorama test shootings had already been organized and the results seemed promising. After the first meeting with him, we decided to continue. Mika Seppälä focused on continuing his study on finding the best approach for the panorama photography methods in Louhisaari in addition to finding the right equipment to work with. The focus was on capturing these panoramas with the best possible quality available.

For me, the focus was more on the technical implementation and utilizing the image data, building both the panoramas and the technical solution for viewing them. Therefore the topic of this thesis, how “art meets technology” was really the focus, the heart of this project. We wanted the public to have access to this beautiful art, through commonly used technologies provided and to present what kind of information these special photography methods really give about the paintings. Some methods can reveal even sketches hidden underneath the painted layers while other methods show how the ageing process has really resulted in the flaking and chipping of the paint.

The technological solution needed to solve the following questions/problems:

- The application needed to be built with accessible technologies
- Include a virtual reconstruction of the site, preferably a panorama
- Present high-resolution images, with the possibility to compare them
- Solve the problem of physical inaccessibly during the conservation
- Solve the problem of huge image data, bandwidth, storage and processing

Based on these ideas a consensus was found. The first phase was to create a consumer-oriented platform / application to accommodate the fact that the hall was inaccessible due to conservation. This way the visitors of the manor could have an opportunity the see and study the banquet hall virtually. After the conservation work would be finished, the application would work as an additional tool for the guides. With the application, the guides would be able to show the visitors details of the paintings, not visible for the human eye or zooming into details otherwise too small to distinguish when viewed from the ground level.

We felt that our approach, based on the work of multiple disciplines, could contribute significantly to the field of conservation and public awareness of the project and the hall itself.

2.2 Louhisaari Conservation Documentation - Emblems and Allegorical Figures

The ceiling in the Louhisaari banquet hall covers an area of over 100 square meters and is covered with beautiful paintings showing landscapes, birds, flowers and different figures. The figures included in the ceiling paintings, see Figure 3, represent a collection of allegorical figures typical of the era.

Figure 3. A detail of the Louhisaari ceiling paintings [7]

From Jouni Kuurne, the National Museum of Finland, excerpt from the Louhisaari Application:

> In Renaissance Italy, several guides were written on how to describe emotions and other abstract things in visual terms – often in human form. Thanks to the recently reawakened interest in classical antiquity, many writers compiled different interpretations of the meanings of figures or emblems. The best known of these

theoreticians was Cesare Ripa, who hailed from Perugia. His work on allegorical figures, Iconologia overo Descrittione Dell'imagini Universali cavate dall'Antichità et da altri luoghi ("Iconology, or a description of universal figures from antiquity and other sources"), was published in 1593. [8]

In search of his allegories, Cesare Ripa drew inspiration from Roman coins of the classical period, which became a source for some of the most important emblems. A profile picture of e.g. an emperor usually adorned the front side of coins, but their reverse side featured various types of images – often personifications, or symbols for things or attributes presented in human form. In classical art, different kinds of allegories were used to signify the real or purported power of a ruler, or attributes one either wished this person possessed or ones one would want people to believe in. [8]

The figures included in the ceiling painting at Louhisaari, represent a collection of allegorical figures typical of the era, symbolizing various kinds of personal virtues or virtues of the state. Of course, the underlying idea was that these virtuous characteristics or practices would help society, and thus the individual, to prosper. [8]

Coordinated by Mika Seppälä and Heidi Söderholm, together with the help of students, the entire paint surface of the ceiling was fully documented and photographed in small sections. Adapting special documentation methods, typically used with conventional smaller paintings, to the whole ceiling proved challenging. The photographed area was large and some of the beams were heavily collapsed and curved. Therefore new methods for shooting were innovated and created by both the students and the teachers. [9]

To provide uniform shots with each of the documentation methods, the shooting was made with a camera dolly and a modified tripod that held up to four individual cameras. Each camera had its own special flash connected. With this special mount, see Figure 4A, it was possible to take individual shots from the exact same position of the ceiling, with both normal and infrared, ultraviolet and ultraviolet fluorescence lighting, see Figure 4B.

Figure 4. Special tripod mount A and ultraviolet fluorescence shooting B [9]

The first panoramas were shot from the center of the hall also with camera and lighting setups for normal, infrared and ultraviolet photography. These first panoramas were not as high resolution as the later ones, which were shot with a better camera setup, from both the center and the corners of the room.

Since the roof was fully documented, the material for the flat panoramas of the figures, used in the application was already shot. But the early trials to shoot the panoramas proved to be insufficient in quality and additional shootings were planned and later implemented.

See: Appendix 1 for more details about the figures and the special photography methods.

2.3 From Project Goals into Application Requirements

Big data is a broad term for data sets so large or complex that traditional data processing applications are inadequate. Challenges include analysis, capture, data curation, search, sharing, storage, transfer, visualization, querying and information privacy. [10]

At the very basics, it is just a term, a buzzword that essentially describes the enormous amount of information generated by consumers. World information is doubling every two years and the rate is ever growing. [11] The term, big data is quite unspecified, vague, since no one can clearly pinpoint what is big data and what is not. Big data is not a technology, but represents the ability to process large amounts of data-sets beyond the ability of commonly used software tools to capture, curate, manage, and process within a tolerable elapsed time. [12] Big data looks for techniques not only for storage but also to extract information hidden within. [3] Unstructured data is a term used for data that does not follow a specific format for big data, such as video, image, text and audio. [13] Although big data is typically coined with structured data-sets of even petabytes in size, it is still evident, that the problem we were facing with the Louhisaari documentation data fit in well within the characteristics of unstructured big data.

The ability to capture extremely detailed, high-resolution data has truly changed our relationship to images. High-resolution digital photography combined with digital imaging techniques such as IR photography provide us new ways of exploring and studying objects, such as paintings. But it also brings us challenges.

Digital photography in conservation documentation, especially special imaging methods have so far been in use of a small group of experts. This is probably due to the fact that the examination and the comparison of these images are difficult. It is fairly hard to distinguish small details from a traditional print. [14] Also, while it is possible to use digital images and even compile large image mosaics from individual digital images, comparing the images in real-time has it's own limitations: this requires a powerful computer with dedicated software for viewing the files, the possibility to download/transfer these files and most importantly – how to use all of these mentioned

Since the whole ceiling was shot in a section by section manner, with hundreds of high-resolution images, it would be clearly impossible to view the combined image of the whole roof with commonly used software tools. The seamless stitched, high-resolution image of the Louhisaari ceiling would result in an image mosaic, up to even hundreds of gigapixels in size. Even the single shots were quite large in size, up to 10 MB, that the common user, with adequate hard- and software faced issues with loading, managing and handling them.

Handling this big data, big photography as we might call it, became our main focus and challenge to solve that steered and led the project all the way to the finished application. Some of the questions leading this project were: How can we make the process, storing and sharing this documentary data intuitive, easy and accessible for all? Are our devices capable of displaying this data and how? Can we use this imagery online? And most importantly, how can we show the rich information they contain?

To solve this problem an intensive study was made to find out how this imagery could be presented with standard web technologies, making it accessible for the common user. The characteristics of big data helped out determining where to focus, thus forming our goals.

Search

The provided application could help the user in searching specific figures or images in the ceiling.

Acquisition
Panorama photography could make the documenting of the paintings faster and cost effective.

Storage
An easy to use online storage could be used, accessed with an intuitive interface, the application.

Distribution
High-resolution images could be accessed with multiresolution technologies.

Analysis
The application could provide an easy tool to compare the high-resolution data.

Visualize
The application could make it possible to visualize the physical space and use it as an interface for the distribution of the more detailed images.

Focusing on the facts addressed above, the aim and goal of this project was outlined to:

> Solving how to access, utilise and present the ultra-high resolution images generated during the scientific documentation phase.

After many conversations and brainstorming sessions, one possible solution to this problem arose. Instead of relying on traditional image viewers a more appropriate solution would be e.g. a functional tablet/desktop application which allows viewing and comparison of images directly form the network, but still in high-quality. The application could provide the users 3-dimensional panoramas and other contextual information of the space, the banquet hall. The user of the app could place him/herself to the physical space in a way that the banquet hall panorama could work as an interface to access more detailed high-resolution images of specified figures and details in the painting.

Creating digital panoramas and wrapping the panoramas in an application is a lengthy process with multiple steps involved, as seen in Figure 5. Choices in the data-acquisition phase, the panorama photography, reflect and affect the later steps of the project such as generating the panoramas and the application and vice versa. Each of these steps are explained in the following chapters. In order to successfully carry out the project, project goals were carefully established.

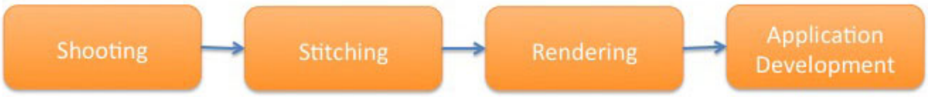


Figure 5. Phases in creating Panoramas

As stated earlier in Chapter 2.1 Introduction and History, the application needed to solve the following issues/problems now paired with a possible goal, see Table 1.

Table 1. Project Goals and Requirements

Problem	**Goals - Requirements**
Virtual reconstruction of the site	360 panorama and flat panoramas
Comparing high-resolution images in real-time	Specially designed comparison tool
Physical inaccessibility of the space	The application
Use of accessible, easy to use technologies	Browser based, web standards
Big data, gigapixel sized images	Multiresolution technology
Long term storage or archive	CDN, Online storage

As the project goals and the idea of the application was formed, so was the need for the panoramas:

- One ultra high-resolution 360˚ x 180˚ panorama of the banquet hall, for the core, the heart, of the application. Shot with visible light.
- Additional 18 flat panoramas, two for each chosen figure: one with visible light and one demonstrating a documentation method used
- One flat panorama showing a detail from the ceiling where the Louhisaari Manor is depicted

Although only one ultra high-resolution 360˚ panorama ended up in the final application, 360˚ panorama material was shot from 5 different locations inside the hall (from the center and the corners of room) for further use and development..

3 Louhisaari Application

The Louhisaari “Juhlasalin Kattomaalaukset” Application is an advanced technical solution that allows users to access a virtual three-dimensional reconstruction of the banquet hall. A 360˚x180˚, high-resolution, gigapixel sized multiresolution panorama of the hall is paired with information and additional high-resolution zoomable images of 10 different allegorical figures and emblems depicted on the paintings. In addition to the 360˚ panorama, one of the essential components of the application is the comparison tool, specially designed to allow users to study the paintings of the figures with high-resolution detail.

The comparison tool provides the users a possibility to examine the paintings by placing two multiresolution panoramas of the figures side-by side or blending them together for comparison. One image shot with visible light is always paired with an image shot with a different documentation method. This tool and further details of the figures can be accessed both from the 360˚ panorama or a figure gallery, see screenshots from the application in Figure 6.

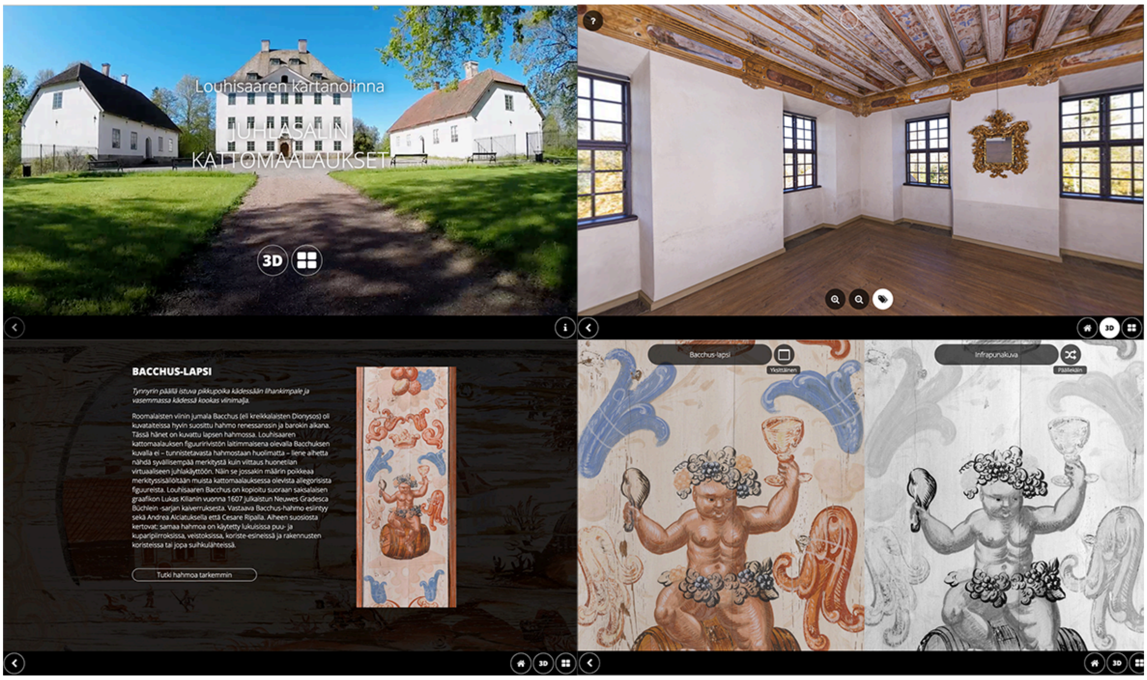


Figure 6. Screenshots from the application

The Louhisaari App is based on standard web technologies (HTML5, CSS, JavaScript and WebGL) and can be used and accessed by anyone using a browser that supports these technologies. In the heart of the application is the krpano HTML5 viewer (www.krpano.com), which is an HTML5/JavaScript, based client-side application that uses either the HTML5 CSS 3D Transforms or WebGL for displaying panoramic images directly in the Browser. [15]

From the user's perspective, the Louhisaari App is a browser-based application that works without the need to install plugins and on most of the modern browsers on almost all platforms and devices. The App is designed with usability first and a lot of detail has been put into user-centered design (UCD), information architecture and application logic to make the application usable, easy, accessible and effective for anyone to use. More about the different phases of the design and the workflow are explained in detail in the upcoming chapters.

The first version of the project was finished and released in August 2015, after a year of active development. This version was in test use on a tablet that mirrored the screen on a large display, in Louhisaari for a few months, as long as the manor was open during the summer season. Louhisaari is closed during the winter period. Since the first release, the application has been developed and optimized for online use, as well as language options and other minor fixes made.

At the time of writing, spring 2016, the text content of the application is in translation and as soon as these translations are done and implemented into the application, the online version of the application is ready to be published.

3.1.1 Target Users

User-Centered Design (UCD) is the process of designing a tool, such as a website's or application's user interface, from the perspective of how it will be understood and used by a human user. Rather than requiring users to adapt their attitudes and behaviors in order to learn and use a system, a system can be designed to support its intended users' existing beliefs, attitudes, and behaviors as they relate to the tasks that the system is being designed to support. [16]

The application has been developed to meet the needs of four specific user groups: the visitors of Louhisaari, guides working in Louhisaari, researchers studying the paintings and lastly the general public using the application online.

Since the paintings were going to undergo conservation during the summer 2015, this meant that the hall would be closed and inaccessible to the public. Therefore the first phase of this project and the application design concentrated in creating an application for a mobile device, preferably a tablet and the screen would be mirrored on a big display in the main lobby. The visitors as well as the guides could then showcase and have access to the hall and the paintings virtually.

There was also another side to the mobile first design idea. Since the ceiling is very high, visitors looking at the ceiling are not able to separate the small details easily provided by the high-resolution images with excellent lighting conditions. After the conservation of the paintings would be finished and the hall open for public again, the application could work as a tool for the guides. When entering the room and showing & explaining the meaning of the emblems, the guides could carry a tablet device, containing the application. The device could be used as a reference for extra information and zooming in and showing these details.

Thirdly, the application was designed to act as a prototype for a tool where the researchers are able to study the figures in the emblems digitally and with high-resolution. Therefore the comparison tool was designed and built to act as a prototype for a future application, dedicated for research use.

At the time of writing the development of the application is continuing with the fourth user group in mind: the general public. Until now, the application has been in use in the Louhisaari premises and only an unofficial online version has been provided for demo purposes. The launch of the official online web app version is due in May 2016. This launch requires multiple language options provided as well as optimizing the performance of the application to ensure the users to have the best experience with highest possible performance.

3.1.2 Production Phases: from Physical Space into Digital App

Panorama application creation from the first steps of photography into building a fully functional app is a highly complicated process. The aim of this thesis is to open up the process and explain the underlying concepts and steps needed to create beautiful high-resolution panorama applications. As a process, it is by far too comprehensive and large to explain in great detail, but mastering the basic steps and understanding how these steps and decisions relate to each other's makes the process more tangible.

Step 1: Pre-Production - Concepts

-> Explained in Chapter 4: Panoramas

- History of panoramas
- Understanding the basic concepts of panoramas and projections
- Understanding how distortion affects different stages in production
- Different types of panoramas
- Multiresolution approach with panoramas

Step 2: Production - Shooting

-> Explained in Chapter 5: Panorama Creation – Capturing Digital Panoramas

- Panorama Creation: Methods, Techniques and Equipment
- Acquisition of 360˙ panoramas
- Flat panoramas

Step 3: Post Production - Stitching

-> Explained in Chapter 6: Panorama Creation - Stitching

- History of stitching
- Manual and automatic stitching and blending
- Algorithms

Step 4: Post Production - Rendering

-> Explained in Chapter 7: Panorama Creation – Rendering

- History of web3D and viewing panoramas
- Generating, rendering and viewing the panoramas
- Software used
- Multiresolution solution

Step 5: Application Development

-> Explained in Chapter 8: Louhisaari Application

- Application logic
- Using the 360˙ panorama as an interface
- Technologies used

Challenges and key concepts

One of the hardest parts of the whole panorama and application development workflow is choosing the right technology, equipment and the right software to put it all together. Panorama technology has advanced from the early traditional film photography into a market with multiple cameras and lenses to choose from, some specially designed for panorama photography. Static tripods, where the user rotates the camera manually have developed into panoramic automated and motorized heads. Software stitching the image mosaics into the panoramas and viewers, giving users the access to study the panoramas even by panning tilting and zooming have all matured and evolved. [17] Web Application developers also have a multitude of technologies, software and languages to choose from. Within each Chapter in this thesis, some of the key issues and choices are explained to help out the users to choose the right equipment and technology:

Creating panoramas:

- What panoramas are and how they can be categorized
- The difference between the types of panoramas and how they affect photography, stitching and viewing
- How to handle distortion in both shooting, generating and viewing the panoramas
- Cameras, Lenses and using a robotic panoramic head, how they all affect the final panorama
- Best possible projections and photo stitching solutions to preserve original painting data and high-quality
- Suitable panorama creation software
- Using multiresolution technology to preserve high, rich detail and accessibility
- Choices in web technologies

Also the final application had to:

- Use the panorama as a gateway through which to explore additional scientific data
- Work without the need of plugins
- To be optimized to work on all platforms
- Have the ability to work with gigapixel images

- Responsive and adapt to different devices
- Have a ability to work without a dedicated server
- Use standard file formats
- Be user friendly, usability first

Looking at the provided list above, it is obvious that the choices were made through intensive research and trial and errors to put it all together into a working, fully functional application fulfilling all of the above criteria.

4 Panoramas

Panoramic photography is creating landscapes, wide-angle views or even 360 degree views from multiple, overlapping images stitched together to create a seamless panoramic image. The word panorama is derived from the Greek words pan "all" + rama "sight". [18]

4.1 History and Definition

Panoramas, first became known when in 1787, an Irishman Robert Barker patented his plans for a cylindrical building that was to be erected around a large, panoramic painting. His goal was to produce the perfect illusion of a real scene. [19] This experience of immersion, the perception of being physically present in a non-physical world was created so that the viewer stood on a special platform, placed in the center of the circular room, walls covered with a panoramic painting, as seen in Figure 7.

Figure 7. A panorama of London by Robert Barker, 1792 [20]

A few years later his panorama building, a rotunda, Figure 8, was erected in Leicester square London and was in use for half a decade. The panoramic paintings in the exhibition changed every year or two and later were even reinforced with different 3d object placed inside the space.

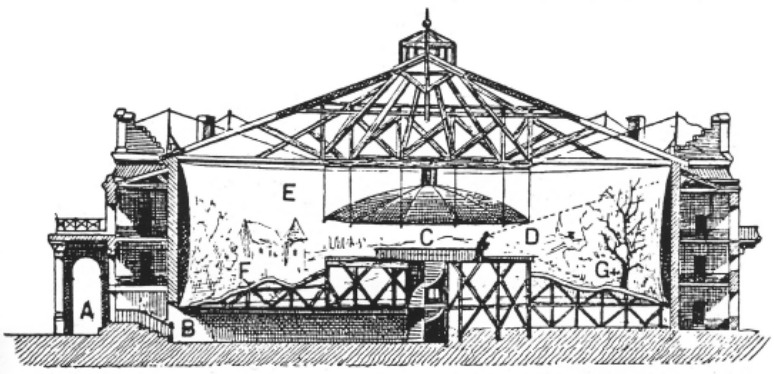


Figure 8. Diagram of a rotunda [19]

Around the mid 19th century, these panorama installations gained huge public awareness and popularity and similar panoramas were soon exhibited widely around the world. It is actually quite obvious why these panoramas gained such popularity. The people from that time did not have the ability to travel like we do now. Tourism and mass travel were terms unheard of, TV and radio did not exist and photography was taking its first infant steps. Through these panoramas, the public could visit places, otherwise not able to visit or reach, radically changing the way they viewed the world. The opening ceremonies of new exhibitions were important, anticipated social events.

Soon after the birth of photography around the 1830s, the first steps of panoramic photography were also made. At first the photographers were rotating their cameras to capture several parts of the scene and then trying to reassemble the single shots to a panoramic image. These early panoramas were made by placing two or more daguerreotype plates, in a side-by-side manner, see Figure 9. Daguerreotypes, the first commercially available photographic process, used silver- coated copper plates to produce highly detailed images. [21]

Figure 9. San Francisco, 1851, from Rincon Hill - digital file from intermediary roll film copy [22]

It was still quite evident that with the technology present these panoramas resulted often with visible seams, perspective mismatches and other deficiencies. Therefore this period saw early experiments with rotating cameras and lenses and curved focal planes to overcome these problems.

As time went on, especially after the introduction of flexible film in 1888, panoramic photography was revolutionized and may new and successful cameras were specially designed to create seamless panoramic images. [23] These special cameras were able to produce cylindrical panoramas by rotating either the lens or the camera evenly in time with moving and exposing a particularly long strip of film, see Figure 10. Similarly functioning short- and full rotation cameras are still used and manufactured today, but are highly specialized and expensive.

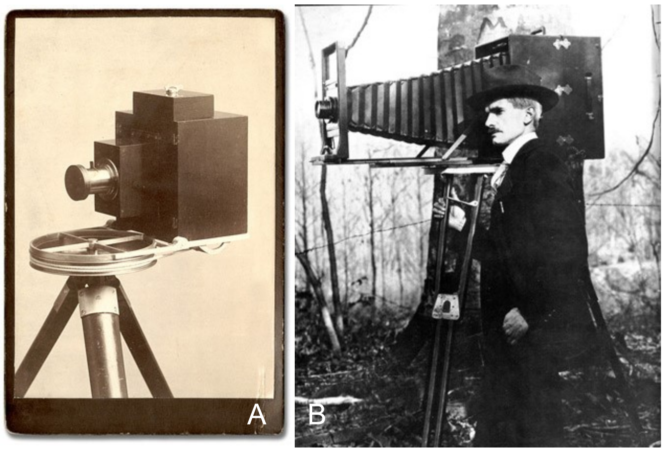


Figure 10. A: John Connon's whole circuit panoramic camera was the first camera to record 360 degrees of the horizon on a long roll of film. Patented in Britain and the United States in 1887, and in Canada in 1888. [24] B: A panoramic photographer with his Cirkut camera. Around 1910. [24]

Nevertheless, the most common form of panoramic camera was and still is the conventional fixed lens camera, also called the flatback. Fixed lens cameras have a flat image plane and as the name states, fixed lenses and are fundamentally different from ones mentioned above. The cameras expose the film in a single exposure and therefore rely on the use of wide-angle lenses or extended film planes.

The digital era has naturally brought advances to panoramic imaging. Just as the early pioneers in panoramic photography trying to compose panoramas from separate images, segments, we now have the possibility to do the same digitally. With digital photography we can take several of digital images, often with the help of a panoramic tripod head and combine, stitch them together with special programs and algorithms for composition. These panoramas can be flat, two-dimensional images, cylindrical images or even mapped onto a 3D space covering a horizontal angle of view of 360° and a vertical angle of view of 180°. [25]

Choosing the right technologies can be a daunting task for the photographer. There are always tradeoffs, no matter what the chosen method is: some of these are explained in the following Chapters. This study does not explain the advantages / disadvantages of traditional analog over digital technologies. It focuses in capturing multiple images with a conventional DSLR camera, together with a robotic panohead, which are then stitched together to create the 360˚ panorama.

4.2 Panoramic Image Concepts

The definition of a panoramic image greatly varies. From the early pioneers of panoramic imaging, the photographers have been trying to capture areas wider than the conventional photograph and human vision. Therefore usually an image can be called panoramic when it is showing a field of view greater than the human eye is capable of seeing. Panoramas have an aspect ratio of 2:1 or larger, the image being at least twice as wide as it is high. [26]

4.2.1 Field of View and Angle of View

When creating panoramas, it is crucial to understand how the field of view and the angle of view affect the final result. These are most definitely, one of the most important factors when choosing the right equipment to shoot, stitch and view panoramic images. The terms field of view (FOV) and angle of view are typically mixed, but are different things, as explained in Figure 11.

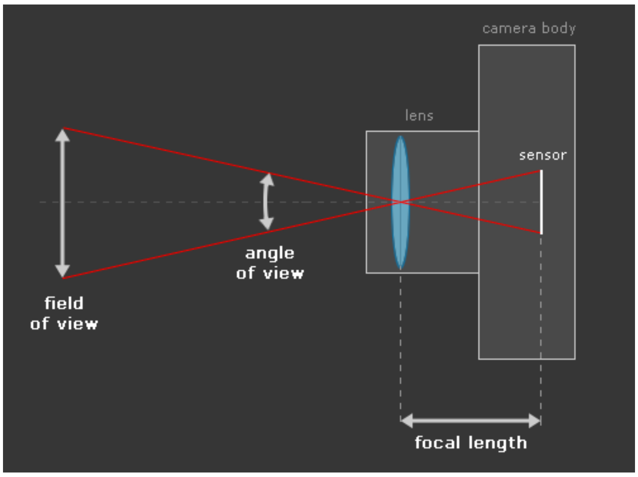


Figure 11. Field of view, angle of view and focal length (top view) [27]

Field of View = measurement of the area that is captured
Angle of View = determined by the cameras lens’s focal length + sensors size. It is the maximum angle the camera is capable of seeing.

Unfortunately our eyes are not as straightforward as cameras are. Although the human eye has a focal length of approximately 22 mm, this is misleading because the back of our eyes are curved, the periphery of our visual field contains progressively less detail than the center, and the scene we perceive is the combined result of both eyes. [28]

If the analogy of the eye's retina working as a sensor is drawn upon, the corresponding concept of the angle of view in human vision is the visual field. It is defined as "the number of degrees of visual angle during stable fixation of the eyes". Humans have an almost 180-degree forward-facing horizontal diameter of their visual field and the vertical range is typically around 135 degrees. [29] All of these issues have an effect on what we see and perceive as normal. Our eyes are great at adapting to different lighting and brightness situations and our mind can interpret and adjust the information in a way that we always see the best possible, distort free world.

So how does one capture and shoot wider fields of view that we are capable of seeing? And how can the 3D dimensional world around us be projected, mapped onto a 2D image. One way of thinking about the surrounding world is to place oneself in the middle of a huge sphere. Everything around can be then "painted" on the sphere. Then the 3D view is projected onto a 2D surface, this is called as: image or map projection, see Figure 12. Unfortunately, the mathematics of all of the known projections results in some distortion in the resulting image.

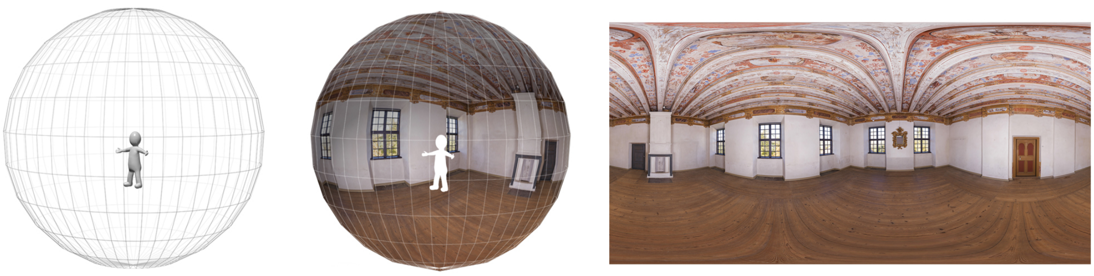


Figure 12. Mapping a 3D world onto an equirectangular mapped 2D image

The steps in creating these 360˚ panoramas are explained in greater detail in the following chapters. The workflow described follows closely the workflow in creating the 360˚ panorama on the Louhisaari banquet hall.

4.2.2 Distortion

In photography, there are two kinds of distortion: optical distortion, which is caused by the design of the lenses and perspective distortion, which is caused by the position of the camera to the subject. Distortion makes the photographed object or world seem unnatural. [19] Also projecting a location from a round surface, as the 3D world surrounding us, to a flat plane causes distortion. [30] Therefore the biggest challenge in panoramic photography, especially with spherical panoramas, is to shoot, assemble (stitch), project and view these multiple images to match the real world and the way we perceive it.

Optical distortion is generally referred to an optical aberration that deforms and bends physically straight lines and makes them appear curvy or deformed in images, which is why such distortion is also commonly referred to as “curvilinear”. [31] Figure 13 shows the two fundamental manifestations of the aberration, barrel and pincushion distortion. [32] Curvilinear distortion is typical in wide-angle lenses such as the fisheye lens.

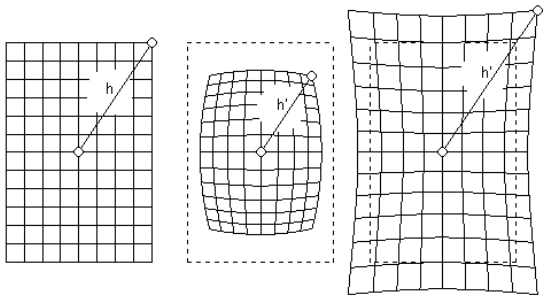


Figure 13. Distortion of a rectangular grid. Left: undistorted grid. Middle grid: barrel distortion. Right grid: pincushion distortion. [32]

Some photographic lenses are optically designed so that straight lines in a scene are represented by straight lines in the images. Lenses that achieve this effect are said to be rectilinear. [33] These lenses yield images where the physically straight lines, such as roads or the walls of buildings remain as such. In other words, it is a lens with little or no barrel or pincushion distortion. At particularly wide angles, however, the rectilinear perspective will cause objects to appear increasingly stretched and enlarged as they near the edge of the frame. [33]

Single shots with a narrow angle of view offer us usually a quite distortion free experience, but as we use wide-angle lenses with shorter focal lengths, distortion is evidential, as seen in Figure 14. This is caused by the fact that a 3-dimensional world is mapped to a 2-dimensional image. The bigger the viewing angle, the "more of the surrounding grid the user sees" and therefore the image gets distorted causing severe curving of straight lines.

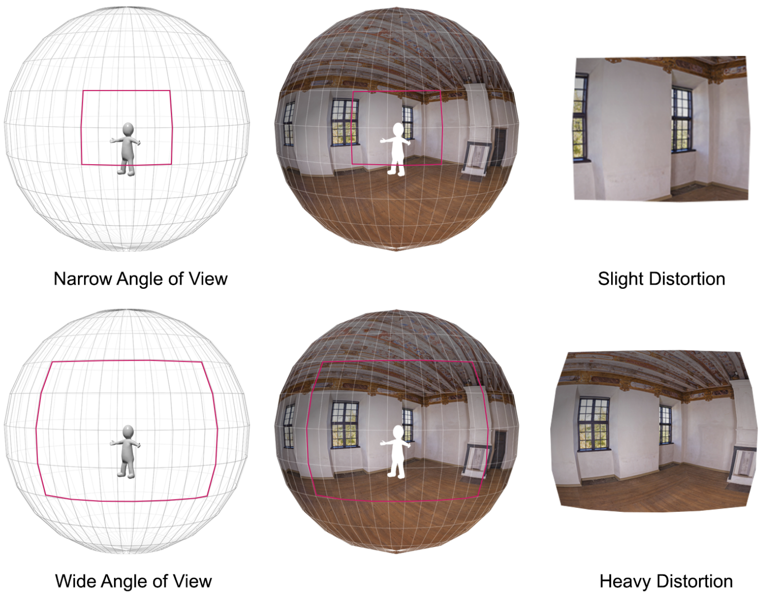


Figure 14. Angle of view comparison

In architectural photography, as in the case in the Louhisaari project, it is essential to keep the straight lines straight and avoid/correct lens distortion, as seen in Figure 15B. Therefore careful planning and implementation of the workflow creating and viewing these panoramas is required. Deep understanding of how different lenses and projections work is essential in all of the phases in creating and viewing different panoramas.

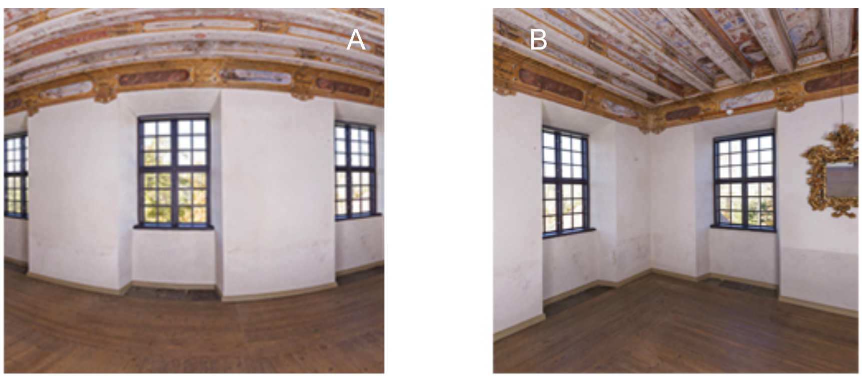


Figure 15. Curvilinear vs. Rectilinear projection

In panorama creation, after shooting the images, they will be stitched together to create the final image mosaic. Distortion in the lens can easily yield in poorly stitched panoramas, unless first corrected in image editing programs. Choosing lenses that have little or no distortion at all often result in better stitching quality. [34]

With perspective distortion parallel lines appear to converge in the image. It appears when the camera is not facing these parallel lines perpendicularly or that the camera is not pointed at the horizon. [30] Perspective distortion is actually a natural characteristic of the 3D world and our how we see it. Again, our brains are capable in correcting the distortion and therefore we do not perceive these lines as converging or unnatural.

The key concepts of projections and how they affect distortion are explained in the following chapters.

4.3 Panoramic Image Projections

To create a panorama of our surrounding world, the sphere as we may think of, it must be first projected onto a flat image. How the locations on the sphere, the 3D world are shown, projected on a flat image or a coordinate system is one of the biggest choices affecting the quality of the panorama. Fortunately, commonly used projections have been established. This chapter explains some of the most know projections and how they work.

Projections convert measured locations of things and events in three dimensions to two dimensions. Projections are important but also complicated because it is impossible using geometric or more complex mathematical methods to simultaneously preserve both the shape and the two-dimensional area of any three-dimensional object, when we depict it in a two-dimensional coordinate system. [30] All projections distort the surface in some fashion.

For example when creating a 360˚ panorama, the entire angle of view of the viewer (360˚ horizontal to 180˚ vertical sphere) is projected, flattened and stretched onto a 2-dimensional plane, the image. The 2-dimensional image shows both vertical and horizontal distortion. When this image in turn is viewed with special panorama viewing software, the flat image is actually warped back into a sphere and can be looked at without any unnatural distortion.

4.3.1 Planar / Rectilinear Panoramas

A planar/rectilinear panorama is either a single wide-format image or a flat-stitched image, sometimes also called as a flat panorama. It is the normal projection our eyes are used to and generated by conventional cameras and rectilinear lenses. [29] In rectilinear images, the straight lines in the 3-dimensional world appear also straight in the image. Rectilinear panoramas can be viewed with standard image viewing software since the panoramas are just images displayed on a flat plane, as seen in Figure 16.

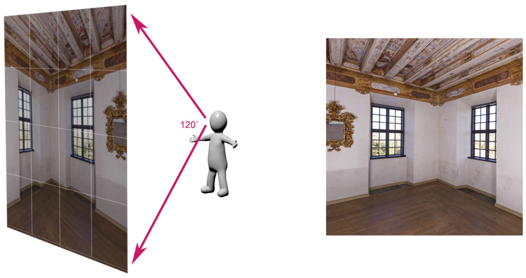


Figure 16. Flat rectilinear image, horizontal and vertical angle-of view 120˚

These panoramas are typically limited to a max 120˚ angle of view. This is due to the fact that as the angle increases the projection stretches the objects near the edges of the frame as demonstrated in Figure 17. This is because straight parallel lines are supposed to converge at infinite distance.

Figure 17. Rectilinear panorama - distortion visible when the angle of view exceeds 120˚

This effect can be quite disturbing and not typically desired in architectural photography. However it is possible to create panoramas up to 180˚ x 180˚ if the image is first manipulated and the stretched areas compressed, so that the distortion is less visible. Usually, if the angle exceeds 120˚, one of the other projections further explained should be used.

Multiresolution, rectilinear images are a special case of panoramas and are explained in Chapter 4.4 - Multiresolution panoramas. Also equirectangular images are typically warped into ‘normal’ rectilinear perspective when displayed with a special viewer program. Many viewers show a maximum 120˚ portion of the scene in rectilinear format. This does not mean that the original image used has been a rectilinear image. This image based rendering method in explained in detail in Chapter 7 - Panorama Creation - Rendering.

4.3.2 Spherical 360˚ Panoramas

A spherical panorama shows the entire 360 degree field of view horizontally and 180 vertically. When a spherical image is displayed as a flat image, it looks distorted. Just as with cylindrical panoramas, these panoramas are intended to be viewed with specialized software and players.

To create a panoramic picture, covering the full 360-degree field of view, multiple images are always needed. Even using the widest wide-angle lens, the tripod that the camera is placed on will be visible in the shot. To create 360˚ panoramas with the least number of photos (using a digital camera + wide-angle fisheye lens + panoramic head and a tripod), only a few images are needed but this naturally results in a lower resolution panorama. Higher resolution panoramas involve more camera shots, thus better resolution and less distortion. These images are then combined, stitched together to create the panorama. This process is explained in Chapter 6 - Panorama Creation - Stitching.

At the heart of these 360˚ panoramas is the equirectangular projection, as seen in Figure 18. Many spherical panorama viewers (applications that allow to interactively look around, up and down in a panorama) use equirectangular source images and the 360° x 180° equirectangular projection has become a standard for exchanging spherical panoramas between applications. [35]

Figure 18. Louhisaari Equirectangular projection image

The equirectangular projection maps the longitude and the latitude of the sphere to the horizontal and vertical coordinates of the image, called as Cartesian coordinates. [30] This projection is often seen in maps showing the earth. The geocentric latitude of a point on the surface of the earth is the angle (φ) between the plane of the equator and the line connecting the point to the center of the spherical earth. Since the plane of equator is the reference plane for measuring latitudes, equator is at 0°. The longitude of a point on the surface of the earth is the angular distance (λ) east or west from a reference line called prime meridian that runs from North Pole to South Pole, see Figure 19. Being a reference line or origin for measuring angles, the prime meridian has a longitude of 0°. [36]

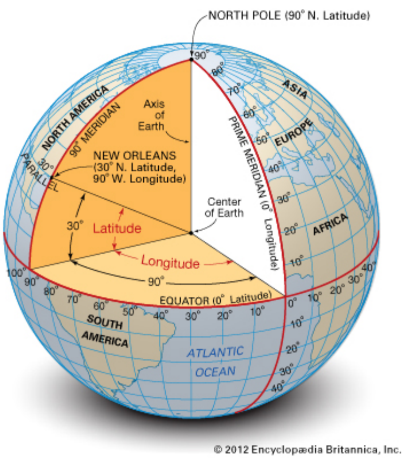


Figure 19. Cutaway drawing showing longitude and latitude [37]

Latitude and longitude coordinates specify positions in a spherical grid called the graticule, see Figure 20. The simplest kind of projection, the equirectangular projection as illustrated below, transforms the graticule into a rectangular grid in which all grid lines are straight, intersect at right angles, and are equally spaced. [36]

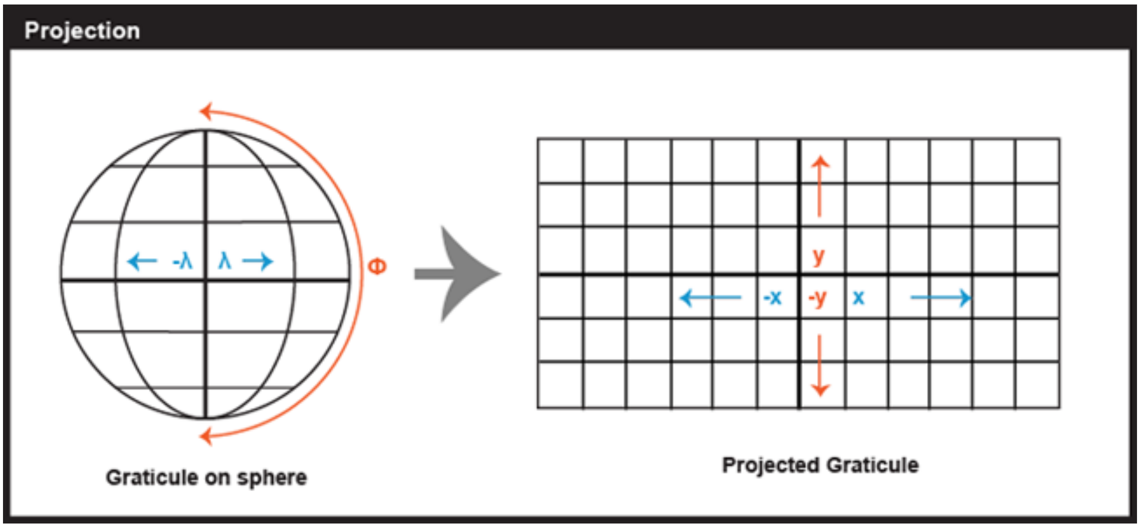
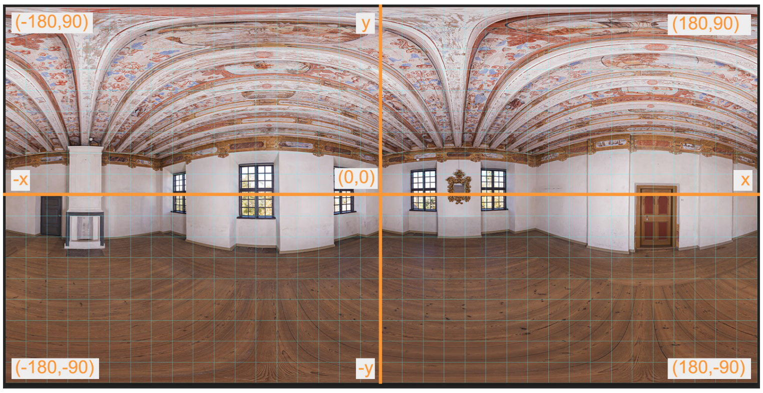


Figure 20. Map projections are mathematical transformations between geographic coordinates and plane coordinates. [36]

For example, with an image 360 pixels wide and 180 pixels tall, each pixel would represent one degree, as shown in Figure 21.



$$x = \lambda$$
$$y = \varphi$$

Figure 21. An enlargement of the equirectangular image showing the map projection

As seen in the image above, with the equirectangular projection the horizontal lines become curved but the vertical lines remain straight. The equirectangular projection is also called "unprojected", since the horizontal coordinate is simply longitude, and the vertical coordinate is simply latitude. Equirectangular images yield significant distortion, since the areas located near the poles stretch to the entire width of the panoramic image, as seen in Figure 22. The advantage is that this unprojected image can be easily transformed into other projections if needed. [30]

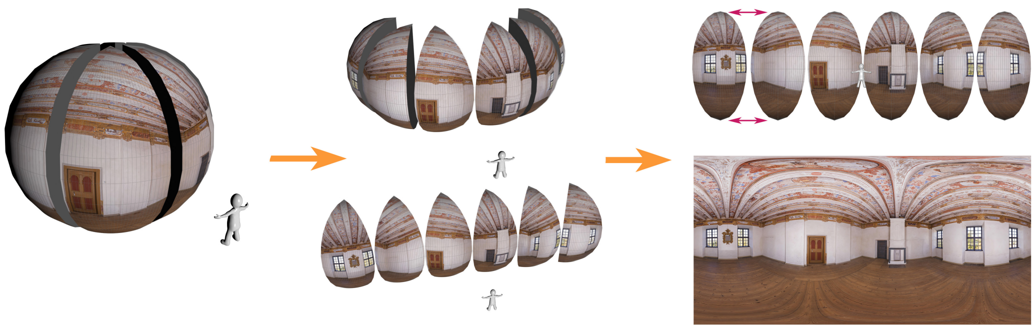


Figure 22. Equirectangular projection – from sphere to flat image

A common method of classification of map projections is according to distortion characteristics - identifying properties that are preserved or distorted by a projection. The distortion pattern of a projection can be visualized by distortion ellipses, which are known as Tissot's indicatrices. Each indicatrix (ellipse) represents the distortion at the point it is centered on. The two axes of the ellipse indicate the directions along which the scale is maximal and minimal at that point on the map, circular shapes indicating no distortion, Figure 23. [38]

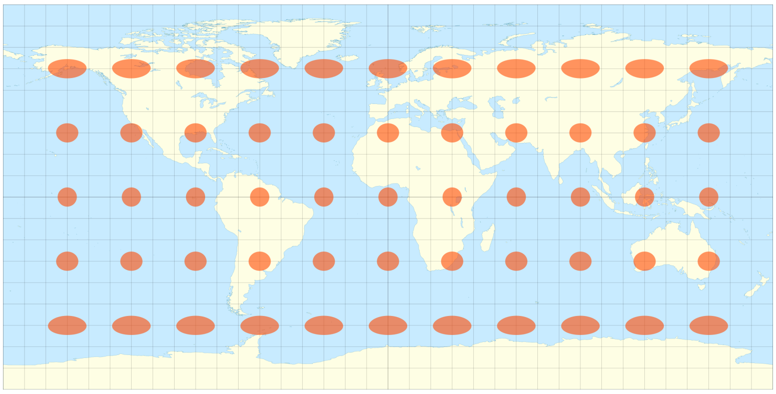


Figure 23. The equirectangular projection with Tissot's indicatrix of deformation [39]

Equirectangular panoramic images are intended to be viewed in a way that the image appears to be warped back into a shape of a sphere and viewed within. These panoramas should be viewed with special software that can display the wrapped image. A digital spherical panorama can also and is actually often converted to a series of rectilinear cubical images prior to viewing. Many spherical panorama viewers accept and use both equirectangular and cube strip images. Each of the images presents a cube face. Four of the faces cover the sides: front, right, back and left, see Figure 24. Two of the faces cover the zenith (top) and the nadir (bottom). Just as the other projections, the cube should be viewed from the center.

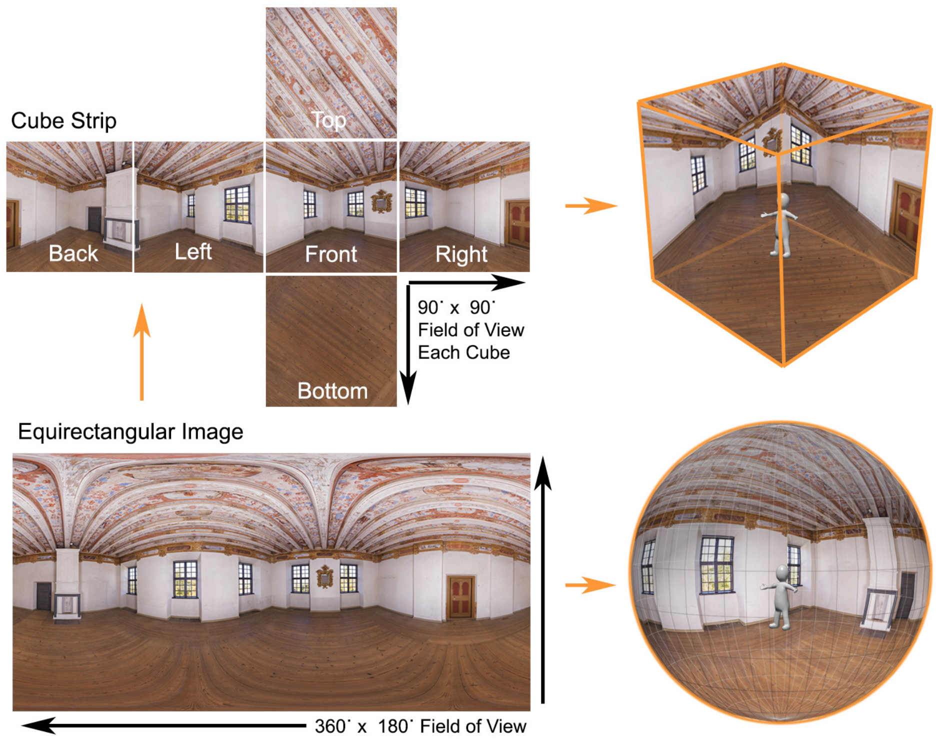


Figure 24. Spherical and Cubical mapping

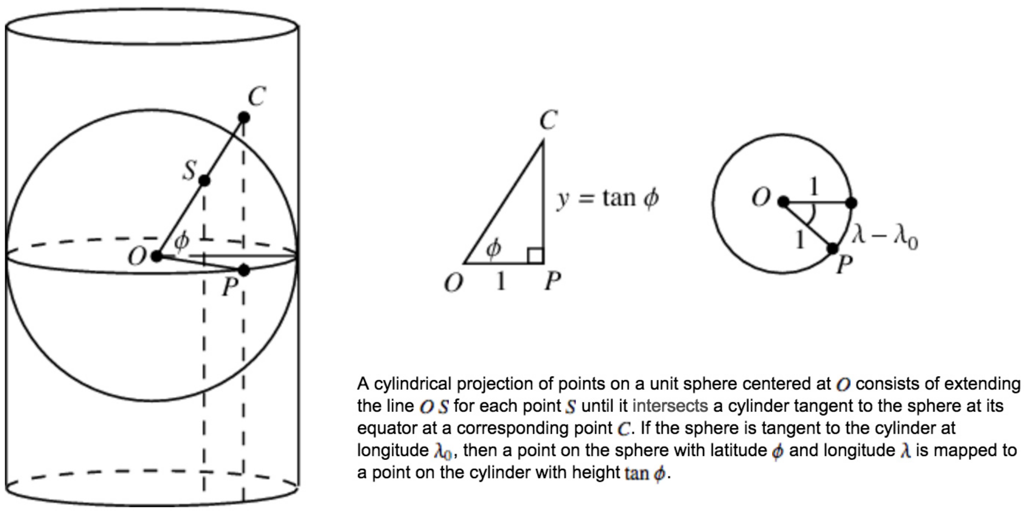
It is possible to extract cubical images from equirectangular images and vice versa. Since each cube face is in rectilinear format, they are good for editing. Equirectangular images have a very large amount of data redundancy near the poles because they are stretched in the 'latitude' direction. Downsizing the images can cause strange artifacts later when viewed in panorama viewers and therefore it is recommended to convent the image to a cubic projection before editing, and later switching back if necessary. [40] In the Louhisaari panorama, the tripod showing in the nadir was removed with this method. The equirectangular image was first converted into six cube faces. Then the nadir cube face was edited to remove the tripod and the cube faces were converted back into the equirectangular image for further use.

4.3.3 Cylindrical Panoramas

Cylindrical Panoramas are intended to be viewed as if the viewer was standing inside a cylinder. These panoramas are similar to the early panorama buildings where the images were painted on the walls and the viewer was standing in the middle of the room. Cylindrical panoramas can depict a horizontal view up to 360˙ but are limited in the vertical direction, usually up to 120˙.

Looking at Figure 25, unwrapping and flattening out the cylinder gives the Cartesian coordinates, Formula 1. [41]

$$x = \lambda - \lambda 0 \qquad (1)$$
$$y = \tan \varphi$$



A cylindrical projection of points on a unit sphere centered at O consists of extending the line $O\,S$ for each point S until it intersects a cylinder tangent to the sphere at its equator at a corresponding point C. If the sphere is tangent to the cylinder at longitude λ_0, then a point on the sphere with latitude ϕ and longitude λ is mapped to a point on the cylinder with height $\tan\phi$.

Figure 25. Cylindrical Projection [41]

Looking at Figure 26, it is easier to understand why these panoramas have a vertical limit. The closer the objects in the image get to the zenith and the nadir (the poles of the imagined sphere), the more stretching and distortion occurs. At the poles and therefore at the top and the bottom of the flattened image, the stretching would be unlimited and therefore cylindrical panoramas are limited in the vertical direction.

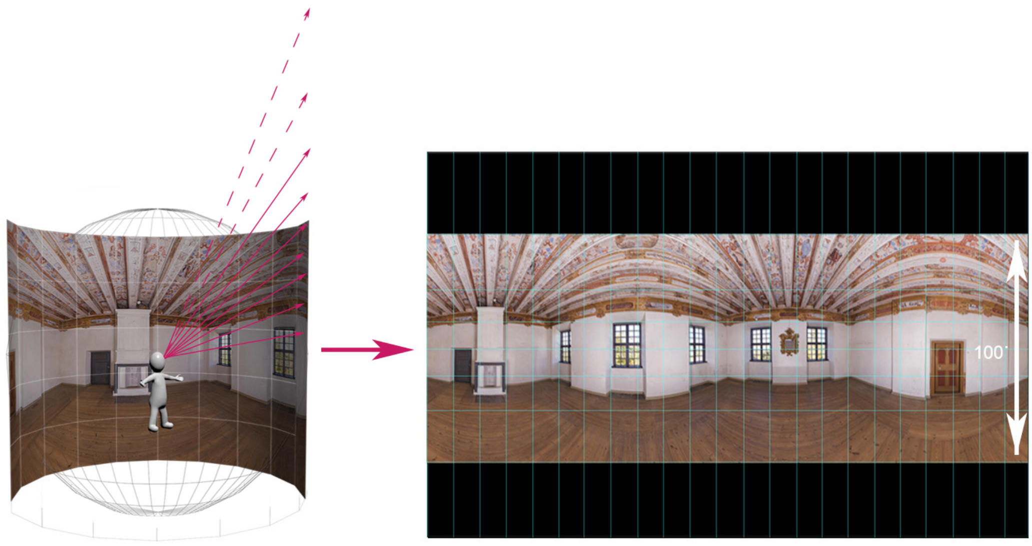


Figure 26. Cylindrical Panorama – angle of view, horizontal 360˚ and vertical 100˚

Cylindrical projection preserves the vertical lines and maintains more accurate relative sizes of objects than rectilinear projections. [42] However the straight lines, which are not vertical, become curved, just as with the equirectangular projection. The further the lines are from the center, the horizon, the more curvature appears.

If a higher vertical angle is needed, the Mercator projection can be used. It is quite near to the cylindrical projection, but offers less stretching at the top and the bottom edges of the frame. The Mercator projection is a good compromise between the cylindrical and the equirectangular projection, although not that often used in panoramas.

4.3.4 Other Panoramas

There are certainly a multitude of other panoramic projections and implementations. Nowadays is possible to create and view even panoramic videos and stereographic virtual reality panoramas. They actually base on the same ideas and projections as conventional panoramas. In panoramic video, the video frames are stitched into a equirectangular format for each frame and then displayed in a spherical panorama. WebVR uses just two separate synchronized panoramas, one for each eye. [43]

This study focuses on the formats and the projections closely related to the project and conventional panoramas.

4.4 Multiresolution Panoramas

Multiresolution technology allows the user to view large files without loading everything at once. This technology can be used for flat, cylindrical and even spherical panoramas. So far in this study, the concept of a panoramic image has meant a single stitched image. Since a panorama is generally a combination of multiple images, the file can quickly become huge. Typically, these panoramas are loaded at once into the memory of your device.

The stitched 360˙ equirectangular panorama generated from Louhisaari is an extremely large high resolution - full spherical panorama. The image is a 24-bit JPG image, with the following pixel dimensions (Formula 2):

$$57956 \text{ pixels} \times 28978 \text{ pixels} = 1.68 \text{ Gigapixels} \tag{2}$$

Since the image is a 24-bit RGB image, this means that for each pixel, 3 bytes of information are stored, one for each color. To display this image at once, a computer would need 5.04 Gigabytes of Memory, Formula 3. This is enough to freeze even the best performing computers.

$$\begin{gathered} 57956 \text{ pixels} \times 28978 \text{ pixels} = 1.68 \text{ Gigapixels} \\ 1.68 \text{ Gigapixels} \times 24\text{bits/pixel} = 40{,}3 \text{ Gigabits} \\ 40{,}3 \text{ Gigabits} / 8 = 5.04 \text{ Gigabytes} \end{gathered} \tag{3}$$

Still, this is truly the size of the panorama used in the app and this panorama can even be viewed on a mobile device. So how is this achieved?

The idea is similar to the satellite images and maps most of us have used online. As the user zooms in to the map, only the part of the map is loaded, that is visible with the current zoom level and the angle of view. With panorama technology the fundamental idea is the same; multiple image resolutions of the panorama image are used. Depending on the zoom level and angle of view, only the parts that best match the resolution are loaded. This means large amounts of data are never loaded simultaneously, and therefore less memory is needed. [44]

The krpano HTML5 viewer used in the Louhisaari app to view the panoramas is HTML5/JavaScript based. The viewer uses HTML5 CSS 3D Transforms or WebGL for displaying the panoramic images directly in the browser. The viewer currently only supports cubical and flat panoramic images, since with the HTML5 CSS 3D transforms it is only possible to transform flat html surfaces in 3D space. Therefore only panoramic formats with flat surfaces like the six flat sides of a cubical pano can be used. [45]

Spherical images, like the equirectangular image used in Louhisaari are actually converted into a cubical panorama when multiresolution panoramas are created. This is described in more detail in Chapter 7.4.1. - Louhisaari 360˚ Multiresolution Panorama.

When the user launches the panorama, the first level image is shown, see Figure 27 Level 1. This first level image can also be split in parts to speed up the loading but presents the lowest resolution image of the panorama. As the user zooms into the image, higher resolution tiles are loaded as the uses progresses deeper into the image pyramid, as seen in Figure 27, Levels 1-4.

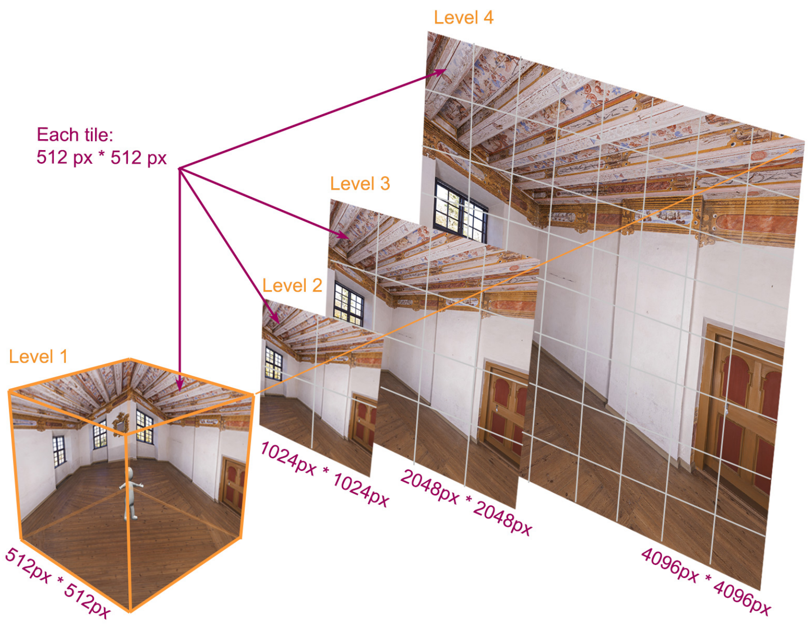


Figure 27. Multiresolution Tile Pyramid – 4 levels

Depending on the zoom level and the angle of view, only the needed tiles are loaded into the memory. If the view is changed, the loaded parts still remain in memory until the memory maximum limit is reached and therefore released (if not required for the current view). This approach allows for rapid access to the individual tiles in the pyramid with minimum server overhead or client side memory overload. The levels, tile sizes, angles of view when the levels are changed and even the memory limit for each panorama can be individually defined and adjusted to provide the optimal viewing experience.

There is also another great advantage in using multiresolution technology. Using different layers with multiple tile sizes of the image solves the problem when viewing the panorama with different size devices. Especially with small, low memory devices it would be impossible to load and view the Louhisaari 360˚ panorama. Now with multiresolution technology, when the screen resolution of the device is detected, the optimal resolution panorama for that specific display is loaded and the panorama is always shown with the best possible quality provided. Therefore multiresolution panoramas are optimal for applications that need to work with different devices. Not only this technology makes high-resolution gigapixel size panoramas reality, but also at the same time make the panoramas adapt to the users device, whether the user is using a smartphone, tablet or a desktop computer. This is essential when thinking about the usability and the user experience of the Louhisaari application.

The panoramas generated and used in the Louhisaari app all use the multiresolution approach. Both the 360˚ panorama of the banquet hall and the flat panoramas of the selected images use this technology to provide a way for the users of the app to study and explore these paintings in ultra high resolution, even with handheld devices.

At the time writing, spring 2016, only a handful of panorama software were able to handle multiresolution panoramas, here is a list of a few alternatives, Table 2.

Table 2. Panorama software

Pano2VR by GardenGnome software	ggnome.com/pano2vr/
The krpano viewer	krpano.com
EP-Sky by EasyPano	sky.easypano.com
Pannelum Viewer	pannellum.org
PanoTour Pro by Kolor	www.kolor.com/panotour/
GigaPan.com (requires to upload the panorama to their server)	gigapan.com

For this project the krpano viewer was chosen. This viewer works also in the core of many other commercial products. The krpano viewer can be easily embedded into html pages and almost everything in the viewer is customizable which was the most important reason this software was chosen over others.

5 Panorama Creation – Capturing Digital Panoramas

This chapter focuses on capturing and shooting digital 360˚ and flat panoramas with examples from the Louhisaari Project, see Figure 28.

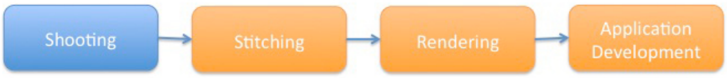


Figure 28. Panorama workflow - Shooting

There are multitudes of methods, ways and techniques in creating panoramic images. Panoramic images can be captured on film or digitally, with special panoramic cameras or more conventional flatbacks, scanners and even mobile devices. Photographers have a choice between normal, wide-angle, fisheye lenses, motorized panoramic heads and tripods and even all kinds of new gadgets and automated technology that can help them out during the process, see Figure 29. [26]

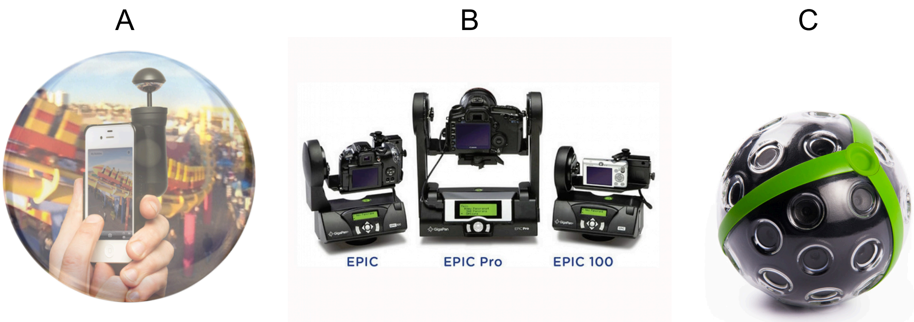


Figure 29. A: BubbleScope gadget [46], B: Epic panoramic heads [47] and C: Panono panoramic ball [48]

The diversity of panoramic equipment can easily overwhelm the photographer, but generally the technology chosen is dependent on the purpose and the quality of the documentation. With the right equipment, basic panoramic photography is fairly simple, even up to the point where the user uses a mobile phone, rotates it around and the application stitches the photo with an app into a panoramic image to view on the fly. Or even throws a panoramic ball into the air, snaps the shot and catches the ball! These panoramas tend to be of low quality. But as the expectations of the quality and the resolution of the final panorama rise, the final quality of the panorama is highly dependent on the skills of the photographer and the chosen equipment.

The photographer has basically a two-way choice: either to use systems that require stitching of the images or systems that do not. To create a panoramic image, covering the full 360-degree field of view, multiple images are always needed. Even using the widest wide-angle lens or ultra high resolution scanning systems, the tripod/mount that the camera is placed on will be visible in the shot. (Unless using a throwable ball) Therefore, multiple images must be captured to cover the full desired angle of view, as shown in Figure 30.

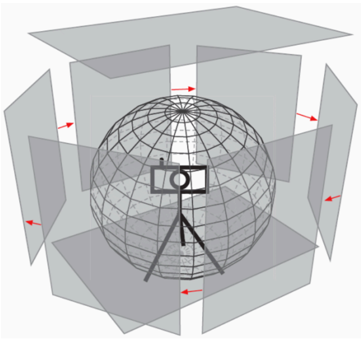


Figure 30. Multiple source images are joined together to produce an all-around view [25]

The final goal of every panorama is to provide a seamless, properly blended panorama by taking uniform images, avoiding parallax, distortion and differences in exposure and focus. [23] The steps in achieving this can somewhat vary, depending on the desired projection and outcome of the panorama, but generally the basic steps are same and apply to all panoramic photography. And the more shots needed, the more complicated it actually gets. The workflow for shooting the Louhisaari 360˚ spherical (equirectangular) panorama is explained in the following chapters.

Since specialized panoramic cameras are superb, but very expensive, the photographer often balances between the quality and the budget. Cost effectiveness and high quality were also the key issues in creating the panoramas in Louhisaari. Therefore the choice was to shoot the panoramas with a DSLR camera, suitable lens and a panoramic head on a tripod. This is also by far the most used setup for modern 360 panoramas. [25]

5.1 Louhisaari Banquet Hall 360˚ Gigapixel Panorama

This chapter briefly describes how the shooting of the 360x180 degree panorama (Figure 31) used in the Louhisaari app was carried out.

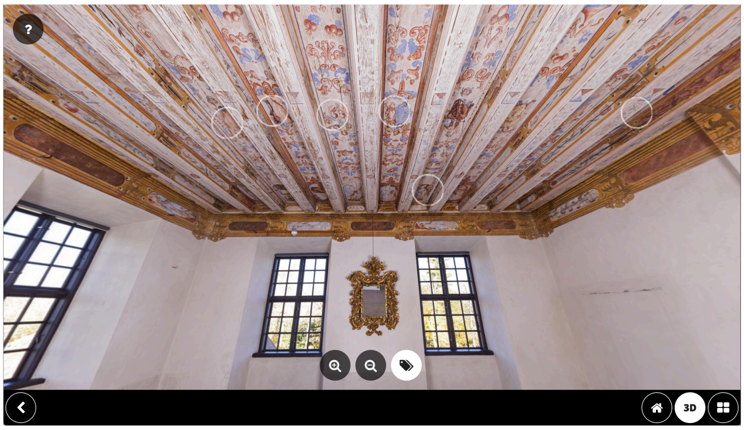


Figure 31. A screenshot from the Louhisaari App showing the 360˚ panorama

Multiple shots are always needed when shooting 360˚ panoramas with a DSLR. This means systematically rotating the camera in increments to the extent of the desired field of view. The size of each rotation increment and the number of images in total depends on the angle of view for each photo, which is determined by the focal length of the camera lens being used, and the amount of overlap between photos. [49]

Naturally, shooting with a very short or a fisheye lens, the number of photos needed is low. For example shooting with a 8mm circular fisheye lens (horizontal, vertical and diagonal FOV of 180 degrees), only 2-3 shots would cover all: one for the top half of the hemisphere and one for the bottom. The problem with such a low number of images would be to deal with the quality, severe distortion and to stitch the images successfully. To blend and stitch the adjacent images together, some overlap is always needed, as shown in Figure 32.

Figure 32. Two adjacent images with 45% overlap stitched

The amount of overlap needed for each shot depends on the quality. The best possible outcome would be using narrow vertical strips and high overlap (easily 50%), since lenses are generally sharpest at the middle. This also offers least lens distortion and aberration. It is often stated that about 30% overlap is sufficient. [49] Going higher only means that one has to shoot more for a given angle of view, as seen with the 45% overlap in Figure 32. The rough number of frames needed to cover the full 360˚ panorama, with the desired overlap, can be calculated with a simple formula or multiple online panorama calculators are provided free of use.

Instead of manually calculating the needed frames and then systematically rotating the camera, a panoramic tripod head can do the same automatically. A panoramic tripod head is a piece of photographic equipment, mounted to a tripod and is designed to help the photographer to set the camera correctly. With a panohead, the photographer can place the camera, so it rotates around the entrance pupil of the lens, eliminating parallax. [50]

There are lots of commercial panoramic tripod heads on the market. The tripod head used in Louhisaari was the GigaPan EPIC pro which is a motorized and computerized panorama head, which is seen in Figure 33A and B. [51] The user can input details, such as the angle of view of the lens and the preferred overlap of the frames. The panoramic head will then make the calculations and take the number of needed shots.

Figure 33. Panorama shooting in action in Louhisaari with the GigaPan Pro

The Louhisaari 360˙ ultra high-resolution panorama was shot with a medium sized full-frame DSLR. The details of the individual images are seen in Figure 34. To cover the full horizontal 360˙ angle of view, of the banquet hall, 240 images were taken. The images were shot from top to bottom, in columns. The vertical angle of view did not cover the full 180˙, since a panoramic tripod head was used and the shots towards the nadir show the panohead and the tripod. Looking carefully at Figure 34, the nadir area and part of the tripod is visible in the bottom part of the equirectangular image. To overcome this problem the photographer has a few choices. The typical one is to take separate shots towards the nadir by removing the tripod and then image editing or stitching them into the final image. If the nadir shots are missing, a nadir logo can be added or conventional image editing used to clone and blend the visible area. With the Louhisaari panorama, the equirectangular image was first converted into cube face and then the bottom face image edited to hide the tripod.

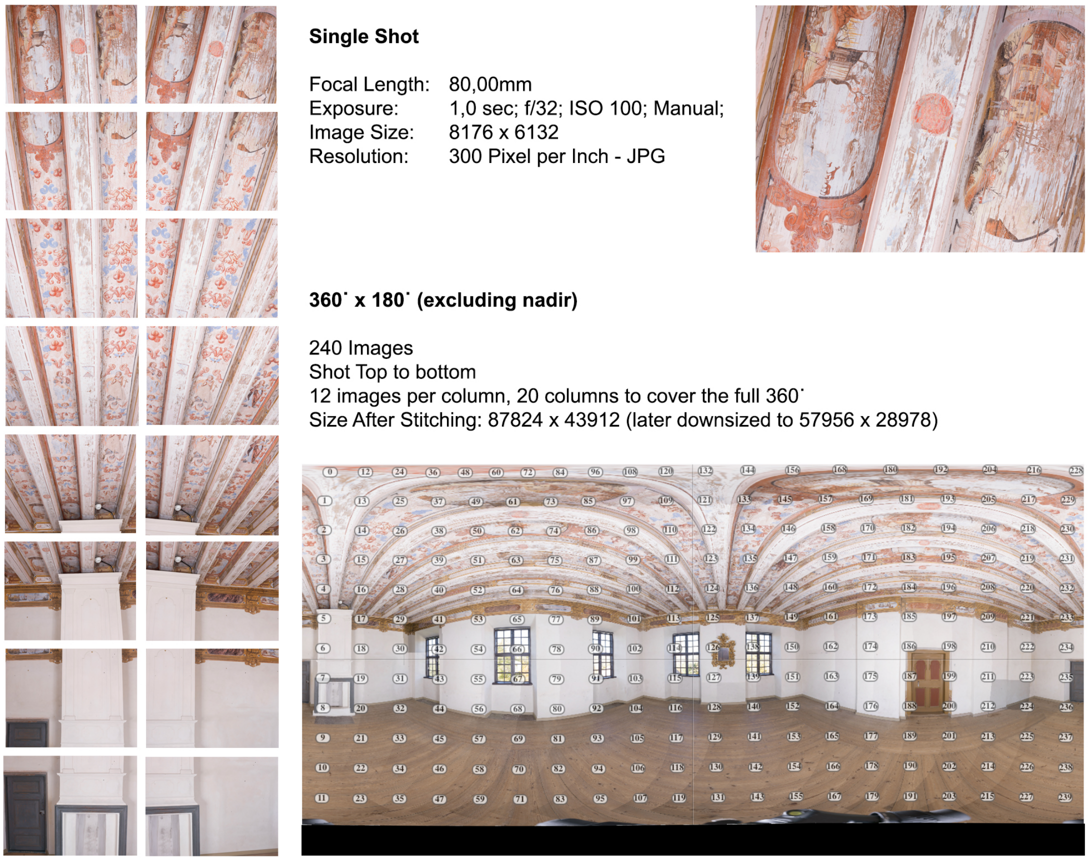


Figure 34. An image sequence, single shot and the final equirectangular image

Taking true high-resolution shots, even with high-end equipment, requires a skilled photographer. A panoramic tripod head can help some in automating the manual work, but it is left to the photographer to choose the right equipment, settings, lighting and to prepare for the shot. The photographer must check multiple of things, such as: for the consistency of shots – shooting in manual mode, no automatic adjustments or focus, constant exposure and eliminating parallax (objects seem to be shifted when viewed from different lines of sight). Also keeping the camera aligned and minimizing camera roll between the adjacent shots will make the stitching of the images easier in the next phases of the development. The overall skill of the photographer is the ultimate determining factor of the quality of the final panorama.

5.2 Flat panoramas – Allegorical Figures

There are several different figures and landscapes included in the ceiling paintings, but for the application ten different ones were selected. Nine figures and one particularly interesting detail where the Louhisaari Mansion is depicted. These selected figures are presented in Figure 35, showing a detail of the image gallery used in the application.

Figure 35. A screenshot of the gallery in the application, 9 figures and one landscape detail

The images for these panoramas were gathered from the extensive documentation material already shot. For each figure, one image shot with visible light was paired with an image shot with a different documentation method. Since each of the images were shot with different camera setups or were subject to image manipulation, some manual adjustments had to be made. Before generating the panoramas from the chosen images, each image pair was manually aligned to create two pixel per pixel matching images. Without this step, mismatches and differences of the images would be evident when later comparing/blending the images with the application.

Rectiliear flat panoramas
Created from:

2500 px x 6000 px
300dpi
24-bit, rgb

Figure 36. The allegorical figure Peace; A: visible light and B: UV fluorescence versions

Each of the flat panoramas were generated from 2500 pixels x 6000 pixels sized, 300dpi, pixel per pixel matched images. Figure 36 shows two different images taken from the allegorical figure Peace. Figure 36A shows the figure Peace shot with visible light and Figure 31B shows the same figure shot with ultraviolet induced fluorescence photography.

The following methods of documentation and analyses were performed on the paintings and the following panoramas were generated during the project, as seen in Table 3.

Table 3. Documentation Methods and Selected Figures

Documentation Method	**Selected Figure shown in the App**
High-resolution macrophotography in Visible Light	360˙ Panorama and all figures
Infrared photography	Bacchus
False color - Infrared photography	Piety
Sidelight photography	Love of One’s Country
Polarized light photography	Charity
Ultraviolet photography	Winged Head
Ultraviolet fluorescence photography	Peace

Image Editing	
Reduction	Liberality
Maximizing color and tone information	Good Fortune
Reinforcing colors	Christian Faith

Creating both the 360˚ multiresolution panorama and the flat multiresolution panoramas from these images are explained in greater detail in Chapters 7.4.1 and 7.4.2.

6 Panorama Creation - Stitching

Image stitching or photo stitching is the process of combining multiple photographic images with overlapping fields of view to produce a segmented panorama or high-resolution image. Commonly performed through the use of computer software, most approaches to image stitching require nearly exact overlaps between images and identical exposures to produce seamless results. [52] Stitching the images is the next stage of panorama creation, after shooting and image editing the individual shots, see Figure 37.

Figure 37. Panorama workflow - Stitching

The quality of stitching varies from program to program. Some programs offer automated ways and some require manual work and utilize different algorithms for the stitching and blending. Nevertheless, all of the stitching programs work with the same basic principles explained in this chapter.

6.1 History and Evolution

As the digital era brought advances into digital photography, image alignment techniques started being applied and developed in the mid 1990s. QuickTime VR was one of the first available commercial stitching software available and is still used for displaying spherical panoramic pictures in a browser. [53] One of the most known contributors during this time period was a German physics and mathematics professor Helmut Dersch. He is the creator of the pano12 library and The Panorama Tools package that are currently used by many front-end and command-line programs. The pano12 library, the core of The Panorama Tools is under the public domain, GNU General Public Licence (GPL). [54]

The PanoTools offer a collection of free tools and libraries for creating panoramas and 3D objects, such as the PTStitcher, but using these without a GUI is very difficult. Therefore a number of GUI front-ends, free and commercial, based on these tools have been written. Now days there are numerous of different photo stitching software created, some richer in features than others. Many of these programs still depend on the same set of libraries and code Dersch created.

The stitching software chosen for the Louhisaari project was the PTGui (Graphical User Interface for Panorama Tools), one of the leading photo stitching applications. As the name implies, it started as a GUI for the Panorama Tools but over the years nearly all of the functionality has been implemented into the program. [55] Since PTGui works under a proprietary license and runs only on Windows and Mac OS X, a good alternative would have been Hugin (http://hugin.sourceforge.net). Hugin is an open source alternative also based on the same Panorama Tools but offers support for various other systems such as Fedora, Ubuntu and FreeBSD. [56]

6.2 Image Stitching

Panoramas are generally composed of an image sequence. Since each input image in the sequence is a perspective image that could be projected onto a sphere, the images must be projected and resampled into a different coordinate system to create the desired panoramic projection. PTGui, as seen in Figure 38A and B, offers a wide choice of panoramic projections.

Instead of covering each step of using a specific program, this chapter explains what stitching software actually do and what steps the stitcher algorithms use - by showing examples of the process used in stitching the Louhisaari panorama, seen in Figure 38B.

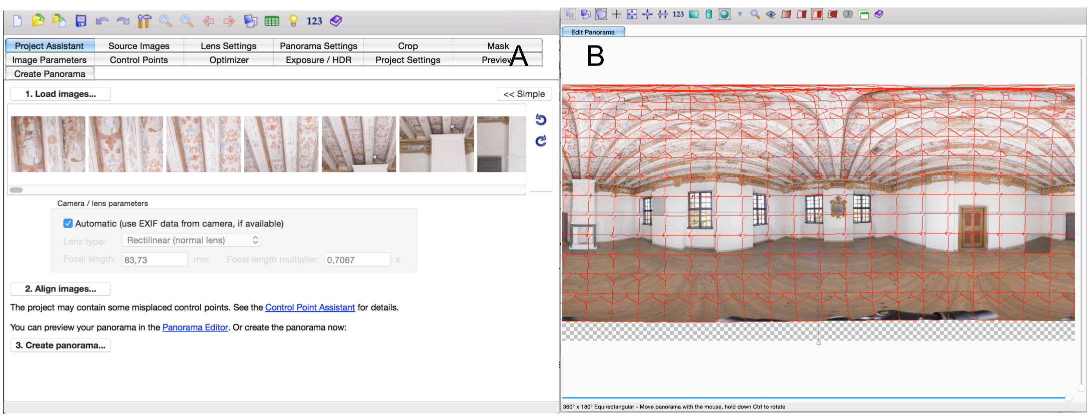


Figure 38. Screenshots from the PTGui software. A: Basic Interface B: Panorama Editor

Considerations

There are a few considerations that can be taken into account for successful stitching of the images.

1. The images in the sequence have a reasonable amount of overlap (at least 20-30%). This is needed both to overcome lens distortion and making sure that the images have enough information for feature points to align and blend properly.
2. The exposure and contrast between the frames is consistent. Image and vignette correction is applied during the stitching but might still result in visible seams.
3. Avoiding misalignment (pan, roll, tilt). This is easier avoided when using a panohead.
4. Zoom. Stitching software assume all images have the same zoom level.

Alignment

The user first selects and loads the grid of input images into the program, as seen in Figure 38A. At this stage the program tries to infer how many rows and how many columns there are in the sequence and to align the images into a grid. Many of the programs have tools for image sequences taken with a panoramic head, where the user can input the number of rows, overlap used, the order of images and other data for more efficient alignment.

PTGui as other programs also use the EXIF data of the images in order to find out the lens type and the focal length used to calculate the lens correction needed and the image shift between adjacent images.

In order to align the images most of the algorithms used with photo stitching software try to search for feature points, often also called as control points. [57] Feature points are features common for each adjacent picture, such as cracks in the wall, or brush strokes in the ceiling paintings. Some images might not have any feature points or only a few and for these images, matching feature points need to be set manually, see Figure 33 A. For example, the algorithm failed to find feature points in some of the images in the Louhisaari panorama, since these were taken of a white wall, with no detectable features.

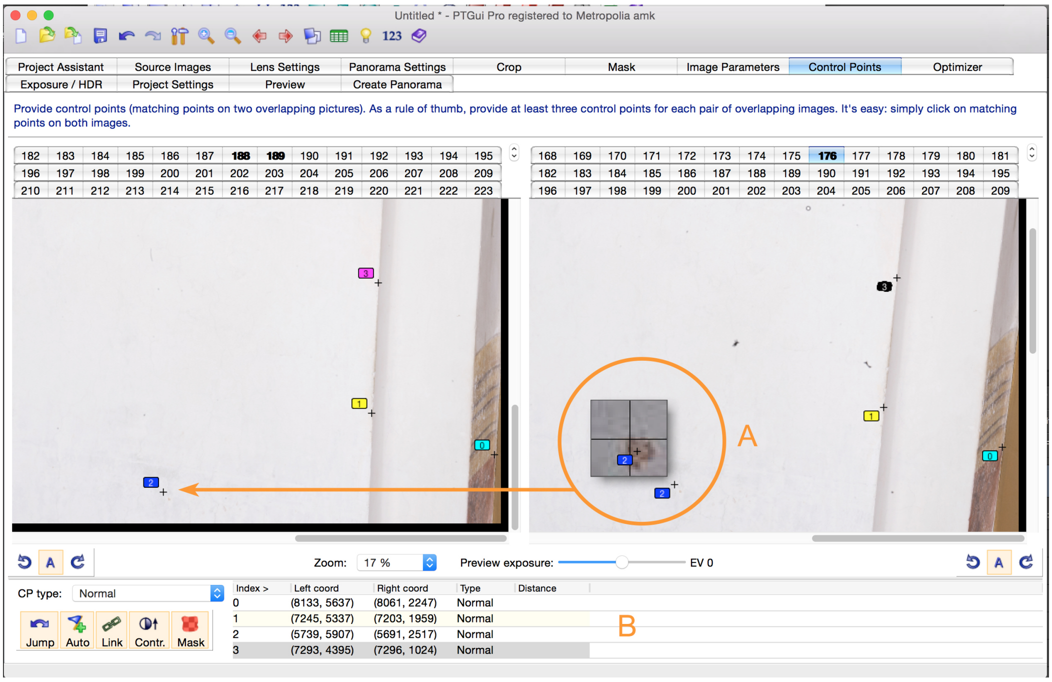


Figure 39. Setting manually feature/control points for two adjacent images in PTGui. A: Visual manual addition and B: the coordinates for the control points

Image Stitching Algorithms Overview [58]

1. Detects feature points for each image

Usually done with algorithms such as the SIFT (Scale-invariant feature transform) which is an algorithm in computer vision to detect and describe local features in images [59]

2. Matches feature points for adjacent images

Finds the best pairwise optimization for all of neighboring images in the grid, see Figure 39A.

3. Finds a set of features that will produce a high-accuracy alignment.

The minimum set is typically four, see Figure 33B. PTGui will inform the user if not enough feature points are found. The RANSAC (RANdom SAmple Consensus) technique is often used to estimate the homography matrix of the feature points. The images are later warped to the desired projection based on the matrix. [60]

4. Optional Image Correction
Vignetting is the reduction of an image's brightness or saturation at the periphery compared to the image center. [61] Vignetting is often caused by settings in the camera or limitations in the lens. Stitching software often provide vignette analysis and correction, since the darkening of the edges in the frame may cause visible seams in the final image. This can again be avoided with proper overlap of the sequence images or image editing the shots in advance.

5. Projecting onto a surface – warping
When warping the images into the projection of the desired panorama, some ‘in between’ calculations are needed. PTGui provides several interpolation algorithms for this. Each algorithm has its pros and cons; which is the best interpolator is a matter of taste and is depending on the actual images. The differences between interpolators can be subtle but they are noticeable around diagonal lines (smoother or slightly jagged) and around high contrast areas (more or less 'halo' artefacts).

6. Blending
The final step is to blend the images together. Blending is the process of merging the warped source images into a single panoramic image. [62] This is done with different blending algorithms. PTGui offers the built in PTGui Blender and the use of two other open source third party applications, the Enblend [63] and the Smartblend [64]

Result

The result of the stitching process varies greatly depending on the desired projection and the input image sequence data. Images with non-uniform lighting, mismatch in focus, poor overlap and lack of detectable detail and feature points (such as a uniform blue sky) might lead in incorrectly stitched images. Stitching success also varies from program to program, as different programs and software use different algorithms for the stitching.

The GigaPanEPIC pro comes with the GigaPan Stitch software and states to “seamlessly work” with the panohead and assemble, align and blend the images into the panorama. [65] But poor stitching results were actually one of the reasons why PTGui was chosen over the Gigapan Stitch. The bundled Gigapan stitch did not give any options for feature detection. The white wall in Louhisaari proved challenging since many of the images were lacking detectable features; the images did not have any control points and could not be blended. Therefore some parts of the final panorama were just “cut off” and no way of fixing it was offered by the software. The same problem was evident when the images were imported into the PTGui, as seen in Figure 40A, B and C where some pixels are missing from the middle of the panorama, but was fixed by manually adding the control points.

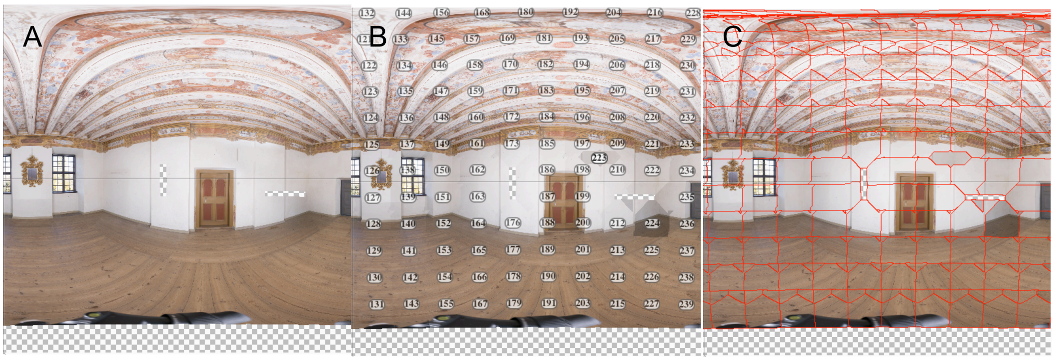


Figure 40. Orphaned images lacking control points resulting in blending errors. Panorama Editor with A: Stitched Image B: Image numbering C: Seams

Stitching can be a very time consuming process. If the algorithms fail to find feature points, it often results in manual feature point detection and tweaking the settings to align the adjacent images precisely. This can take time, if the panorama is composed of many images and have many orphaned adjacent image pairs. It is also a good practice to pre-process the images before the stitching process to correct possible distortion, vignetting or removing spots or other artefacts created by the dust in the lens.

CPU

The stitching is done on the CPU. Stitching and blending requires a lot of disk and memory access. In particular for large panoramas, or on multi core computers, not the processor but the hard disk may be the speed limiting factor. If the aim is to increase the stitching speed the programs supports accelerated stitching and tweaking the assigned RAM and other hardware use to speed up the process. [66] [67] Still, with adequate hardware, after the pre-processing and feature point detections, the actual stitching/blending of gigapixel sized panoramic images can take up to several hours, even days.

Output

The panorama can be rendered into several file formats and projections. PTGui is capable of creating a layered panorama, where the individual source images are converted into separate layers in the output file for further editing or creating a blended single seamless image. Supported output file formats vary among programs but PTGui offers the choice between: [62]

- JPEG (.jpg)
- TIFF (.tiff)
- Photoshop (.psd)
- Photoshop Large Document Format (.psb)
- QuickTime VR (.mov).

Since the psb file format supports sizes up to 300,000 pixels in any dimension [68] and is also one of the formats supported by the panorama viewer krpano, it was the obvious choice for the output format of the Louhisaari panorama.

7 Panorama Creation – Rendering

Now that the panoramic image was successfully created and stitched to the final ultra high-resolution equirectangular image, it was time to put the image mosaic into good use. This chapter explains how the panoramas in the Louhisaari App were created and rendered for viewing purposes (Figure 41).

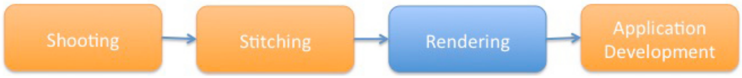


Figure 41. Panorama Creation - Rendering

In this last stage of panorama creation, both the 360˚ panorama and the flat panoramas were rendered, tested and optimized for viewing before integrating them into the application.

7.1 History and Evolution – Web3D

Web3D refers to all interactive 3D content which are embedded into web pages html and that we can see through the browser. [69] Initially, in the midst of the 90s, it started out as an idea to serve 3D content as an alternative to HTML. Websites, instead of text and images, would fully display 3D content. The user would navigate in the 3D space, a virtual cyberspace, for example visiting virtual city offices instead of real ones and meeting, chatting in virtual meeting rooms and spaces.

Although the idea was not that bad, it never really got off the ground. The technology had advanced into a state where all of it was possible, but it was still way ahead of its time. Low bandwidth dial-up connections and computer processing power were an issue. These technologies were invented during a time when 14 Kbps and 28 Kbps Modems were considered superfast and computers had a fraction of the processing power we have now. VRML (Virtual Reality Modelling Language) was created and ISO standardized in 1997, preceded soon by the more mature and refined X3D initiative in 2000. Both of these file formats were designed particularly with the World Wide Web in mind to provide interactive 3D vector graphics. [70][71]

Soon after VRML, java developers created the Java3D, a high-level API to render 3D in Java. It ran atop OpenGL or Direct3D, within an html page as an applet. [72] It had the same fundamental idea as VRML, to structure the scene using a scene graph representation of the objects, but in contra was not a file format as VRML is. Java3D did not gain huge support, mostly for the reasons that it did not function quite acceptably within a browser and creating anything with it required at least some level of coding experience. [73]

From these early pioneers web3D has developed and advanced. In many ways, the browser has become the most important application used across devices. A variety of technologies, tools and formats have been developed for creating and viewing 3D content in the browser. Most of these have required the users to install plugins to view the content in the browser, or used as native applications. As technology has matured, it is now possible to render interactive 3D content in a browser, without the use of a plug-in. At the time of writing, WebGL and CSS 3D, Figure 42, have proved to take the lead as the two major technologies for creating 3D in the browser. [74]

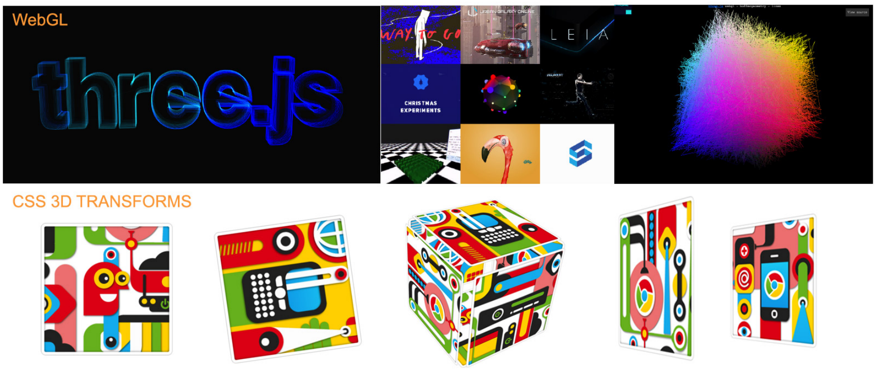


Figure 42. WebGL with three.js [75] and CSS3 3D Transforms Examples [76]

WebGL: a cross-platform, royalty-free web standard for a low-level 3D graphics API based on OpenGL ES 2.0, exposed through the HTML5 Canvas element as Document Object Model interfaces. [77] The WebGL 1.0 specification was released in 2011 and the development of the 2.0 specification started in 2013 and will soon be released.

With WebGL it is possible to build applications with common web technologies: HTML/CSS/JavaScript, plugin free. The control code of a WebGL program is written in JavaScript and shader code. The 3D is implemented straight in the browser and takes advantage of the GPU, bringing hardware-accelerated graphics. WebGL 1.0 is supported in the stable releases of most major browsers. Chrome, Firefox, Internet Explorer, Opera, and Safari are all known to have good WebGL support on both desktop and mobile browsers, see Figure 43. [78] Many JavaScript 3D libraries and frameworks have been created to simplify the workflow, thjee.js being the most popular and widely used one. [75]

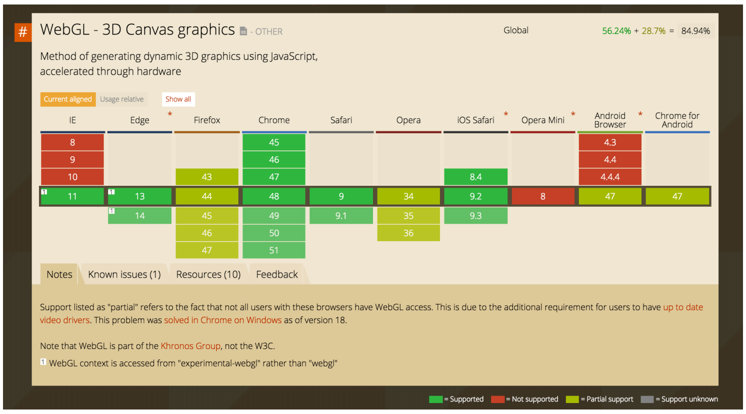


Figure 43. WebGL support in modern browsers [79]

CSS 3D: DOM based 3D. CSS transforms allows elements styled with CSS to be transformed in two-dimensional or three-dimensional space. The fundamental difference to WebGL is that these elements are not real three-dimensional objects and have no real depth. [80] Since the user is manipulating DOM objects, the HTML CSS3 transforms only work for simple geometric shapes like lines or rectangles. CSS 3D is also widely supported on most major browsers, see Figure 44 . However, when dealing with complex shapes and scenes, WebGL is the way to go.

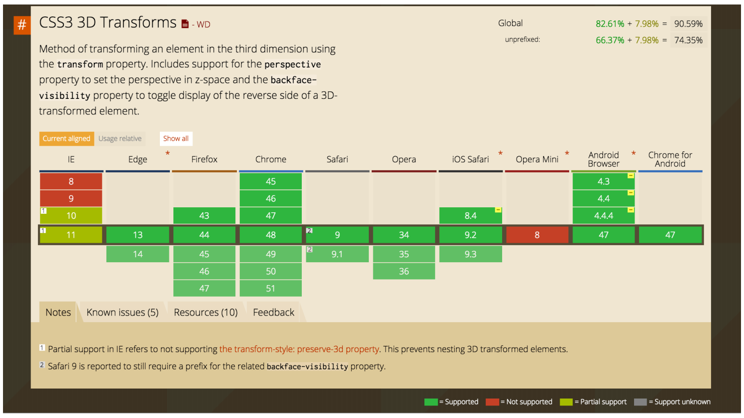


Figure 44. CSS 3D Transforms support [81]

CSS transforms, just as WebGL are managed by the browser and can be accelerated by the GPU. This means that some of the processing work is loaded on the GPU instead of the CPU yielding performance gain of graphic intensive applications. This has an important effect on the overall performance of the browser, resulting in smoother transitions and animations thus improving the user experience.

7.2 Image Based Rendering

In traditional 3D computer graphics, the 3D geometry of the scene is known. VRML, X3D and other technologies have strived to render these virtual environments in real-time and now with the technology maturing, real-time 3D graphics can be implemented efficiently in the browser with WebGL.

Image-based rendering (IBR) refers to a collection of techniques and representations that allows 3D scenes and objects to be visualized in a realistic way. [82] The goal with IBR is to replace traditional modeling and rendering methods by replacing geometry and surface properties with images. This has proved to be a very potential approach with panoramic imaging, since the image needs to be projected onto a simple sphere or a cylinder to achieve the effect or even rendered with no geometry at all.

The previous image-based rendering techniques can be classified into three categories according to how much geometric information is used, see Figure 45. [83]

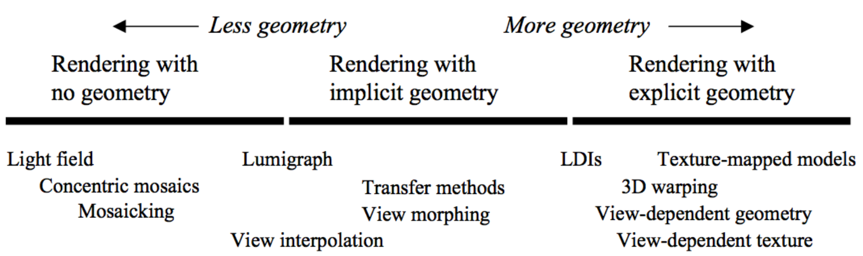


Figure 45. IBR classification [83]

Rendering with implicit geometry is sometimes called as hybrid IBR. Hybrid IBR techniques use some amount of geometric information, in addition to images, to synthesize new views of scenes. The amount of geometric information used varies from per pixel depth, employed in techniques based on 3D image warping, to reasonably detailed polygonal meshes, used in Light Field Mapping. With a large spectrum of possibilities, hybrid techniques are the most numerous among the IBR approaches. [84]

If a scene is rendered with no geometry the technique often relies on the characterization of the plenoptic function. The plenoptic function describes all of the image information visible from a particular position. The original 7D plenoptic function is defined as the intensity of light rays passing through the camera center at every 3D location (V_x, V_y, V_z) at every possible angle (θ, φ), for every wavelength (λ), at every time (t), see Formula 4. [85]

$$P_7 = P(V_x, V_y, V_z, \theta, \varphi, \lambda, t) \quad (4)$$

Since the viewer of a common panorama is stationary and located in the center of the sphere, the viewport is fixed. Therefore the simplest plenoptic function is a 2D panorama (cylindrical or spherical) when the viewpoint is fixed, see Formula 5. [85]

$$P_2 = P(\theta, \varphi) \text{ for pan and tilt} \quad (5)$$

IBR can be viewed as a set of techniques to reconstruct a continuous representation of the plenoptic function from observed discreet samples, such as viewing the panorama from a fixed point of view, see Figure 46. A regular rectilinear image with a limited field of view can be regarded as an incomplete plenoptic sample at a fixed viewpoint. [82]

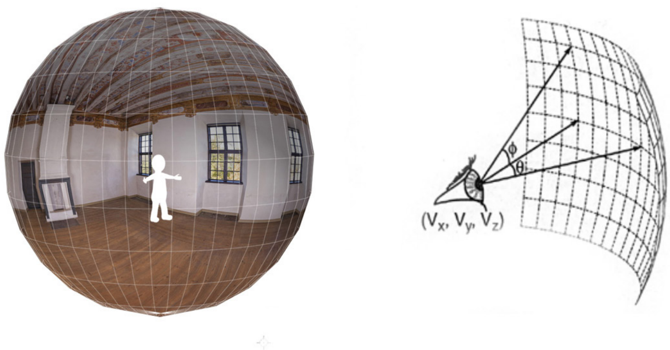


Figure 46. Plenoptic Function Based Data Representation [86]

After shooting and stitching the image sequence, the panoramic mosaic must be projected back on a cylindrical or spherical surface for viewing. This can be done using special software that can display the wrapped image or utilizing web technologies such as CSS 3D or WebGL. The scene can be generated with explicit geometry, such as a sphere and mapping the image on it and then projecting the scene to the user's 2D view. Also, pure IBR can be used, as in the case of the old proprietary format the Apple QuickTime VR, which uses pure image based rendering to view the panoramas. [84] The user views the panoramic image within the cylinder/sphere and is often able to pan, roll and tilt the view as if looking through a virtual camera. When mapping an equirectangular image onto a sphere, the normal/rectilinear projection is commonly used. With special viewing software, the panoramic image can be mapped into several other projections.

Panorama Viewers

There are plenty of free and open source panorama viewers available. They can roughly be categorized into the following categories [87]

Used within a browser

1. Java based Viewers such as Java Applets – support for Java needed from the platform. Not that used anymore.
2. Plugin based viewers that require the user to install a plugin to run in the browser, some are platform specific. (Flash, Quicktime, Shockwave)
3. Native 3D support, typically built using HTML5, CSS3, JavaScript. (WebGL and CSS 3D based)

Stand-alone viewers

1. Executable for a specific platform
2. Java based viewers that run on any Java supported platforms

In addition to these, panoramas can be generated and viewed without a proprietary viewer. For example, utilizing WebGL and the Canvans element a panorama can be created by mapping the equirectangular image onto a hollow sphere, and placing the camera inside the sphere for viewing. Naturally, this would be a very simple way of using the panorama and would lack the special features and options for viewing and optimizing the panoramas that many of the viewers provide.

7.3 Krpano Viewer

The krpano Panorama Viewer (www.krpano.com) is a small and very flexible high-performance viewer for all kind of panoramic images and interactive virtual tours. [15]

Krpano offers both a Flash and a HTML5 viewer, but for the Louhisaari application the HTML5 version was the obvious choice. Flash is slowly getting obsolete and replaced with HTML5, as support for the Flash player drops and the possibilities of HTML5 grow. The krpano HTML5 viewer uses HTML5 CSS 3D Transforms or WebGL for displaying the panoramic images directly in the Browser. [15] It has multiresolution support and can handle Giga-, Tera- and even Petapixel sized panoramas.

To use the viewer, the user embeds the viewer files (krpano.js for HTML5 and krpano.swf for FLASH) into a HTML page. As krpano is a commercial product, these files are encrypted, but can be controlled from en external interface. To access the krpano from JavaScript a krpano JavaScript-Interface object is required. The interface object provides functions to get and set variables to given values and call and execute krpano action code. A command or a function is called as a ‘krpano action’. With this action code, the krpano scripting language, the krpano viewer can be customized in multiple ways. Almost everything is customizable with the viewer. In addition to the scripting language, Krpano offers a small interface for developing your own plugins with either JavaScript or Actionscript3.

The krpano offers the following core files, as seen in Figure 47:

Figure 47. krpano core files

krpano.js or **krpano.swf**- The HTML5 Viewer or the Flash Viewer core file

embedpano.js - Used for embedding the viewer into a HTML page. Device/system feature detection is done in this file; also many cross-browser fixes for mouse wheel and touch support. If the panorama is created with the help of the droplets provided by the software (Windows batch / AppleScript files); embedpano.js and krpano.js are merged into a tour.js file.

krpano.html - A template page to embed the viewer.

krpano.xml - Krpano uses simple XML files to store the settings for the viewer. These files use the basic XML syntax with krpano specific elements, attributes and values. The XML files are used as a transport-format. When the viewer is launched, the XML is parsed and the elements transformed/mapped into krpano internal data structures.

The application logic is explained in Figure 48 as follows:

A. The user opens the krpano.html page with the embedded panorama
B. Tile indexes are used to address right multiresolution image tiles
C. From the source code it can bee seen that the embedpano.js, krpano.js and krpano.xml files are embedded
D. As the krpano.js file is launched it parses the krpano.xml page, and the XML elements get mapped into internal data structures
E. External JavaScript code can call and execute krpano Actions

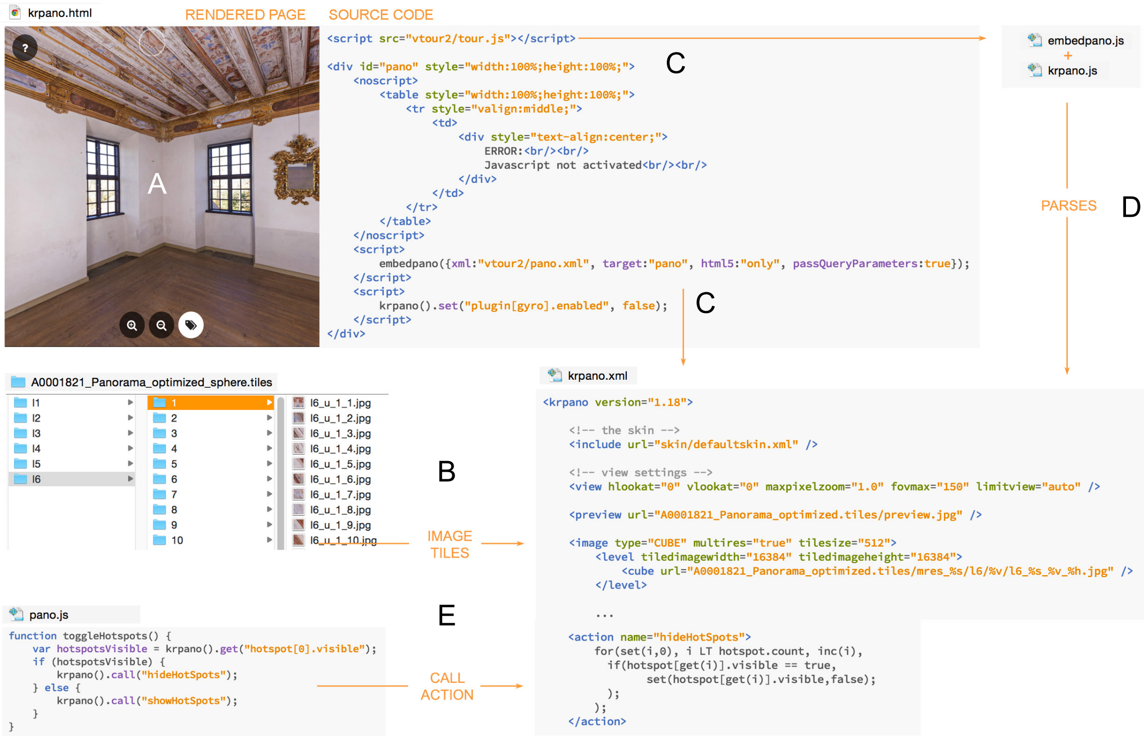


Figure 48. Krpano viewer logic

Krpano also includes a set of tools, called the krpano Tools - to help with preparing the panoramic images for viewing. The tools can be called from the command line or from Windows batch / AppleScript files. The default droplets are paired with a config-file and these can be customized. With the droplets, Figure 49, users can easily create virtual tours, panoramas, tile, resize, convert and transform images etc.

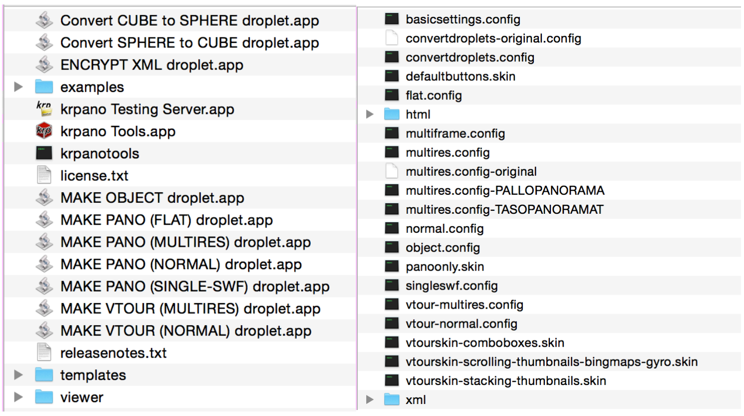


Figure 49. krpano droplets and config files

Supported input and output image file-formats for the krpano Tools, all formats only in 8 or 16bit RGB or RGBA format [88]:

- BigTiff (.btf, .tf8, .bigtiff)
- JPEG (.jpg, .jpeg)
- Photoshop Document (.psd)
- Photoshop Big Document (.psb)
- Kolor RAW (.kro)

It is good to note that krpano offers support for both the BigTiff and the Photoshop Big Document formats. This enables the user to work with even larger panoramic images in size than the conventional file formats allow.

7.4 Multiresolution Panorama Creation

The krpano (Version 1.18) Flash viewer supports a variety of panoramic and traditional image-file formats, such as spherical, cylindrical and partial panoramas, panoramic videos, object movies and much more. The krpano HTML5 Viewer, is much more limited. When used with multiresolution technology, it supports only cubical and flat panoramic images. Spherical images, like the equirectangular image used in Louhisaari are actually converted into a cubical panorama when multiresolution panoramas are created.

This is not actually a limitation of the krpano viewer, but a limitation of the technologies it utilizes. HTML5, CSS 3D Transforms and WebGL are all very new technologies and different systems and browsers offer various support. The HTML5 viewer tries to deal with different situations by switching between WebGL and CSS3D modes depending on the browser and the device used. [89]

While WebGL is capable of dealing with complex geometry, CSS 3D transforms can only deal with simple geometries, such as rectangles. If WebGL support fails or is not supported, the viewer tries to render the scene with CSS 3D transforms. [90] Therefore to secure that the panoramas can be viewed with a variety of devices and platforms, only panoramic formats with flat surfaces like the six flat sides of a cubical pano can be used.

7.4.1 Louhisaari 360˚ Multiresolution Panorama

Multiresolution technology allows the user to view large files without loading everything at once. Instead using and displaying a single huge panorama, the image is split into tiles and different resolution levels. As mentioned above, multiresolution with the krpano HTML5 viewer can only be used with flat surfaces. Cubical images also give a better display quality and a better rendering performance. [91] Therefore, the equirectangular image needs to be converted into a cube strip or separate images for each of the sides of the cube, see Figure 50. Both the PTGui and the krpano Tools offer tools to make the conversion.

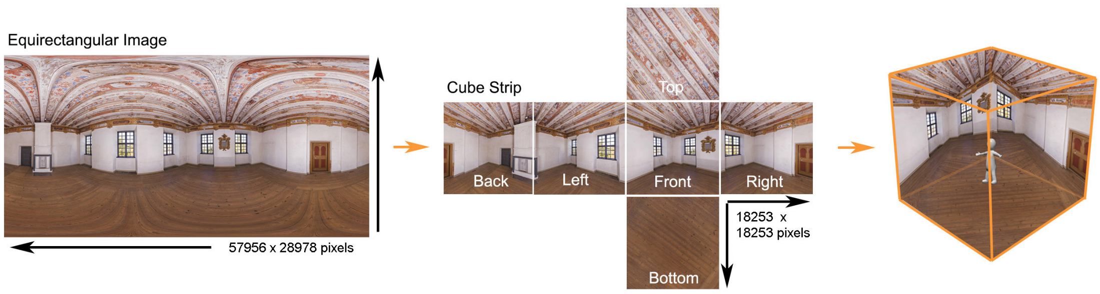


Figure 50. Equirectangular to 6 cube faces conversion

Krpano offers a few automated ways to achieve this. First, a droplet can be used only for the conversion, called the convert SPHERE to CUBE droplet. To determine the size of each cube the following calculation can be used, Formula X:

cubewidth = spherewidth / PI (6)

Louhisaari cubewidth = Equirectangular image width / PI

Louhisaari cube face:

Louhisaari cubewidth = 57344 pixles / PI

Louhisaari cubewidth = 18253 pixels

The user can also use the MAKE PANO (multires) or MAKE VTOUR (multires) droplets that both convert the image and create the basic files. [92] This is the easy way out, but offers limited control over the final quality, as the droplets are making estimated calculations of the level steps and the tile sizes. When working with projects, where optimum quality across devices is an issue, the config files and settings can in most cases should be manually edited and optimized. It is important to study the final panorama to ensure that the settings give a good degree of sharpness and quality throughout the zooming levels.

For the multiresolution tiles in Louhisaari, the equirectangular image was first converted into the cube faces and then the level tiles were created from the cube faces and the XML modified accordingly.

To understand the principles of how the multiresolution approach works in krpano and how to find the optimal size for the images, we have to look at some fragments of the code, created by the droplets used. Code example 1 shows how the cubical panorama is defined in the XML file structure. With the <image> element, the user can set the type of the panorama and the sizes & the paths to the image tiles. Depending on the image type and the related settings, the url of the image should contain several placeholders to allow addressing the right image file; for the cube side: %s ,for the horizontal tile index: %h and for the vertical tile index: %v. [93]

From the code example below, it can be seen that this generated multiresolution panorama has six resolution levels, the first level being 512px x 512px and the deepest, highest resolution level 16384px x 16384px for each face. The size of each tile is 512 pixels. Note, that all of the tiled image heights have been set as multiple of 512.

```xml
<image type="CUBE" multires="true" tilesize="512">
    <level tiledimagewidth="16384" tiledimageheight="16384">
        <cube url="A0001821_Panorama_optimized.tiles/mres_%s/l6/%v/l6_%s_%v_%h.jpg" />
    </level>
    <level tiledimagewidth="8192" tiledimageheight="8192">
        <cube url="A0001821_Panorama_optimized.tiles/mres_%s/l5/%v/l5_%s_%v_%h.jpg" />
    </level>
    <level tiledimagewidth="4096" tiledimageheight="4096">
        <cube url="A0001821_Panorama_optimized.tiles/mres_%s/l4/%v/l4_%s_%v_%h.jpg" />
    </level>
    <level tiledimagewidth="2048" tiledimageheight="2048">
        <cube url="A0001821_Panorama_optimized.tiles/mres_%s/l3/%v/l3_%s_%v_%h.jpg" />
    </level>
    <level tiledimagewidth="1024" tiledimageheight="1024">
        <cube url="A0001821_Panorama_optimized.tiles/mres_%s/l2/%v/l2_%s_%v_%h.jpg" />
    </level>
    <level tiledimagewidth="512" tiledimageheight="512">
        <cube url="A0001821_Panorama_optimized.tiles/mres_%s/l1/%v/l1_%s_%v_%h.jpg" />
    </level>
    <mobile>
        <cube url="A0001821_Panorama_optimized.tiles/mobile_%s.jpg" />
    </mobile>
</image>
```

Code example 1. Cubical panoras defined with XML

For krpano, the size of the tiles can be anything between 256px and 1024px. A today recommendation with a view to future versions with GPU rendering and HTML5 would be using 512 as tilesize. The tilesize affects the loading and decoding time and also the rendering performance. [94] This is for performance reasons, but again a feature of WebGL, not the viewer software. Since krpano uses WebGL, power of two (POT) textures are recommended to be used. OPENGL ES 2.0 and WebGL have only limited non power of two support and if NPOT textures are used they are rounded up, decreasing performance and increasing the memory load. [95] With desktop browsers, this might feel like a small issue and generally not noticeable, but can have a bigger effect with handheld devices. Also since the Louhisaari application is very graphics intensive, anything that can increase the usability and optimize the performance must be thought of. In any case, it is a good practice to keep the texture as POT to ensure the best possible performance. Therefore the size each cube face was first downsized from 18253 to 16384 pixels (which is perfectly dividable by 512) before generating the tiles.

When the user opens the application and launches the panorama, the viewer maps the cubical images into the faces. For the first level in the Louhisaari panorama, a single 512px x 512px texture is loaded for each face, see Figure 51A. As the user zooms into the panorama, depending on the zoom level and the angle of view, the viewer calculates when the transition from the first level onto the next is made and the new tiles are loaded. Again, looking at Code example 1, the second level tiled image height is 1024 pixels. Therefore with the level 2, each cube face is generated from 4 tiles, 512 pixels each.

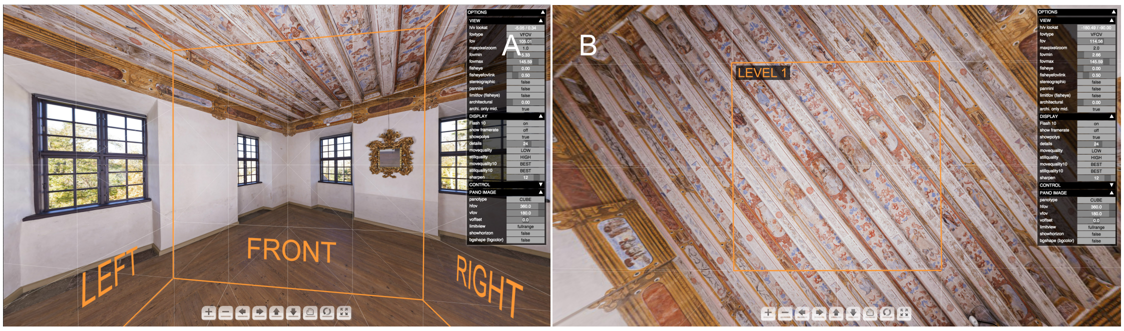


Figure 51. A: Level 1 view of the panorama & B: level 1 top cube face

As the user zooms and progresses in the multiresolution tile pyramid, the required levels and tiles are loaded, as seen in Figure 52. The total cube face size of the last level 6, is 16384px x 16384px.

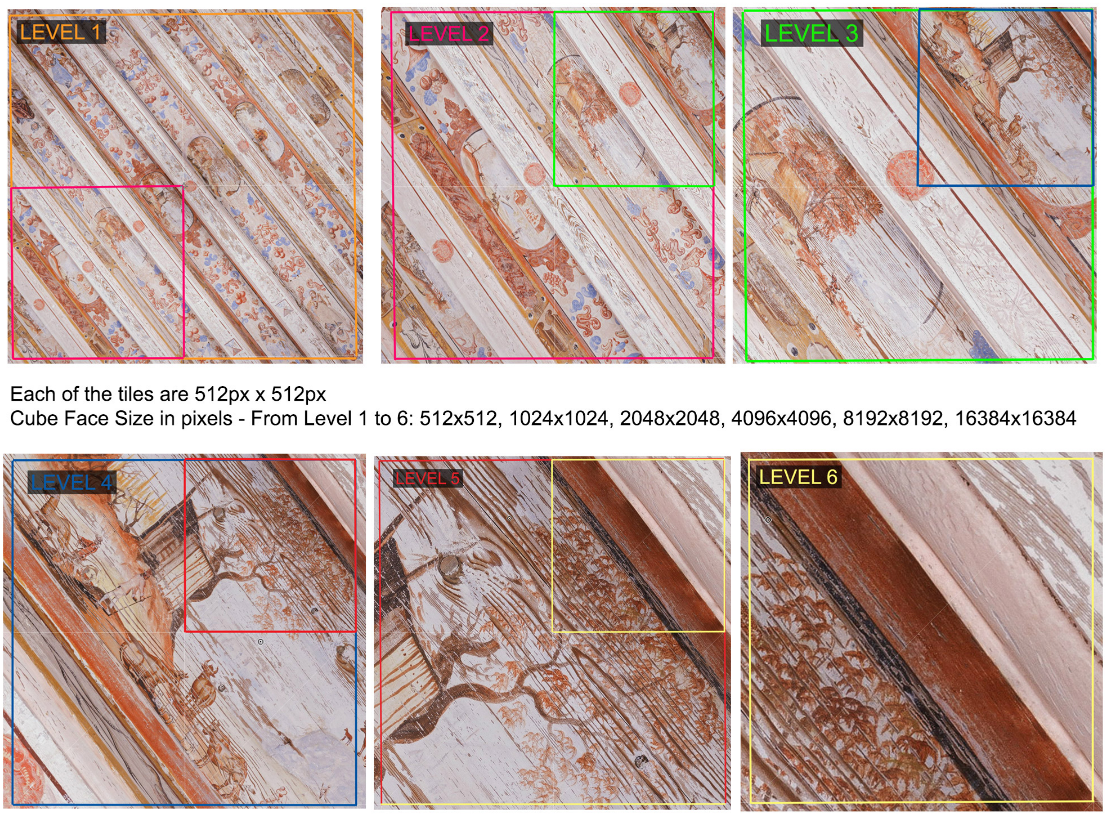


Figure 52. Zooming and progressing in the tile pyramid from level 1 to 6

To fully show one cube face with this level resolution, 32 tiles would be needed both horizontally and vertically, 512 pixels each. Naturally, since the user has zoomed in, only a few tiles from the level are loaded and shown. As the users pans, tilts or zooms the camera, new tiles are loaded into memory. If the memory limit is full and the tiles are no longer needed for the current view, they are released from the memory. The user can individually set the memory limit, even for different devices. The current default memory maximum settings for the html5 viewer are [96]:

- Desktop: 150-400 MB (depending on the full screen size)
- Tablet / Mobile: 50 MB (Android, Windows, Silk, Blackberry)
- iOS (before 7.1): 40 MB
- iOS (7.1 and above): 50 MB
- iPhone 4/4S: 40 MB

This is the key how the multiresolution technology handles the gigapixel size images. Large amounts of data are never loaded simultaneously. To view the Louhisaari, stitched 360˚ equirectangular panorama the memory need can be calculated as shown in Formula 7:

$$\begin{gathered}57956 \text{ pixels} * 28978 \text{ pixels} = 1.68 \text{ Gigapixels} \\ 1.68 \text{ Gigapixels} * 24\text{bits/pixel} = 40{,}3 \text{ Gigabits} \\ 40{,}3 \text{ Gigabits} / 8 = 5.04 \text{ GB}\end{gathered} \tag{7}$$

For the multiresolution approach, it is harder to calculate the memory need, since the tiles loaded differ in each device, depending on the browser dimensions and the angle of view + zoom level. But lets make an estimate and say that the uses is viewing the panorama with a MacBook Pro, with a 2880 by 1800 pixel retina display. The krpano viewer shows a 30˚ FOV (zoomed in) and is able to show 6 tiles horizontally and 4 tiles vertically, naturally some of the tiles are only partially visible.

To calculate the memory needed for the 12 visible tiles, assuming they have full JPG quality, no alpha channel and no compression, is shown in Formula 8:

$$\begin{gathered}12 * (512 \text{ pixels} * 512 \text{ pixels}) = 3.15 \text{ Megapixels} \\ 3.15 \text{ Megapixels} * 3 \text{ bytes per pixel} = 9.44 \text{ MB}\end{gathered} \tag{8}$$

Now this is a really rough calculation and only for the textures showing currently on the display. Still it gives and understanding of how efficient this approach is. To find out size of the loaded textures, the memory & CPU load and details on how processes are using this memory over time, browsers offer a variety of tools to do that. For example the Chrome DevTools are a set of authoring and debugging tools built into Google Chrome. [97]

For the Louhisaari 360˚ panorama, the following files and multiresolutions were created, presented in Table 4, with the following folder structure as seen in Figure 53.

Table 4. Multiresolution level details

Original Resolution:	57344 x 28672 ~ **4,62 Gigapixel psb File**
Multiresolutions/ Each Face:	16384x16384, 8192x8192, 4096x4096, 2048x2048, 1024x1024, 512x512
Tilesize/Tiles:	512x512 = 32x32, 16x16, 8x8, 4x4, 2x2, 1x1 = 1366 Tiles per Cube Face x 6 Faces = 8196 Tiles (**~583 MB Jpegs**)
XML:	pano.xml

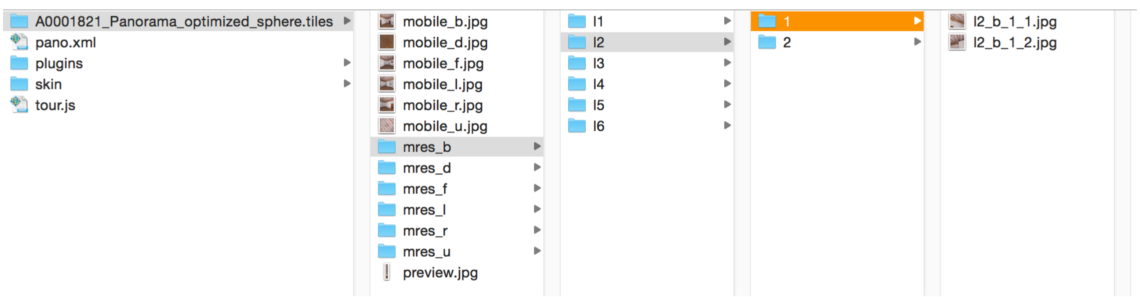


Figure 53. Generated files showing: Bottom Face -> Level 2 tiles

After the required tiles and the XML have been created (with the droplets or even manually with image editing programs), it is a matter of optimizing the viewer and the panorama. Creating panoramas can be fairly easy with the provided tools, but to fully optimize the panoramas for both performance and quality, a good understanding of basic concepts and the way the viewer plus the underlying technology works is needed.

The krpano HTML viewer has actions for controlling the view, projections, display quality, controls for mouse/touch events etc. Plugins can be used for extra functionality or the user can create his own. Understanding the fundamentals of panoramas and how they work presented in the earlier chapters make customizing and presenting the panoramas easier. For example, as seen in Code example 2, krpano offers many settings for controlling the view.

```
<!-- view settings -->
<view
   hlokat="0.0" vlookat="0.0" camroll="0.0" fovtype="VFOV" fov="90.0"
   fovmin="1.0" fovmax="80" maxpixelzoom="1.0" mfovratio="1.333333"
   fisheye="0.0" fisheyefovlink="0.5" stereographic="false" pannini="0.0"
   architectural="0.0" architecturalonlymiddle="true" limitview="auto"
   hlookatmin="" hlookatmax="" vlookatmin="" vlookatmax=""
 />
```

Code example 2.   Just a few of the initial settings provided for the viewer

All of these settings are just initial values and can be adjusted and set later with the JavaScript functions provided by the Interface object.

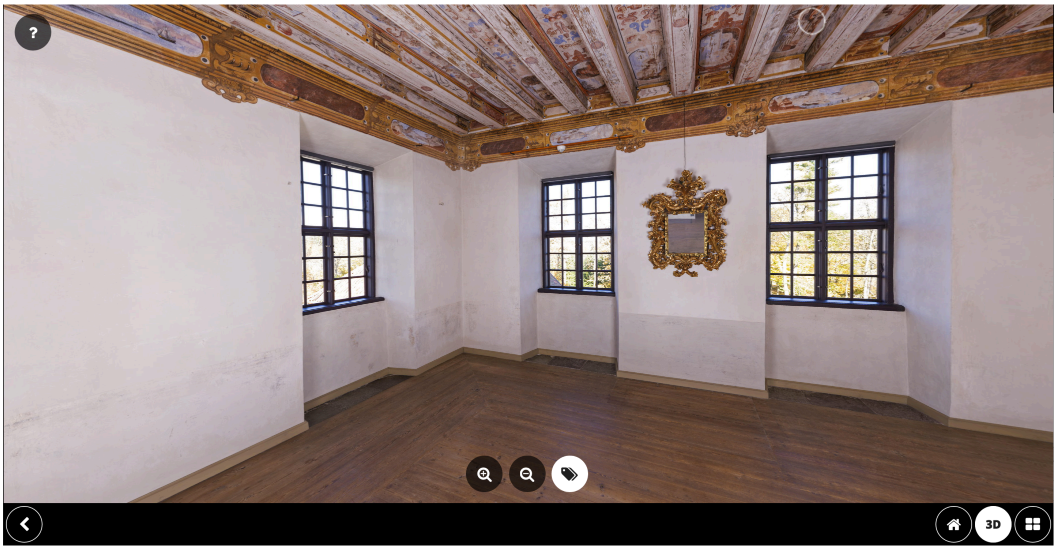


Figure 54.   Louhisaari 360° x 180° panorama screenshot from the application

After all of the initial settings have been adjusted, needed files and image tiles created; Figure 54 shows the optimized Louhisaari 360° panorama. The panorama has an initial and maximum field of view of 80° and the projection used is the normal rectilinear projection. The tiles have been optimized and compressed, still retaining the best possible quality.

7.4.2 Louhisaari Flat Multiresolution Panoramas

In addition to the 360˙ panorama, 19 flat multiresolution panoramas were created, with the multiresolution details shown in Table 5. The aspect ratio of the original images, being over twice as high as wide, proved to be a bit tricky, since it was not easily divided into POT textures and automatic tiling was generating non-optimized tiles. To optimize the tiling and division of the images for the multiresolutions, each image was carefully upscaled to a dimension of 2560 by 6144 pixels, with Abode Photoshop (www.adobe.com), “Preserve Details” upsampling algorithm. This provided a resolution dividable with a POT value, 256 pixels, to have more control over the quality of the levels. All of the other steps described earlier with the 360˙ panorama, apply also to the creation of the flat panoramas created of the allegorical figures.

Table 5. Multiresolution level details

Original Resolution:	2500 x 6000 (upscaled to) 2560 x 6144
Multiresolutions:	2560x6144, 1280x3072, 640x1536
Tilesize/Tiles:	256x256 = 10x24, 5x12, 3x6 = 363 Tiles (~3.6 MB Jpegs)

Just as with spherical panoramas, Krpano offers tools for creating the tiles and the XML / HTML template files for flat panoramas. [92] Since we are now dealing with flat panoramas, instead of spherical panoramas, the control settings and the view settings are a bit different, as seen in Code example 3. Krpano treats flat panoramas as cylindrical panoramas, with a horizontal field of view 1.0. Because of the image proportions, fitting the whole image to the viewer was a bit tricky and did not work with just adjusting the FOV and other basic values and needed additional scripting to work.

```

    <!-- view settings -->
    <view hlookat="0" vlookat="0" maxpixelzoom="1.0" fovmax="150"
        limitview="fullrange" fov="0.8" />

    <!-- control settings (drag2d for flat pano) -->
    <control mousetype="drag2d" touchtype="drag2d" zoomtocursor="true"
             zoomoutcursor="true" bouncinglimits="false"  />
    <contextmenu fullscreen="false" versioninfo="false" />

    <!-- preview pano image -->
    <preview url="infrapuna.tiles/preview.jpg" />

    <image type="CYLINDER" hfov="1.0" vfov="2.400000" voffset="0.00"
           multires="true" tilesize="256" progressive="true">
```

```
        <level tiledimagewidth="640" tiledimageheight="1536">
            <cylinder url="infrapuna.tiles/l3/%0v/l3_%0v_%0h.jpg" />
        </level>
        <level tiledimagewidth="1280" tiledimageheight="3072">
            <cylinder url="infrapuna.tiles/l2/%0v/l2_%0v_%0h.jpg" />
        </level>
        <level tiledimagewidth="2560" tiledimageheight="6144">
            <cylinder url="infrapuna.tiles/l1/%0v/l1_%0v_%0h.jpg" />
        </level>
    </image>

    <!-- events and actions for adjusting the view settings to
         see the whole image and sycronizing the views -->
    <events name="flatpano_events" onnewpano="flatpano_imagefit();"
            onresize="flatpano_imagefit();" />

    <action name="flatpano_imagefit">
        if(image.vfov GT 0,
            div(aspectratio, stagewidth, stageheight);
            if(aspectratio GE 1,
                if(image.vfov GE image.hfov, set(view.fovtype,VFOV),
                                             set(view.fovtype,HFOV));
              ,
                if(image.vfov GE image.hfov, set(view.fovtype,HFOV),
                                             set(view.fovtype,VFOV));
             ););
    </action>

    <events onmousedown="set(syncother,true); events.onviewchange();
                         if(syncother, js( sync_2to1() ) );"
            onmousewheel="events.onmousedown();"
            onviewchange="if(syncother, js( sync_2to1() ) );"/>

```

Code example 3.   Flat panorama initial values

For each of the allegorical figures, two flat multiresolution panoramas were created. One showing the visible light image, such as the figures Bacchus and Piety in Figures 55 A/C and the other presenting one of the special techniques used, such as IR photography as seen in Figure 55C and false color IR photography as seen in Figure 55D.

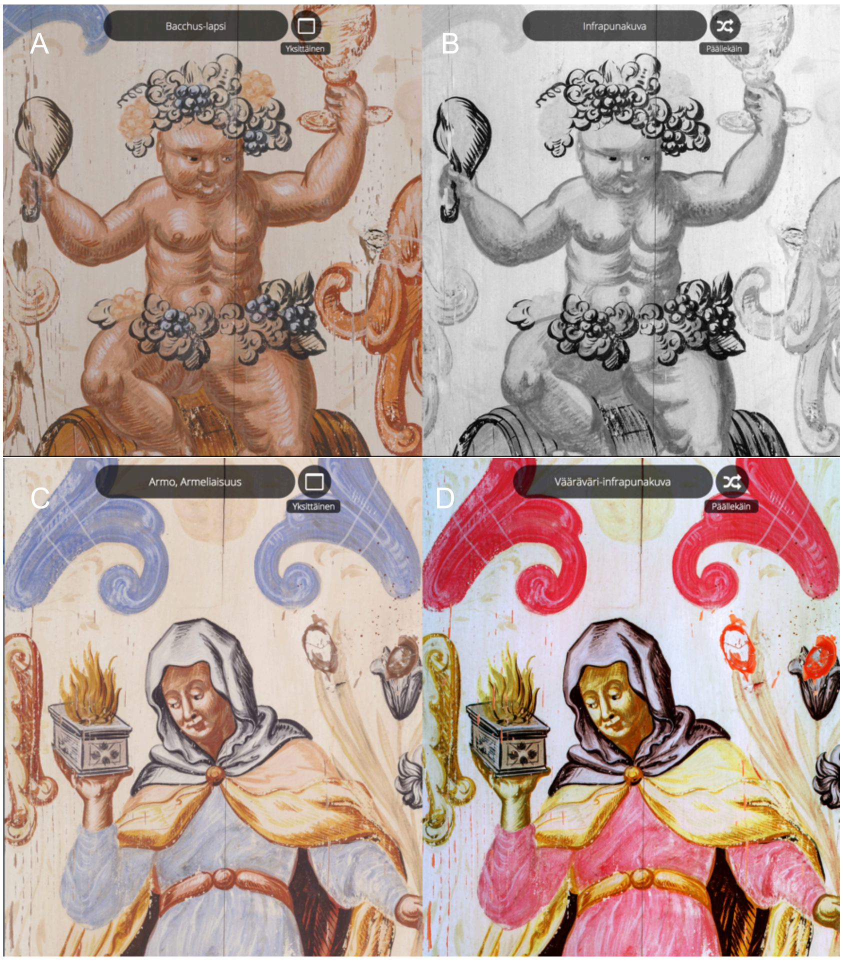


Figure 55. Flat panoramas created from the figures Bacchus is A/B and Piety C/D

All of the panoramas and the descriptions of the figures are listed in Appendix 1 for further study.

7.4.3 Projections

With the HTML5 viewer, multiple different projections are supported. Different projections can be controlled, by changing the krpano view values, such as the fisheye parameter. The viewer then makes the needed calculations for warping and displaying the projected image. Figure 56 displays all of the projections provided and supported by the krpano viewer.

Figure 56. 3D Projections supported by the viewer

Different projections cause different distortion. With rectilinear projection, the view starts to stretch at the edges with wide viewing angles, so the view angle should be kept under 100˚ to avoid the effect. Other projections provided by the viewer are the fisheye, stereographic (extreme fisheye), architectural, pannini and little planet projections, as seen in Figure 56, but all of the projections apart from the normal projection distort the panorama and were discarded as alternatives for the application. For the Louhisaari panorama, the normal projection was chosen as this yields the best, natural result for viewing.

7.4.4 Hotspots

A hotspot is an area in the panorama that reacts when the user hovers over it or clicks to activate some behavior. Hotspots can be used for linking other panoramas or pages, changing the view or any other behavior. Images and polygonal (set of points that define an area) hotspots can be used and animated, as seen in Figure 57 where the hotspots are visible as white circles. [98] Using the onclick event, hotspots can trigger JavaScript functions. For the 360˚ panorama, 10 hotspots were create, one for each figure. At this stage of the project, the hotspots did not still trigger any behavior, this was added later, when the application logic was implemented.

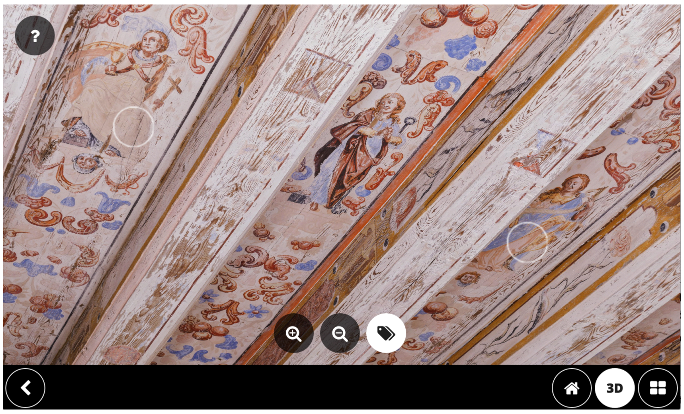


Figure 57. Two hotspots for the figures

As seen in Code example 4, the hotspots are defined with variables ath and atv.

```
<hotspot name="anteliaisuus" style="hs_ani" ath="-104.089555" atv="-
35.03758" />
<hotspot name="armo" style="hotspot_ani" ath="169.355149" atv="-
32.515306" />
```

Code example 4.

These are the spherical coordinates in degrees for the hotspot, ath from -180 to +180 and atv from -90 (zenith) to +90 (nadir), just as the mapping of the projected graticule of a sphere, explained in Chapter 4.3.2. Spherical 360˚ Panoramas, see Figure 58. [99]

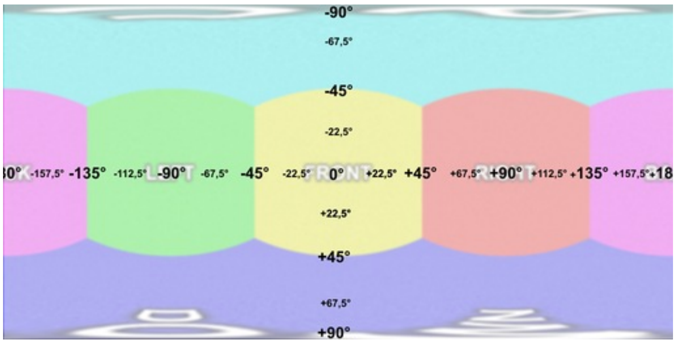


Figure 58. krpano coordinate system [99]

Now that both the optimized 360˚x180˚ multiresolution panorama and the 19 flat, multi-resolution panoramas were created, optimized and the hotspots and other needed behaviors created, it was time to put it all together. More about the Louhisaari Application in the next chapter.

8 Louhisaari Application Development

The “Louhisaaren Kartanonlinna – Juhlasalin Kattomaalaukset” Application is based on standard web technologies - HTML5, CSS, JavaScript and WebGL - and can be used and accessed by anyone using a browser that supports these technologies. The current version of the application can be accessed in the following address:

http://louhisaari.metropolia.fi

Soon after the final release, May 2016, the link to the Louhisaari application can be found at:

http://www.kansallismuseo.fi/

In the heart of the application is the krpano HTML5 viewer and the provided panoramas wrapped together for a seamless experience. From the user’s perspective, the Louhisaari App is a browser-based application that works without the need to install plugins and on most of the modern browsers on almost all platforms and devices. The App is designed with usability first and a lot of detail has been put into user-centered design (UCD), information architecture and application logic to make the application usable, easy, accessible and effective for anyone to use.

8.1 Design

Although seen as the last phase of the development, in Figure 59, the application design actually followed the panorama creation phases in parallel. This means that every choice made during the shooting, stitching and panorama rendering phases took the final application into account and further formed the final application idea and implementation. This chapter covers this last stage of development and explains the technologies used and how the application logic was implemented.

Figure 59. Last Phase: Application Development

Many of the choices regarding the technologies used have been presented in the preceding chapters. The 360˚ and the flat panoramas were carefully designed and optimized so that they could finally be wrapped into a fully functional Application with additional info and text provided. The flow and the logic of the application can been seen in Figure 60 with screenshots from the final Louhisaari Application.

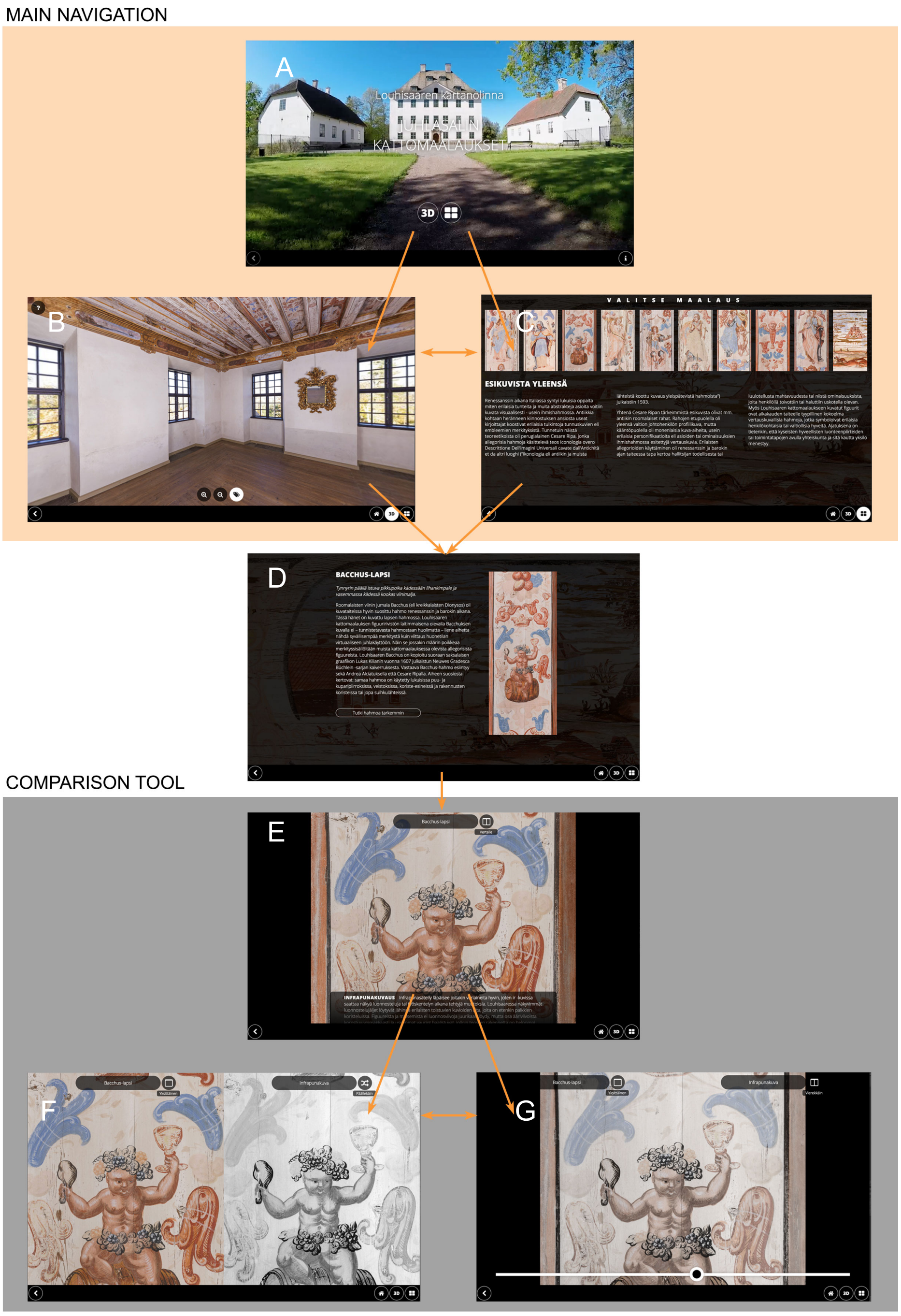


Figure 60. A flow chart of the “Louhisaari Manor Ceiling Paintings” Application. A: Landing Page B: 360 Panorama C: Gallery D: Figure Info E: Flat panorama F: Panorama Comparison G: Panorama Blending

The main navigation is located in the bottom of the page and is accessible during every state of the application. It has 4 different buttons. 3 to the main categories: Home, 3D and Gallery and the Back button, see Figure 61.

Figure 61. Main Navigation

The Application provides the user the following pages/states, as seen in Figure 60. (Descriptions of each page do not include the navigation bar, which is always accessible):

A. **Landing Page** – When the user opens the application, this page is shown. A specially made coptercam video from Louhisaari is running in fullscreen mode in the background (if the users browser is compatible with fullscreen video). The user can navigate to the 360˚ panorama or to the gallery by clicking on the provided buttons.
B. **360˚ Panorama** – Here the user can fully enjoy, zoom and study the 360˚ multiresolution panorama of the banquet hall. The application provides hotspots (that can be also disabled for better viewing of the panorama) for some of the figures depicted on the ceiling and the user can navigate to the “Figure Info” page by clicking on the hotspots.
C. **Gallery** - A gallery, listing all of the figures that are available to study through this application. By clicking on the selected figure in the gallery mosaic, the user is guided to the “Figure Info” page. A short description of the paintings is also shown.
D. **Figure Info** - This page contains a short description of the selected figure with a small thumbnail of the flat panorama. The user can navigate to the first page of the comparison tool to study the flat panoramas by clicking on the “Study the Figure” button.
E. **Flat panorama** – The first page of the comparison tool. The user is provided a single flat multiresolution panorama of the chosen figure, shot in visible light. Additional info of the documentation method used for the figure is also shown. The user can fully zoom and study the multiresolution panorama of the figure in great detail. Clicking on the “Compare” button provided on right hand side of the

figure title, the user will notice that an additional panorama slides next to the visible light panorama.

F. **Panorama Comparison (state)** – This is not actually a page. When the user has clicked on the “Compare” button an additional flat panorama will slide from the right and the ”page” shows two panoramas, each taking 50% of the space. The panorama to the left is the flat panorama shot with visible light and the panorama to the right the pixel by pixel aligned panorama showing the same figure with the chosen documentation method. These two panoramas are synced together. If the used moves either one or zooms in, the other panorama is synced and will move/zoom accordingly. The user has the option to return to previous state by clicking on the “Single” button or to click on the “Blend”.
G. **Panorama Blending** – If the user clicks on the “Blend” button, the synced panoramas will now slide on top of each other and occupy the full 100% size of the browser. An additional slider is shown. By dragging the slider to the right or to the left the user can blend these synced panoramas together, right showing 100% of the documentation method panorama and the left showing 100% of the visible light panorama.

The above listing only explains the logic of the application, the technologies used are presented in following Chapter 8.3 Full Implementation.

8.2 First phase: Comparison Tool and Application Logic

Before the full application logic was ready, early trials of syncing two panoramas together were made. As soon as the first panoramas were created, this was the first step, the starting point in building the application.

The first phase of the application design was determined by one of the main goals: To compare high-resolution images in real-time and therefore to build a special comparison tool.

The aim was to build a tool where the user could study detailed high-resolution images taken with normal visible light, paired with a pixel-by-pixel matching image taken with special lighting to give users an opportunity to study and compare them. Since comparing two conventional high-resolution images in real-time had its limitations and multiresolution technology was chosen to provide an easy access to the files the problem that needed to be solved was:

How to synchronize two panoramas together, especially multiresolution panoramas? And if they can be synced, is there a way to blend these together?

Surprisingly, the end solution was actually very simple, but to come up with the solution took months of studying and trials. So far, at the time of writing, I have not seen one implementation yet where syncing and blending of two multiresolution panoramas has been implemented.

The krpano viewer offers examples how to blend two panoramas together, with a slider example. [101] The problem is that this is not true blending of two panoramas. The slider example is using distorted hotspots and changes the alpha transparency of the hotspots for the blending. Normal or multiresolution panoramas can be only blended on loading another pano and using the blend() setting. [102] This means that to blend two panoramas together, one has to jump from one panorama to another and back and the as the new panorama loads, the view can be blended. This seemed very discouraging, as krpano offers now ways of blending of two panoramas internally.

As the search continued, I started studying the syncing of the panoramas instead. The krpano viewers syncing feature is not in any way documented in their XML reference list, but by studying their pages a syncscreen example was found. At that moment, I realized that there is also a way to use this achieve the blending too.

The idea behind the syncing of two panoramas is quite simple. Instead of embedding only one panorama, both of the panoramas are embedded on a HTML page. An on-mousedown event can be used to call a function that synchronizes both the vertical and horizontal looking directions and the current field of view of the panorama to another. The XML Code example for the event is shown in Code example 4.

```
<events onmousedown="set(syncother,true); events.onviewchange();if(syncother, js( sync_2to1() ) );"
    onmousewheel="events.onmousedown();"
    onviewchange="if(syncother, js( sync_2to1() ) );"
    />
```

Code example 4.   Krpano.xml event for synchronizing two panoramas

The event must used in both of the panoramas. In the first panorama, sync_1to2() is called, and in the second one, sync_2t01() is called on a mousedown event, see Code example 5 for the functions.

```
// sync viewer movement
function sync_1to2()
{
    var krpano1 = document.getElementById("krpanoLeftViewer");
    var krpano2 = document.getElementById("krpanoRightViewer");

    krpano2.set("syncother",    false);
    krpano2.set("view.hlookat", krpano1.get("view.hlookat"));
    krpano2.set("view.vlookat", krpano1.get("view.vlookat"));
    krpano2.set("view.fov",     krpano1.get("view.fov"));
}

function sync_2to1()
{
    var krpano1 = document.getElementById("krpanoLeftViewer");
    var krpano2 = document.getElementById("krpanoRightViewer");

    krpano1.set("syncother",    false);
    krpano1.set("view.hlookat", krpano2.get("view.hlookat"));
    krpano1.set("view.vlookat", krpano2.get("view.vlookat"));
    krpano1.set("view.fov",     krpano2.get("view.fov"));
}
```

Code example 5. – Sync viewer movement functions

Now the two panoramas are loaded onto the page. If the user moves or zoom either of the panoramas, the other panorama will move accordingly. Since the panoramas are typically embedded into basic HTML DOM elements, as seen in previous code examples, such as inside <div> elements, it is just a matter of moving and blending the DOM elements together with basic CSS to achieve the desired effect.

In Figure 62, the visible, rendered browser window is shown as an orange rectangle. When the user enters the “Flat panorama” page, she is only shown the <div> element containing the visible light panorama, as seen in Figure 62 A. The second panorama is hidden by moving the containing <div> to the side of the screen. When comparing the panoramas, the hidden div slides back to the visible are and now both of the <div> elements occupy 50% of the screen. The best part of this approach is that it can be used for blending. By sliding both of the <div> elements and panoramas on top of each others and changing the opacity level of the topmost <div> element with CSS, blending is achieved. The opacity-level describes the transparency-level, where 1 is not transparent at all, 0.5 is 50% see-through, and 0 is completely transparent. With a slider, the user can be given a chance to change the opacity (transparency) of the topmost <div> from 0 to 1.

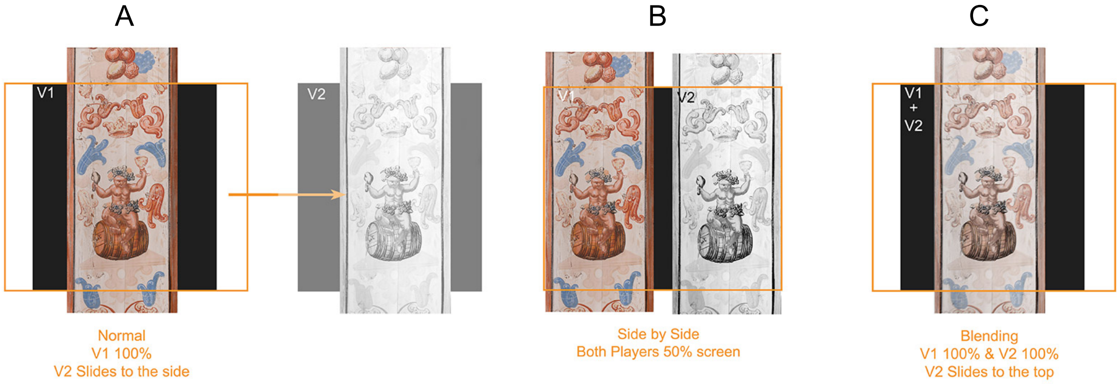


Figure 62. The logic in syncing, comparing and blending two panoramas. V=Viewer

To achieve precise blending, it must be noted, that the images used and panoramas generated must match in terms of resolution and content. If two images or panoramas are blended together and are not identical, it is easy to see small artefacts and ghosting of the lines and the blended image looks blurry and mismatched. Therefore, in the early stages of panorama creation, the images of the figures were shot with top precision and preprocessed to match. Figure 63 shows screenshots of the comparison tool and the different ways these figures can be studied, compared and blended.

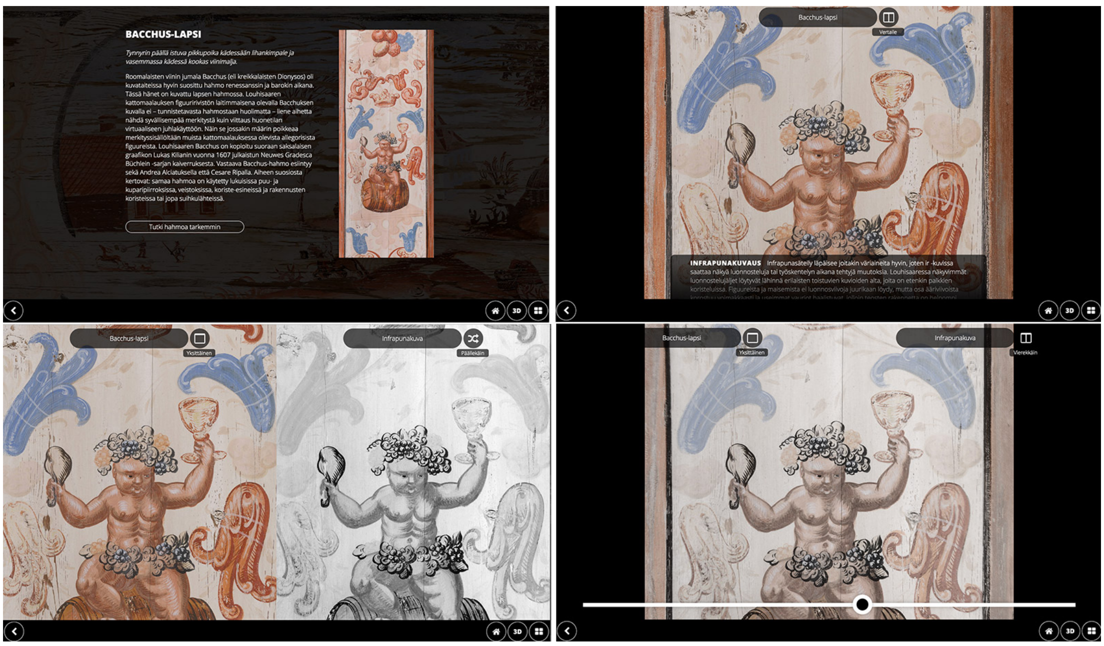


Figure 63. The Comparison Tool - Sceenshots

The comparison tool is a great tool to interactively study the figures and different documentation methods used. All of the flat panoramas are interactive, responsive and zoomable. Even the brush strokes of the artist can be seen in these high-resolution panoramas.

8.3 Second Phase: Full Implementation

At this stage of the application design, it was evident that the comparison and the blending would fully work with the multiresolution panoramas generated. Now, the flat panoramas, 360˚ panorama and the comparison tool needed to be built into a fully functional app. To pull it all together, a group of Media Engineering students from the Metropolia University of Technology: Ilpo Oksanen, Joonas Mononen and Kati Kallioniemi joined the project during the spring 2015 and have been involved in programming and developing the application in many ways. To build the app, the group worked closely together and all of the aspects of the design, usability and logic were discussed and agreed on. As the project is now finally ending, they are working with the different language options, soon implemented in the final application.

The application has been designed with usability first and to work on multiple platforms. Therefore it utilizes the jQuery JavaScript library (https://jquery.com/). jQuery is a fast, small and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. [104] Some of the reasons to choose this library were; it is open source, cross-platform and it uses HTML, CSS and JavaScript and therefore easily integrated to work with the generated panoramas and files.

The Louhisaari App:

- Utilizes jQuery
- Provides a set of touch-friendly UI widgets
- jQuery-powered navigation system
- Animated page transitions
- Follows the Visual Guidelines of the National Museum of Finland
- Uses scalable vector icons with the Font Awesome toolkit

(https://fortawesome.github.io/Font-Awesome/)

- Is fully responsive
- Is designed for usability and optimal performance
- Is designed for further development such as wrapping into a hybrid mobile app

As stated above, since the application was developed using HTML5, CSS and JavaScript (jQuery) with a few changes it could be later wrapped into a native iOS or Android mobile app. At the time of writing, this implementation of the application is left for further development.

8.4 Future Implications

At the time of writing, the online version of the Louhisaari Application is almost finished and soon to be published. The release is due May 2016. There are still a few issues that needs to be solved and changed to be made, before the Application can be published:

Language options
The content of the Application is currently in Finnish. Translations have been made for both English and Swedish and these will be implemented to the final app.

Storage and Preservation
The Application runs currently on a server owned and maintained by Helsinki Metropolia University of Applied Sciences. The final application will be placed on a server maintained by the National Museum of Finland and linked to their main website. Therefore additional testing and tweaking needs to be made to ensure that the Application runs fine and error free on the new server environment.

Security
When the final Application is transferred to the final destination, custom krpano viewer files will be created to provide special access / content protection. The krpanotools encrypt tool can be used to encrypt files for the krpano viewer. As key for the encryption the registered license will be used. That means only a viewer file with the same embedded licenses will be able to load / to decrypt the encrypted files. [105] The XML files

will be encrypted, as well as allowing the viewer to load xml and plugins files only from the same domain.

Analytics

Integrate Google analytics to the application to follow the app users and to further evaluate the performance of the application. [106]

9 Conclusions

This thesis explored the possibility of using high-quality gigapixel sized, multiresolution panoramas as way to study, access, present and share high-resolution documentary image material created during the Louhisaari Conservation Project. The project approach focused on retaining the best possible quality and accuracy of the generated panoramas, on user experience and the use of panoramas as the core of an application.

The Louhisaari Application project has been a lengthy but an inspirational journey into art and technology. High-resolution digital photography combined with digital imaging techniques, panorama technology and web technologies has given insight to the complexity of creating and retaining high fidelity and quality panoramas. A few years ago, this project would still have been impossible to carry through. The latest developments in both panorama photography and web technologies has provided us new ways of exploring and studying paintings and accessing high-resolution image data, more detailed than ever before.

Creating digital panoramas and wrapping the panoramas in an application is a lengthy process with multiple steps involved. Choices in the data-acquisition phase, the panorama photography, affect the stitching, rendering and the viewing of the panoramas. And vice versa. Creating panoramas can be a fairly easy process, with automated hardware and software, but as this study shows, as the requirements for quality and accuracy rise, the process get more complicated and the overall skill of the content provider is the ultimate determining factor of the quality of the final panorama and the app. It fascinating, as this study shows, that with multiresolution technologies, it is now possible to access even gigapixel-sized panoramas with a common browser, use them real-time, even in mobile devices.

Choosing the right technologies and equipment can be a daunting task both for the photographer and the web application developer. The aim of this thesis was to explore these choices and explain the choices made during the Louhisaari Application project. As the project progressed, it was evident that the field of panorama technologies is vivid, growing and expanding. Understanding the basic concepts of panoramas, such as projections, distortion, multiresolution technology and the workflow in general is crucial. With the knowledge gained, it is easier to create panorama applications, no matter what the chosen technology or implementation is. With the right choices, gigapixel sized panoramas, utilizing common web technologies can be fully integrated into a seamless working application.

The future of creating high-resolution gigapixel panoramas looks interesting and promising. It is now possible to create and view even tera- and petapixel sized panoramas and it looks like the sky is the limit. This technology could also be used for any imagery, not just what we regard as panoramas. As the sensor technology skyrockets and the size of images get bigger and bigger, this approach could be implemented and used for any ultra-high resolution imagery.

The project has been a success and will soon been fully published for everyone to see. User feedback has been encouraging and the development of the application will be continued and the application optimized when finally on the final server location. The future looks promising, as this project has yielded a new similar project with the National Museum of Finland of another important historical site.

References

1 Big Data, for better or worse: 90% of world's data generated over last two years
http://www.sciencedaily.com/releases/2013/05/130522085217.htm
Accessed 13.2.2016

2 Smithsonian Museum Conservation Institute – Infrared and Ultraviolet Imaging
http://www.si.edu/MCIImagingStudio/IR_UV
Accessed 4.3.2016

3 Mohanty, Hrushikesha. Big Data. Springer 2015

4 Louhisaari Manor
http://www.kansallismuseo.fi/en/louhisaari-manor
Accessed 1.3.2016

5 Louhisaaressa tutkitaan ja dokumentoidaan 1600-luvun kattomaalauksia
http://www.nba.fi/fi/ajankohtaista/tiedotearkisto?Article=5924
Accessed 20.10.2015

6 A. Wallert, M. Rey. Historical Painting Techniques, Materials and Studio Practice
http://www.getty.edu/conservation/publications_resources/pdf_publications/pdf/historical_paintings.pdf
Accessed 20.3.2016

7 Mika Seppälä, Louhisaari. 2015. Photograph.

8 Discussion with Jouni Kuurne, the National Museum of Finland
17.5.2015

9 Katso ja tutki ainutlaatuisia Louhisaaren kartanolinnan kattomaalauksia multi-resoluutiopanoraamassa
http://www.metropolia.fi/ajankohtaista/uutiset/?tx_ttnews%5Btt_news%5D=5182&cHash=67d3606df6f6b6a2699601faf1c2d4af
Accessed 15.9.2015

10 Shyam J. Dhoble, Prof. Nitin Shelke. Investigative Research on Big Data: An Analysis - IJIRSET
http://www.ijirset.com/upload/2015/june/58_12_Investigative.pdf
Accessed 1.2.2016

11 J. Gantz. The Digital Universe in 2020.
https://www.emc.com/collateral/analyst-reports/idc-the-digital-universe-in-2020.pdf
Accessed 1.2.2016

12 James J. Jong Hyuk Park, Yi Pan, Cheon-Shik Kim, Yun Yang. Future Information Technology. Springer Science & Business, 3.5.2014.

13 Unstructured Data
http://www.webopedia.com/TERM/U/unstructured_data.html
Accessed 25.3.2016

14 Discussion with Mika Seppälä, Metropolia University of Applied Sciences 4.5.2015

15 Krpano HTML5 Viewer
http://krpano.com/docu/html5/
Accessed 1.3.2016

16 Introduction to User-Centered Design
http://www.usabilityfirst.com/about-usability/introduction-to-user-centered-design/
Accessed 2.2.2016

17 Frich, Arnaud. Panoramic Photography. Focal Press, 2007.

18 Panorama
http://www.dictionary.com/browse/panorama
Accessed 1.3.2016

19 Harald Woeste. Mastering Digital Panoramic Photography. Rocky Nook 2009.

20 Panorama of London Barker
https://commons.wikimedia.org/wiki/File:Panorama_of_London_Barker.jpg
Accessed 1.3.2016

21 A brief history of Panoramic Photography
https://www.loc.gov/collections/panoramic-photographs/articles-and-essays/a-brief-history-of-panoramic-photography
Accessed 23.2.2016

22 [San Francisco, 1851, from Rincon Hill] digital file from intermediary roll film copy
https://www.loc.gov/resource/pan.6a01850
Accessed 1.3.2016

23 IAETH Best Practices Guide to Digital Panoramic Photography
http://www2.iath.virginia.edu/panorama/print/version2_pdf/PanoGuide.pdf
Accessed 5.1.2016

24 The forgotten ancestors of virtual reality
http://www.indestry.com/blog/2016/1/7/the-forgotten-ancestors-of-virtual-reality
Accessed 23.3.2016

25 A History of Panoramic Image Creation
http://www.graphics.com/article-old/history-panoramic-image-creation
Accessed 1.3.2016

26 Panoramic Photography
https://site.douban.com/195886/widget/notes/11812927/note/252569163/
Accessed 1.3.2016

27 Understanding Your Camera: Focal Length, Field of View and Angle of View Defined

http://martybugs.net/blog/blog.cgi/learning/Field-Of-View-And-More.html
Accessed 2.2.2016

28 Cameras vs. the human eye
http://www.cambridgeincolour.com/tutorials/cameras-vs-human-eye.htm
Accessed 2.1.2016

29 Hans Strasburger; Ingo Rentschler; Martin Jüttner. Peripheral vision and pattern recognition: A review. 2011

30 Francis Harvey. A Primer of GIS. Fundamental Geographic and Cartographic Concepts. The Guildford Press. 2008.

31 What is Distortion
https://photographylife.com/what-is-distortion
Accessed 6.1.2016

32 Distortion
http://toothwalker.org/optics/distortion.html
Accessed 3.3.2016

33 Rectilinear Wide-Angle Lenses and Fish-eye Lenses
http://www.unc.edu/~johnsonc/Fisheye_csj.pdf
Accessed 23.3.2016

34 The Guide to Panoramic Photography
http://www.panoramic-photo-guide.com/virtual-tour-360-photography/how-to-choose-photo-lens-virtual-tour.html
Accessed 1.3.2016

35 Projections
https://www.ptgui.com/man/projections.html
Accessed 2.2.2016

36 What are map projections
https://www.e-education.psu.edu/geog160/node/1918
Accessed 3.3.2016

37 Latitude and Longitude
http://global.britannica.com/science/latitude
Accessed 6.2.2016

38 Map Projections - types and distortion patterns
http://geokov.com/education/map-projection.aspx
Accessed 15.3.2016

39 The equirectangular projection with Tissot's indicatrix of deformation
https://en.wikipedia.org/wiki/Equirectangular_projection#/media/File:Tissot_indicatrix_world_map_equirectangular_proj.svg
Accessed 5.1.2016

40 Equirectangular Projection
http://wiki.panotools.org/Equirectangular_Projection
Accessed 23.2.2016

41 Cylidrical Projection
http://mathworld.wolfram.com/CylindricalProjection.html
Accessed 2.2.2016

42 Snyder, J.P. (1989). Album of Map Projections, United States Geological Survey Professional Paper. United States Government Printing Office. 1453.

43 WebVR Plugin
http://krpano.com/plugins/webvr
Accessed 5.3.2016

44 Multiresolution Panoramas
http://krpano.com/docu/multires/
Accessed 10.12.2015

45 krpano HTML5 Viewer
http://krpano.com/docu/html5/?version=118
Accessed 2.2.2016

46 Bubblepix
https://bubblepix.com/
Accessed 23.3.2016

47 Gigapan
http://www.omegabrandess.com/Gigapan
Accessed 23.3.2016

48 Panono Panoramic Ball
https://www.panono.com
Accessed 23.3.2016

49 Photo Stitching Digital Panoramas
http://www.cambridgeincolour.com/tutorials/digital-panoramas.htm
Accessed 15.1.2016

50 Heads
http://www.panoramic-photo-guide.com/panoramic-heads-to-make-panoramic-photography.html
Accessed 23.3.2016

51 Gigapan Epic Pro Introduction
http://gigapan.com/cms/manuals/epic-pro-introduction#intro2
Accessed 23.3.2016

52 S.J. Pise. ThinkQuest 2010: Proceedings of the First International Conference on Contours of Computing Technology. Springer Science & Business Media, 2011.

53 QTVR
https://www.ptgui.com/info/qtvr.html
Accessed 3.3.2016

54 Panorama Tools
http://www.panoramatools.com
Accessed 12.12.2015

55 Panorama Tools
https://www.ptgui.com/panotools.html
Accessed 12.12.2015

56 Hugin - Panorama photo stitcher
http://hugin.sourceforge.net/
Accessed 2.2.2016

57 Stitching Gigapixel Panoramas
https://www.youtube.com/watch?v=VAc7rjnjr3Q
Accessed 20.3.2016

58 Image Stitching – Computational Photography, Derek Hoiem, University of Illinois
https://courses.engr.illinois.edu/cs498dwh/fa2010/lectures/Lecture%2017%20-%20Photo%20Stitching.pdf
Accessed 20.3.2016

59 Introduction to SIFT (Scale-Invariant Feature Transform)
http://docs.opencv.org/3.1.0/da/df5/tutorial_py_sift_intro.html#gsc.tab=0
Accessed 20.3.2016

60 Richard Szeliski. Image alignment and Stitching: A Tutorial
http://sse.tongji.edu.cn/linzhang/ComputerVision/Projects/Image%20alignment%20and%20stitching%20a%20tutorial.pdf
Accessed 20.3.2016

61 What is Vignetting
https://photographylife.com/what-is-vignetting
Accessed 5.12.2015

62 Main Window: Create Panorama tab
http://www.ptgui.com/ptguihelp/main_createpanorama.htm
Accessed 13.3.2016

63 Enblend
http://enblend.sourceforge.net/
Accessed 23.3.2016

64 SmartBlend
http://wiki.panotools.org/SmartBlend
Accessed 23.3.2016

65 Stitch 2.0, Stitch.Efx, and Upload
http://gigapan.com/cms/manuals/epic-pro-stitching-and-uploading-panoramas
Accessed 23.3.2016

66 PTGui Support
https://www.ptgui.com/support.html#2_16
Accessed 2.2.2016

67 PTGui Options / Preferences
http://www.ptgui.com/ptguihelp/options.htm
Accessed 2.2.2016

68 Adobe Photoshop File Formats Specification
https://www.adobe.com/devnet-apps/photoshop/fileformatashtml
Accessed 2.2.2016

69 Web3D
http://www.web3d.org
Accessed 2.2.2016

70 VRML
https://www.w3.org/MarkUp/VRML
Accessed 2.2.2016

71 X3D
http://www.web3d.org/x3d/what-x3d
Accessed 31.1.2016

72 Java 3D
https://java3d.java.net
Accessed 31.1.2016

73 Josh Juneau, Mark Beaty, Carl Dea, Freddy Guime, John OConner.. Java 3D Recipes: A Common Problem-Solution Approach. Apress. 2012.

74 3D in the Browser: WebGL versus CSS 3D Transforms
http://blog.teamtreehouse.com/3d-in-the-browser-webgl-versus-css-3d-transforms
Accessed 13.3.2016

75 Three.js
http://threejs.org
Accessed 13.3.2016

76 Animate your HTML
http://animateyourhtml5.appspot.com
Accessed 23.3.2016

77 OpenGL ES 2.0 for the Web
https://www.khronos.org/webgl
Accessed 13.3.2016

78 Getting a WebGL Implementation
https://www.khronos.org/webgl/wiki/Getting_a_WebGL_Implementation
Accessed 1.3.2016

79 “Can I Use” - WebGL - 3D Canvas graphics
http://caniuse.com/#feat=webgl
Accessed 1.3.2016

80 CSS Transforms Module Level 1
https://www.w3.org/TR/css3-transforms/
Accessed 1.3.2016

81 "Can I Use" – CSS3 3D Transforms
http://caniuse.com/#feat=transforms3d
Accessed 1.3.2016

82 Heung-Yeung Shum, Shing-Chow Chan, Sing Bing Kang. Image-Based Rendering. Springer Science & Business Media, 2008

83 Heung-Yeung Shum and Sing Bing Kang, Microsoft Research. A Review of Image-based Rendering Techniques.
http://research.microsoft.com/pubs/68826/review_image_rendering.pdf
Accessed 1.3.2016

84 Manuel M. Oliveira. Image-Based Modeling and Rendering: A Survey
http://www.inf.ufrgs.br/~oliveira/pubs_files/RITA_IBMR.pdf
Accessed 1.3.2016

85 David D. Zhang, Mohamed Kamel, George Baciu. Integrated Image and Graphics Technologies. Springer Science & Business Media, 2006.

86 Computer Graphics - The Plenoptic Function
http://resources.mpi-inf.mpg.de/departments/d4/teaching/ws200708/cg/slides/CG21-IBR.pdf
Accessed 1.3.2016

87 Panorama viewers
http://wiki.panotools.org/Panorama_Viewers
Accessed 15.12.2015

88 krpanotools Makepano Tool
http://krpano.com/tools/kmakemultires/
Accessed 15.12.2015

89 HTML5 Panoramic Video Support - Cube Map and/or Equirectangular
http://krpano.com/forum/wbb/index.php?page=Thread&threadID=10939
Accessed 2.4.2016

90 Set the krpano HTML5 Viewer usage
http://krpano.com/docu/html/#html5
Accessed 3.3.2016

91 Converttocube
http://krpano.com/tools/kmakemultires/config/#converttocube
Accessed 22.3.2016

92 krPano Droplets
http://krpano.com/tools/droplets/
Accessed 1.2.2016

93 krpano XML Reference - Image
http://krpano.com/docu/xml/#image
Accessed 24.3.2016

94 krpanotools Config File Reference - Tilesize
http://krpano.com/tools/kmakemultires/config/#tilesize
Accessed 3.3.2016

95 WebGL and OpenGL Differences
https://www.khronos.org/webgl/wiki/WebGL_and_OpenGL_Differences#Non-Power_of_Two_Texture_Support
Accessed 23.1.2016

96 krpano XML Reference - Memory
http://krpano.com/docu/xml/#memory
Accessed 3.3.2016

97 Chrome DevTools
https://developers.google.com/web/tools/chrome-devtools/
Accessed 24.3.2016

98 krpano XML Reference - Hotspot
http://krpano.com/docu/xml/#hotspot
Accessed 12.12.2015

99 Hotlinks to picture within a panorama
http://krpano.com/forum/wbb/index.php?page=Thread&threadID=1914
Accessed 2.2.2016

100 Embedding into HTML
http://krpano.com/docu/html/#top
Accessed 12.12.2015

101 Krpano 1.18 Examples
http://krpano.com/examples/118/examples/examples.html?html5=prefer
Accessed 28.3.2016

102 Discussion Forum: How can I blend multiresolution panoramas
http://krpano.com/forum/wbb/index.php?page=Thread&threadID=11069
Accessed 28.3.2016

103 Js Syncscreen Example
http://krpano.com/examples/118/examples/javascript-interface/js-sync/syncscreen.html
Accessed 28.3.2016

104 jQuery
https://jquery.com
Accessed 28.3.2016

105 krpanotools Encrypt Tool
http://krpano.com/tools/kencrypt
Accessed 28.3.2016

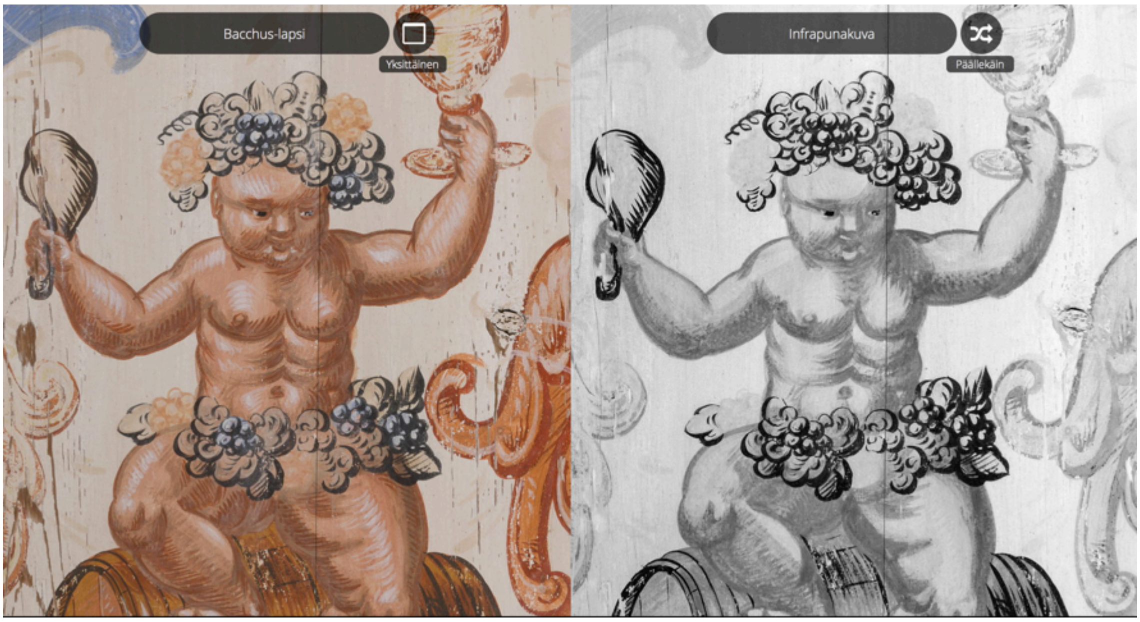
106 Google Analytics 360 Suite
https://www.google.com/analytics/360-suite/analytics/#?modal_active=none
Accessed 28.3.2016

107 Discussion with Mika Seppälä, Metropolia University of Applied Sciences 2.3.2016

Allegorical Figures and Special Documentation Methods

Infrared photography - Bacchus

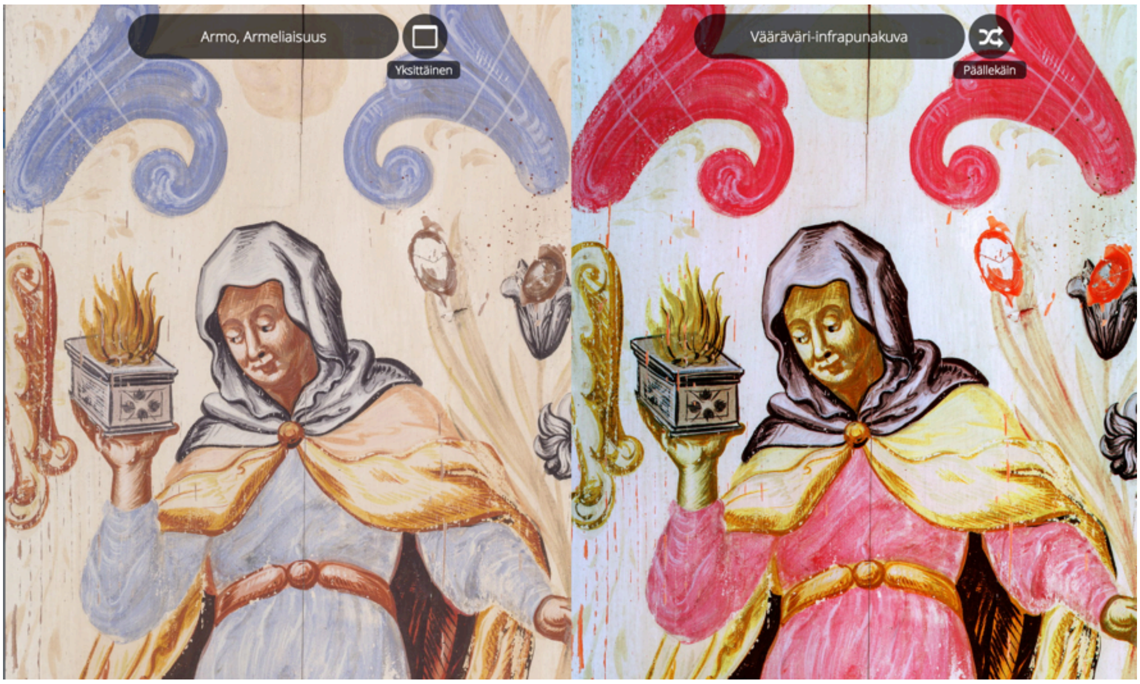
A small boy sitting on a barrel, holding a piece of meat and a large cup of wine. The Roman god of wine, Bacchus, was a popular character in Renaissance and Baroque art. Here he is depicted as a child. Despite the recognisable figure, Bacchus, the outermost depiction in the row of figures in the painting at Louhisaari, should not be considered to hold any deeper meaning than a reference to the festive use of the room space. Therefore, its meaning is slightly different from the other allegorical figures included in the painting. The Louhisaari Bacchus was copied directly from an engraving by the German graphic artist Lukas Kilian, published in 1607 as part of the Neuwes Gradesca Büchlein series. A similar Bacchus figure is included in Cesare Ripa's work. The fact that the same figure has been used in innumerable wood and copper engravings, sculptures, decorative items, building decorations and even fountains speaks to its popularity. [8]



Infrared radiation penetrates some pigments well. Therefore outlines and changes made while working might turn up on IR pictures. In Louhisaari the most visible outlines are mostly found on repeating pictures. These are mostly to be found on the beams. There are almost no draft lines on the figures or the landscapes, but some of the outlines are very defined and most of the damage fades, whereupon the structure becomes easier to examine. [107]

False colour – Infrared Photography – Piety

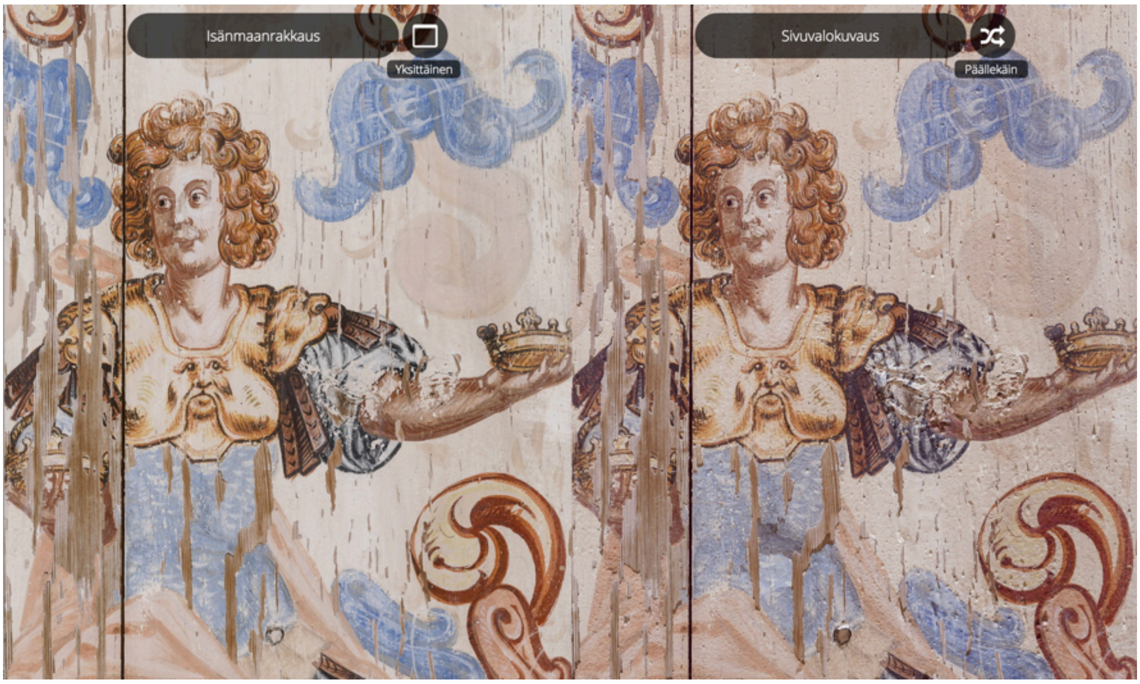
The woman, clad in a hooded cape, is holding a cube in her right hand. Flames are burning on top of the lid of the cube. She is holding long-stemmed flowers in her left hand. Her eyes are downcast. The figure resembles an image printed on the flyleaf of Queen Christina's Bible, printed in Sweden in 1646. This basic form of the figure symbolising piety also comes from Cesare Ripa. In Ripa's depiction, the woman is holding the crane standing next to her by the neck, while her other hand rests on top of the altar. An elephant and child are included next to the figure. The crane standing next to the Pietas figure in the image in Queen Christina's Bible has been omitted from the painting at Louhisaari, apparently due to lack of space. It is most likely that the flowers replacing the bird were included in order to balance the composition and represent improvised changes dreamed up by the artist. The cube held by the figure symbolises an altar with a fire burning above it. In the image included in Queen Christina's Bible, the altar has also shrunk into a handheld cube, which, to be fair, is larger than the one in the Louhisaari painting. [8]



False colour - Infrared photography combines normal photography and infrared photography and produces as the name says, pictures where some of the colours change. The method might make it possible to tell the difference between different materials, as well as their possibly different ages. The damages that go all the way to the wooden surface are seen in the Louhisaari false colour pictures as a vibrant red. [107]

Sidelight Photography – Love of One's Country

A man wearing a breastplate holds a crown in each hand. This figure partially matches Ripa's description of a young and powerful warrior figure symbolising patriotism. The flames and smoke appearing behind the soldier, quoted in Ripa's description, are omitted here. However, there would have been no space for flames in the Louhisaari ceiling painting. All of the figures in the ceiling are missing a background, making the sitting figures look as if they are floating in the air. According to Ripa's description, the warrior should be holding garlands in his hands. Garlands, which were used for paying special tribute in antiquity, are related to bravery: braided using blades of grass, a garland was given to an individual who had saved his hometown in the face on an enemy attack. An oak leaf garland was given to an individual who had saved some-one else's life in battle. The crowns included in the Louhisaari Manor's ceiling painting are a form of tribute that is easier for the viewer to understand. [8]



Sidelight highlights the surface structure. The lower the angel of the light, the more emphasised the effect becomes. The direction of the light is important, too. With this method damages, the paint strokes of the painter, the composition of the base material, etc. can be brought out. Under a steeply angled light even the smallest of details and damages in the surface are emphasised. [107]

Polarised Light Photography – Charity

A seated woman, with two infants to her left. A flame-like pattern is depicted on her forehead, and a heart is depicted under her bare right breast. The figure adheres to a type widely used since the Renaissance: the woman protects the small children, who symbolise helplessness and vulnerability. The flame burning on her forehead indicates that mercy is never idle, but rather always active. The heart flaming on her chest is a symbol of the love of Christ, or Christian Charity. It is a common symbol in Catholic culture. The children (of whom there should be three, according to Ripa), symbolise the power of mercy. For without mercy (i.e. Charity), Hope and Faith – the other two Christian theological virtues – are insignificant. [8]

The use of polarised light lessens reflections and problems stemming from diffused light by producing clearer pictures. The ceiling paintings look more colourful and it is easier to distinguish between tones. The effect, however, is modest and is similar to what can be done with image processing. [107]

Ultraviolet Photography – Winged Head

A blonde head with round cheeks whose body is replaced with an oblong, anconal pattern adorned with oval bosses, and with small wings in place of shoulders. This is the outermost image in the row of allegorical figures and, since a similar image can be found at the other end of the row, it can be considered a complementary image with no particular meaning. Similar images also appear in connection with the rest of the painting's ornamentation. The need to add these pictures to complete the row of figures probably resulted from the fact that there were fewer images on which the ceiling painting was modelled than there was space on the ceiling. Different variations of the winged head appear as a decorative motif in places as varied as decorative items and burial monuments. Similar figures can be found in 17th-century typographical ornaments, for instance. [8]

Ultraviolet photography reveals things that are otherwise invisible to the naked eye. Different kinds of pigments, retouching and damages that cannot be seen in normal light are revealed. In Louhisaari colours that have faded becomes brighter and some details that have almost entirely disappeared, suddenly turns up on UV-pictures. [107]

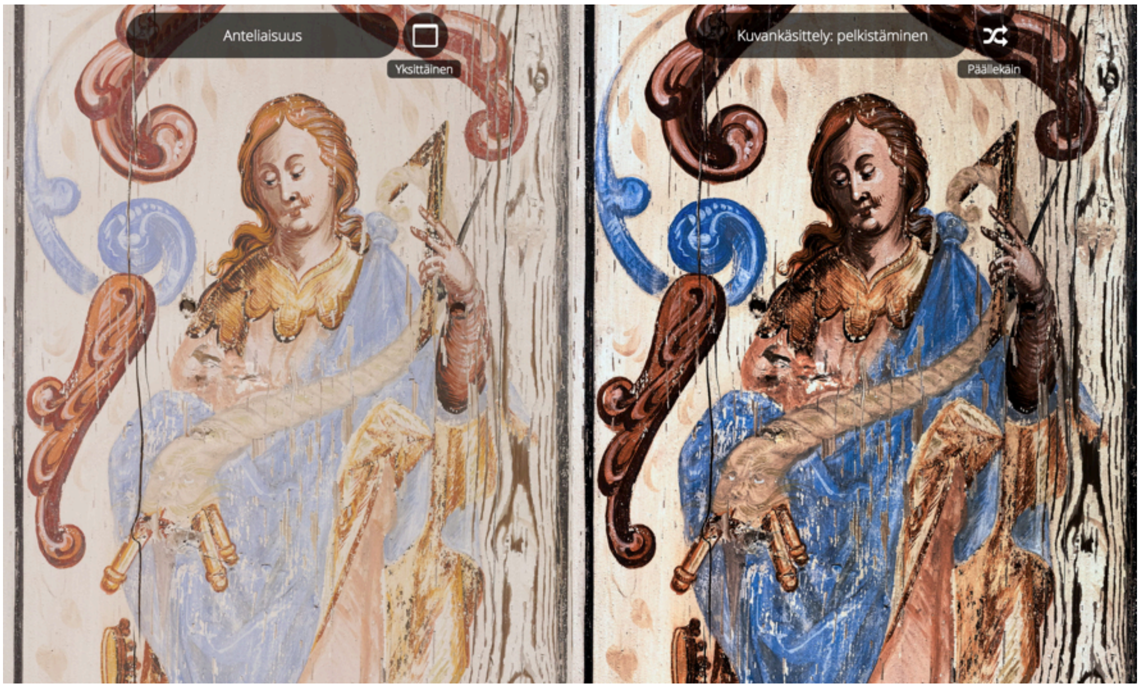
Ultraviolet Fluorescence Photography – Peace

A standing woman, with her left hand touching her temple and the left elbow resting on top of a column standing next to the figure. She is holding a branch in her right hand. This figure was based a coin minted in the mid-3rd century for Gaius Valens Hostilianus Messius Quintus, the son of the Roman Emperor Decius. It depicted the same figure: a woman, carefree, resting one of her hands on a column, symbolising determination, self-confidence and certainty, and holding a palm branch in the other. The image symbolised the time of peace enjoyed during the Emperor's reign. [8]

Ultraviolet fluorescence photography is the most user friendly of all photographic research methods. Its usefulness can be seen by naked eyes. Ultraviolet light creates a colourful lustre in some materials and pigments that can then be photographed. Some colours change and often details that cannot be seen in normal light becomes visible. The method reveals both faded areas as well as damages and makes them easier to examine. [107]

Image Processing: Reduction – Liberality

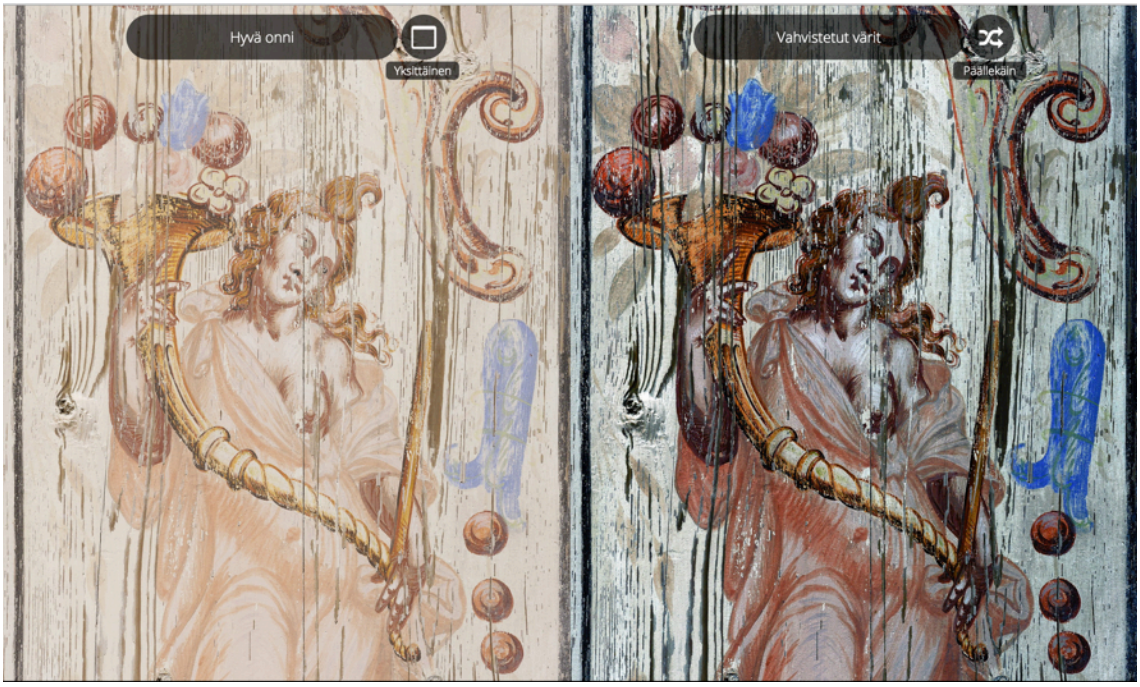
A standing woman is holding a cornucopia upside down. She also holds a set square and other paraphernalia in her left hand, which is supporting the upper end of the horn. A crown and sceptres are falling out of the mouth of the cornucopia, pointing to the right. The figure at Louhisaari is a simplified version, but still recognisable with some assistance from Cesare Ripa. According to Ripa's depiction, the woman should be clad in white, with a small eagle sitting on her forehead or shoulder. According to his interpretation, the eagle is the most generous of all the birds, since it always leaves some part of its prey for other animals. The figure should also be holding a compass divider and a cornucopia with jewellery, money and other valuable items falling out in one hand, and a cornucopia filled with fruit and flowers in the other. The content of the cornucopias symbolise both material prosperity and the bountiful gifts of nature. A generous person must pass on good things to other people because he wishes to help those people, not out of vanity or boastfulness. This is another figure that appears on the coins of several Roman emperors. [8]



Sometimes it is possible to get more information from an ordinary picture file of good quality as one would get from taking photographs taken with special methods. The surface structure and faded areas can be highlighted with simple image processing adjustments and makes it easier to make discoveries for example while mapping damages. [107]

Maximizing color and tone information – Good Fortune

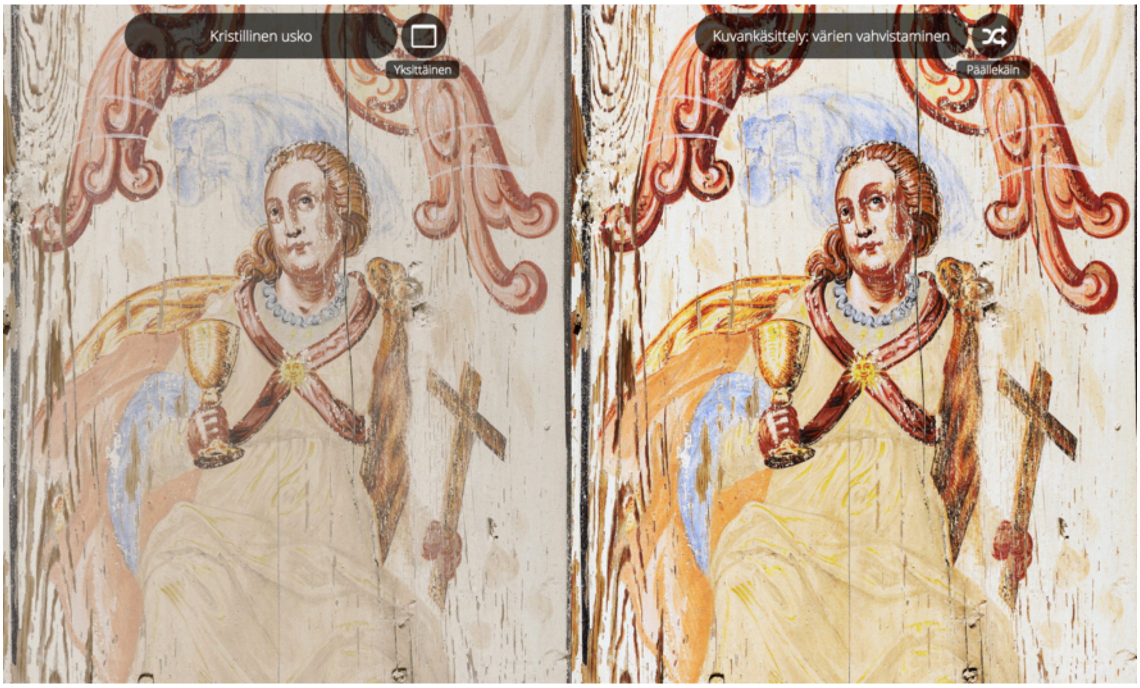
The standing woman has a jar beneath her foot. Water is pouring out of the jar. The woman is holding aloft a cornucopia whose mouth, sprouting flowers and fruit, is to the right of her head. Her left hand is holding a long-stemmed rudder that is partially hidden behind her feet. This figure matches a number of depictions included in iconological guides from the 16th and 17th centuries. For Cesare Ripa, Good Fortune, or Fortuna
was modelled on a coin from the early 2nd century of the Roman Emperor Antoninus Pius. The rudder held by the figure suggests that fortune, Fortuna, guides all human activities. The cornucopia, much like the Sampo in Kalevalan mythology, yields an endless supply not only of riches but also of commodities necessary for humans, represented here by flowers and fruit. Of course, whether a person gets to enjoy riches and prosperity depends on fortune, making this selection random. [8]



Shooting with RAW or different exposures, the dynamic range and the tones in an image can be increased. This is useful when the image goes through lots of pre processing and manipulation. Adjusting and using strong tones and settings can bring details visible, which would otherwise go unnoticed. [107]

Image Processing: Reinforcing colours – Christian Faith

A seated woman with a cube underneath her right foot; in her right hand she holds an open chalice. A sacramental wafer is positioned above the chalice. She holds a cross in her left hand. She looks up and askance. A sun is depicted at the intersection of the bands running across her chest. This female figure can be interpreted as representing Christian faith – a view supported by the paraphernalia she holds. One of Cesare Ripa's published descriptions of Fede Christiana – the figure symbolising Christian faith – mentions a square rock that represents the solid foundation of faith. On the other hand, a cube underneath a figure's foot may also refer to divine wisdom, in which case the cube would symbolise perpetuity. The artist's signature is located inside the cube. In the mid-17th century, Jochim Langh had moved from Germany to Sweden, where he worked as a decorative painter for a number of aristocrats. Langh worked for the constructor of Louhisaari Manor, Herman Fleming, in the Admiralty in Stockholm before transferring to Louhisaari in around 1660. [8]



Barely visible colours and areas can be emphasised by examining and adjusting pictures in various colour modes, or by manipulating their individual image channels. It is possible to make assumptions about the splendour of the original colours of the ceiling paintings by reinforcing the existing colours. [107]

Louhisaari Manor