TECHNISCHE UNIVERSITÄT DRESDEN

Faculty of Environmental Sciences | Institute for Cartography

Master Thesis

On the Use of Adaptive Sketch Textures in a 3D Model of a Townscape

submitted by Cinthya Morquecho Diaz
born on 01.04.1990 in CDMX, Mexico

submitted for the academic degree of
Master of Science (M.Sc.)

Submission on 10/09/2019

Supervisors Dr.rer.nat. Nikolas Prechtel
Technische Universität Dresden

Reviewer Dr.-Ing. Mathias Jahnke
Technische Universität München

Task for the preparation of a Master Thesis

Name:	**Cinthya Morquecho Díaz**
Matriculation number:	4755904
Matriculation year:	2018
Title:	**On the Use of Adaptive Sketch Textures in a 3D Model of a Townscape**

Objectives of work

Scope/Previous Results: Schematic 3D-Models of a townscape bridge the gap between abstract and "realistic" representations. Different to the latter they do not aim at maximum resemblance between real world and model. Alternatively, they present classes of geo-objects. They will (at least partly) re- place individual appearance by categorical prototypes comprising uniform surface characteristics. Geometric shape is not easily modified in contiguous schematic 3D models, where each space is assigned to a defined class. Thus, model design variation basically relies on variable texturing. As stated in publications, hand-drawn sketch textures can be more expressive, compact and flexible than most photo-graphic or algorithmic textures pretending visual realism. Their expressiveness is based on an artistic enhancement of characteristic elements associated to the underlying object properties. The transfer of semantics may thus be handled in an iconic way similar to how signatures in 2D maps often work. Since a monochrome (pencil) sketch may only depict the most prominent structures and edges of a surface (a pattern of sets, principal façade openings, outlines of a vegetation canopy, etc.), that sort of surface design can easily adapt to different illumination and seasonal situations. This property shall be used to visually convey both a daytime and a night- time model from the same geometry source.

Motivation: 2D landscape models can be acquired for major parts of the world. It could be shown that such models can - along with digital elevation and digital surface models – be upgraded to 3D land- scape models with fairly little interaction (Prechtel et al. 2013 and 2015). Although the claim for realism of photo-textured models is questionable, there is a significant lack of attempts to produce convincing non-photorealistic geo-models in an operational way. A recent paper (Prechtel 2016) documents among others sketched depictions of architectural styles of building facades. With the TUD 3D campus model, a test bed comprising four square kilometres of Dresden town, is available, and has, furthermore, been extensively documented.

Objective: An introductionary summary of the limited number of NPR geo-models based on web examples and literature review (e.g. Semmo, Jahnke, Prechtel) shall be given. Theoretical considerations have to encompass ways of stylisation of the geo-object inventory of the TUD 3D campus model. In order to balance the visual emphasis of the different type of objects, a number of sensible use cases of the 3D model might be another good starting point.

A major component in the visualisation of an urban scenery will be the depiction of the range of architectural design found within the model extent. Furthermore, different street surfaces as well as the patterns of public and private garden sites have to be symbolised including various forms of vegetation cover. It has to be taken into account that the design of simple prototype textures can flexibly be attached onto surfaces of different dimensions. Thus, repetitive patterns have to be found in most cases. Indicative studies in the graphic density, plausibility, readability and the degree of differentiation possible on the basis of a limited set of well-sketched textures shall be carried out. Obviously, aesthetics of the final assembly as achieved in different perspective views and scales are an important criterion for final quality. Right from the beginning a simple transformation from a daytime into a night-time look shall be envisaged. Something like a positive/negative switch of a drawing might be an idea to start from. The night-time look might be emphasised by radiance originating from windows (and doors) connecting to illuminated rooms.

Summarising, the proposed model design should react on various use cases (different object emphasis, different simplicity of textures), and the time component (day and night). It should be avoided to produce overly complicated rendering pipelines, since slow rendering greatly reduces the attractiveness of a model. A publication within a web environment with limited bandwidth should be one important final use option. As a visual reference, and in order to minimise ground truth efforts "Google Street View" and "Bing Maps" can assist in the design.

Deliverables: The results have to be submitted as a written document in two identical copies along with digital versions of the document on CD/DVD. All data, relevant for further scientific treatment, has to be stored on digital media, and to be added to the final submission. Finally, an A0 poster (template can be downloaded from the institute's home page) has to be produced in order to present the essential parts of the work and the results.

Supervisor: Dr. Nikolas Prechtel

Acknowledgements

Although the results we obtain during our lives depend mostly on the determination and perseverance we have, the unconditional support we receive becomes a key element in the success of our final results.

That is why I want to extend my highest gratitude to all those who were involved in this pleasant experience. First, I would especially like to thank my supervisor Dr. Nikolas Prechtel, for all his invaluable support, suggestions and comments. To my external reviewer Dr.-Ing. Mathias Jahnke of the Technical University of Munich, his suggestions were very helpful for the development of the project. To Prof. Dr.-Ing. Liqiu Meng and all the staff involved in these two years of masters at the Technical University of Munich, Vienna, Dresden and University of Twente, but especially to our coordinator M.Sc. Juliane Cron for the unconditional support she provides to all students.

I would like to thank the 10 participants who participated in my user study. Thank you for your comments and suggestions, they were a great help and added value to my final work.

To my professors of the Faculty of Engineering M.Eng. Adolfo Reyes Pizano and Ing. Benito Gómez Daza for giving me their recommendation and support to study this master's degree. To my former boss Ing. David Ferrer Torres for motivating me to keep going.

Thanks to my colleagues and great friends that I made during these two years of masters. You have been like a family to me, I have no words to express my great affection and care for you: Iñigo, Altynay, Maira, Agnieszka, Zarina, Amber, Victoria, Elif and Nawshin. I am infinitely grateful for all the moments we spent together playing boardgames, the cheerful laughs and everything I learned from you; I will take your friendship with me all my life.

Finally, I want to give a special thanks to my family, thanks for your unconditional support to encourage me to always follow my dreams. I am eternally grateful for all the sacrifices and efforts you have made for me. To my mother Carmen, my father Carlos and my siblings Juan Carlos and Vanessa for such exceptional support, thank you for giving me the greatest examples of success. You are my greatest motivation.

Statement of Authorship

Herewith I declare that I am the sole author of the thesis named

„On the Use of Adaptive Sketch Textures in a 3D Model of a Townscape“

which has been submitted to the study commission of geosciences today.

I have fully referenced the ideas and work of others, whether published or unpublished. Literal or analogous citations are clearly marked as such.

Dresden, 10/09/2019 Signature

Contents

Figures

Tables

Abbreviations

API	Application Program Interface
APP	Application
C4D	Cinema 4D
CityGML	City Geography Markup Language
CNN	Convolutional Neural Network
COLLADA	COLLAborative Design Activity
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
CZML	CesiumLanguage
G-Buffer	Geometry Buffer
GDB	Geo-DataBase
GeoJSON	Geographic JavaScript Object Notation
GIS	Geographic Information System
GLSL	OpenGL Shading Language
glTF	Graphics Library Transmission Format
HTML	Hypertext Markup Language
JPEG	Joint Photographic Experts Group
JSON	JavaScript Object Notation
KML	KeyholeMarkup Language
KML	KeyholeMarkup Language
LOA	Level of Abstraction
LOD	Level of Detail
NPR	Non-Photorealistic Rendering
NST	Neural Style Transfer
OBJ model	Object Format Model
OpenGL	OpenGraphicsLibrary
SPSS	Statistical Package for the Social Sciences
TopoJSON	Topology JavaScript Object Notation
TUD	Technische Universität Dresden
UE	Usability Engineering
UI	User Interface
WebGL	WebGraphics Library
XML	Extensible Markup Language
ZIH	Zentrums für Informationsdienste und Hochleistungsrechnen

1. Abstract

The aim of this thesis is to present a framework of a web-based 3D city model, which will provide a scope in the implementation of sketch style models applied directly to the facades building in the campus of the Technical University of Dresden, likewise the motivation behind the implementation of this techniques and the results given with the application of the Virtual Globe, CesiumJS.

In this project, it is used images (camera images) for the rendering of the facades of the buildings. Therefore, it is part of the review in the background, the application of image-based techniques in 3D objects, which will modify the images by using software techniques and the implementation of Neural Style Transfer. Additionally, an overview of the non-photorealistic rendering (NPR) techniques is provided, mainly focused on the its application to image generation and the 3D models.

As part of the project, it is presented the proposed methodology to achieve the NPR texture aspects, its bind to the geometries, the software, the conversion tools to transform from a static environment into a web environment, which means the acquisition of the glTF format, format that is needed in the Cesium JavaScript library. Furthermore, the steps of the web interface design and the discussion in the tools to be implemented in the web application published in http://cintymd.github.io.

On one hand the results in the implementation of the 3D city model based on Cesium, likewise the results in the User Test. These outcomes will be analysed and discussed, as well as the further development and scope, where these tools can be implemented in the future.

Keywords: 3D; sketch; Non-photorealistic rendering; NPR; adaptive 3D model: Cesium; cartography; WebGL; navigation; 3D cartographic modelling; virtual globes

2. Introduction

2.1. Motivation

Landscape visualisations have gained a great popularity and development through the past years, these kind of conceptions widely employ photorealistic methods to portray objects as authentic as in a real-world perception (Gooch & Gooch, 2001). Thanks to it, there has been a considerable progress in 3D visualisations, so today there are multiple implementations available, such as the vast coverage that Google has in its maps 3D view or the fascinating scenes represented in several video games, as well as animated movies. A breakthrough in the 3D concept, that reached cartography, leads to considerable advance from 2D information collection from major parts of the world, has made the upgrade into 3D or 2 ½ D more manageable; besides, the advance in the 3D model creation caused these transformations to be achieved with fairly little interaction (Prechtel, 2015). All this mainly pushed the development of tools and software to automatize the scenes production.

Stronger abstraction from reality will be at least another option and has not been extensively used in the cartography sector. According to Buchroithner & Knust (2013) some new methods to apply in cartography already exist, but they are partially ignored by the cartographic community, what did hamper further improvements. This holds for the creation of convincing non-photorealistic (NPR) geo-models. Part of the encouragement to create NPR models, is the challenge to create the reality in the mind of the observer without applying realistic modelling methods by combining computer graphics and artistic techniques (Gooch & Gooch, 2001). These representations have the potential to make the graphics clear and comprehensible (Jahnke, 2013). Furthermore, as it is stated by Prechtel (2016), an essential motivation for non-photorealistic 3D geo-visualisations lies in the comparatively low sensitivity of stylised textures to heavy downsizing, which applied in a variable façade representation (where the texture can be selected) allows the comparison between architectural design and current state.

In the field of NPR visualisation-techniques, the use of the use of sketch-based models is generally understood to mean the communication of ideas transmitted with the use of a sparse, simple line drawing (Liu et al., 2013). Leading to some advantages in the use of this method as it is stated by Tano, et al. (2012) media is expected to expand the creativity and intelligence by using three principles “memory capacity”, “cognitive mode” and “life-size and operability”. Sketch models create an easy transition between the experimental cognition (intuition) and reflective cognition (true reasoning). Moreover, Semmo (2016) summarises some of the limitations of photorealistic rendering, such as: the cognitive load carried out by the user, visualisation of uncertainty by using cartographic techniques, the visualisation on small displays and the use of more computational memory resources. Considering that, conceptual designs are traditionally sketched by users and they project the natural way to express the user’s intent in a visual form (Liu et al., 2013).

Whilst the use of photorealistic techniques in many navigation systems and 3D web visualisations is mainly applied to facilitate the mental mapping between the digital visualisation and the reality; the application of sketch-based techniques, as a computer-based visualisation systems, is intended to help the user to carry out a task more effectively (Semmo, 2016). Furthermore, the downsizing of the information in the image would release the overload of the rendering time in a web-based environment. On one hand, thanks to the characteristics to suggest additional semantic information provided by this technique, it is expected for the user to create 3D landmarks in reduced spaces, to search for specific information, which could be applicable in emergency assistance, urban planning, 3D cadastre, navigation, tourist information, planning evacuation, and disaster management (Biljecki et al., 2015), as well as, orientation in a 3D environment provided by virtual games or the availability of semantically structured 3D models in the university environment (Prechtel, 2016), in order to ease the navigation and search effort for specific structural information. On the other hand, the downsizing of information in the texture facades of the buildings, might lead to an efficient render of the geospatial information.

2.2. Objective

The principal aim is the generation of a 3D adaptive sketch geo-model of a townscape that fulfil the requirements and it is based on an efficient pipeline, in order to accomplish this, the objective is to:

- Create the suitable 3D geometry, according to the Level of Detail (LoD) that will be perceived in the townscape model, in order not to overload the rendering pipeline.
- Define the different visual appearances of the objects in the scene, required to create the adaptive interface.
- Generate the sketch models of the building façades and other elements in the scene.
- Create a prototype interface with adaptive functions programmed in CESIUM.
- Carry out a user test in the usability on the functionalities and interaction of the outcome.

2.3. Structure

The present work is divided into 6 chapters. The first section gives a brief overview of the content in this thesis, which also includes the keywords associated to this work. The second section combines the motivation to implement an adaptive sketch textures 3d model UI of a townscape and the objectives to achieve at the end of this project.

This paper third chapter begins by examining the literature on which is based each of the fundamental aspects to developed and implemented of the final product. The methodology is outlined in the fourth chapter, including background in the workflow implemented in projects with a similar scope, giving an overview in the procedure to carry out the data processing and evaluation of the final product.

In the fifth section is presented the practical operation to accomplish the final web-based interface, which covers the objectives described in the second chapter. The next section analyses the results obtained in the evaluation of the application and impressions of the users. Finally, some conclusions are drawn in the final section, including the future outlook in the development if this project.

3. Literature review

3.1. Visual design

3.1.1. Image-based textures

In terms of image abstractionThere is a considerable amount of literature on the use of NPR techniques implemented for 2D and 3D visualisation. This last one, mainly focused on the transformation of the visual appearances of the objects contained in the scene, in order to reduce, generalise the amount of data perceived, add context, etc, giving as a result the fixation of the user's attention to specific elements or areas on the objects.

One preliminary information is the use of the concept of "Semiology Graphics" used by Ware (2004), which helps to understand the aforementioned process of perception. He notes this term, to the attempt the classification of all graphic marks in terms of how to express data. In his analysis, patterns are the very essence of attention, and where the brain direct the eyes to move determines what will become the focus of our attention in the next instant. Therefore, this will assist in the determination, whether the abstraction by the implementation of NPR imagery can focus the viewer's on important information (Klein et al., 2000), or might disorient from the meaningful information

Within the framework of the mentioned criteria, the appearance of most objects can be defined by textures (Semmo & Döllner, 2015). Besides, according to this last reference, the texture mapping is a key technology in today's graphics hardware for the visual design of 3D scenes, in which its important highlight the term "texture". Several authors attempted to define this term; nonetheless, in the work carried out by Tuceryan & Jain (1998), this term is referred as a function of the spatial variation in pixel intensities (grey values).

By using this conceptualization the notion of object detection, as well as its characteristics in digital images was preliminary used for aerial photographic interpretation and mapping, according to Herold, Martin; Liu, XiaoHang; Clarke (2003), where he states a hierarchy and degree of complexity of elements contained in an image (Figure 3-1) and its degree of complexity:

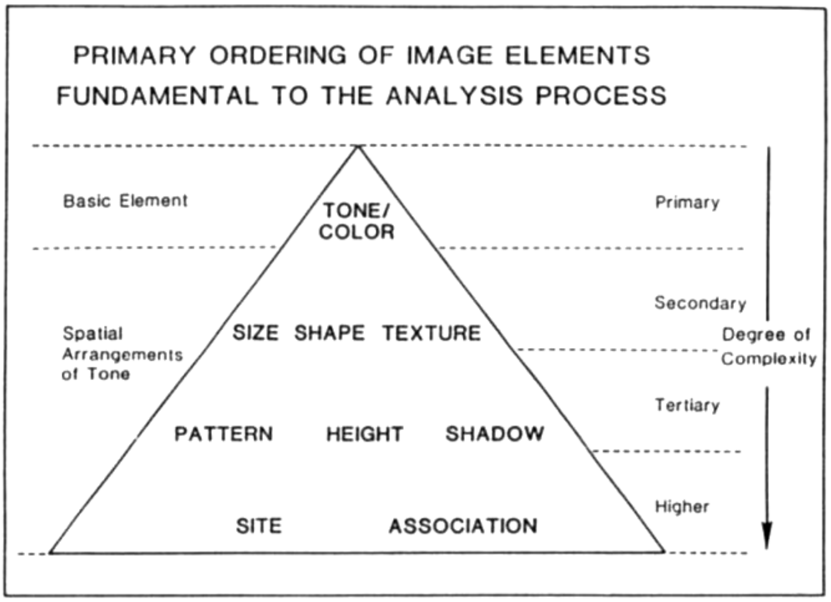


Figure 3-1. The primary ordering of image analysis elements in visual aerial photo interpretation (Herold, Martin; Liu, XiaoHang; Clarke, 2003)

Using this prior knowledge on relate to the NPR model's creation from the image's texture, seems to be a task worth using in 3D environments. The implementation of this technique, according to this project, suggest the employment of camera images, drawings or mathematical-procedural method for texture generation (Figure 3-2), which opens up an unlimited design possibilities, if it is assumed that the texture surface is the most variable design element in a NPR model (Prechtel, 2016). The test of their use by applying sketch techniques, in order to identify the meaningful elements in the structures (DeCarlo & Santella, 2002), which captures all the variations will certainly help to provide semantic information. One major improvement, in the automatization of the non-photorealistic image generation, are numerous filtering techniques, that have been proposed to derive local image structures for feature-aware processing (Semmo, 2016), such as the application of high pass filter techniques.

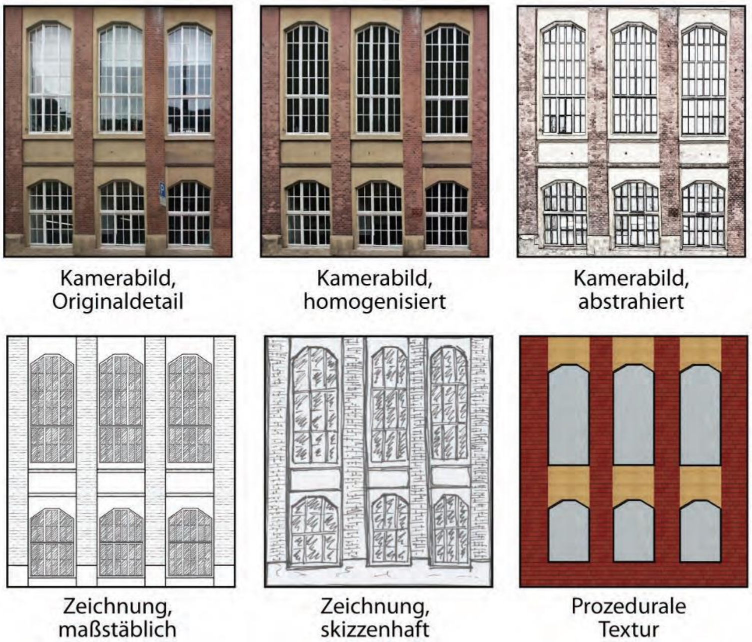


Figure 3-2. Scheme for texture types in 3D models (Prechtel, 2016).

Additional methodology for the automatization of the NPR texture-generation is the implementation of Neural Style Transfer, which is the process of convolutional neural networks (CNNs) (Gatys, Ecker, & Bethge, 2015) to identify the grammar snippet and snippet parameter values in the original appearances (Nishida, Garcia-Dorado, Aliaga, Benes, & Bousseau, 2016). As it is described in this scientific paper, these parameters are the ones that best explain the elements happening in the image as an accurate approach in the sketch model application. They intend to derive local image structures, that can be semantically related by the user and thanks to the characteristics of CNNs, it is possible the creation of a simplified, but effective object recognition.

Certainly, the main downside in the application of 3D townscape models is that depending on the orientation and the extension of the third dimension, building contours may run into each other, making the user unable to distinguish facade features from each other (Jahnke, Fan, & Meng, 2009). Nevertheless, a discussion of the cartographic techniques, which could overcome this major drawback, falls outside of the scope of this paper.

3.1.2. Sketch techniques

As already expressed, the essential based of this work is the implementation of a sketch-based 3D model, within the frame of the non-photorealistic technique application. Semmo (2016) affirms that "non-photorealistic rendering techniques comprehend depiction styles used in computer graphics that relate to regular artistic styles with the intention to communicate complex image contents by emphasizing important or prioritised information and omitting extraneous information". based on the affirmation that the use of sketch techniques is expected to produce a major contribution in the communication of specific data enclosed in the scene. Due to its nature, the application of this technique might be regard as a cartographic tool to apply generalisation methods in the spatial objects.

This sketch effect directly applied in a camera image approach will be determined by using some edge-based techniques and applying defined masks to the images. Therefore, a set of rules will be established. This methodology has its foundations in the application of convolution mask, specifically, the use of high-pass filters (Figure 3-3).

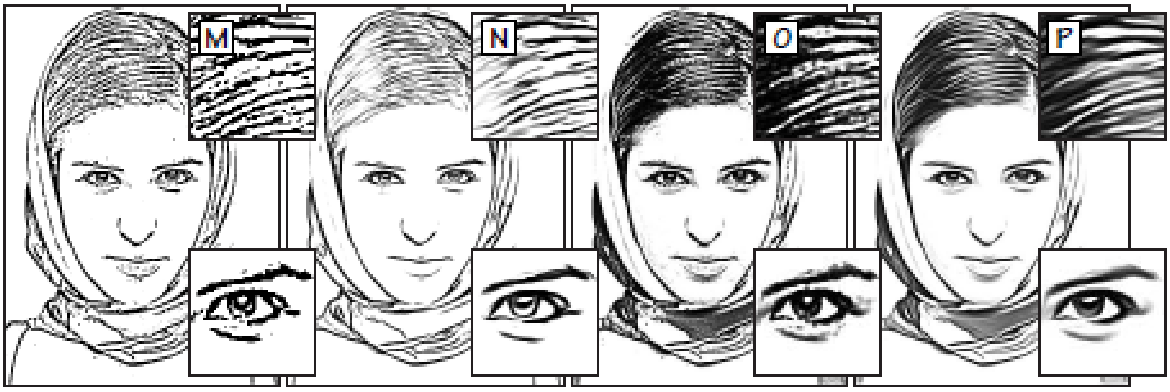


Figure 3-3. Obtained from Winnemöller et al. (2012). Input image © Maryse Casol.(Semmo, 2016)

As part of NPR techniques, sketch models in the building facades will carry the advantages in the visual perception. Within these benefits are the creation of fixation in the user's perception to follow meaningful location, since people can only examine only small visual areas at one time (DeCarlo & Santella, 2002).

What we know, in terms of NPR application and implementation of these techniques is largely based in the work developed by the Hasso Plattner Institute in the University of Potsdam. Nevertheless, in 2012 Tano et al., performed a user study in the implementation of 3D sketch models. This provided a great overview in the memory capacity and cognitive mode, life-size and operability, concepts that have been evaluated in the same work.

3.1.3. Neural style transfer

The search for automatic conceptions of image generation is not a new task. Although many improvements have been made in this field, the use of Style Transfer have become an area of study with great potential. The term Style Transfer is generally understood to mean the texture synthesis, in order to extract and transfer the texture from the source image to the target image by using the capture of image structures and preserve the semantic content in the target image. (Gatys et al., 2015; Gatys, Ecker, & Bethge, 2016).

In his study, Gatys et al., (2015) explains that a neural system applied to the style transfer is trained to perform one of the core computational task of biological vision, automatically learns image representations that allow the separation of the image content from the style. This complex task is achieved by using Deep Neural Network, which are trained to recognise objects using a certain number of filters along the process (Figure 3-4), the size of the filtered image is reduced by some downsampling mechanism, leading to a decrease in the total number of units per layer of the network.

This technique has been applied to generate NPR stylisation algorithms, mainly focused in the design of artistic style (Gatys et al., 2015). Even though its main approach focuses on the artistic style, this mechanism can be implemented for the generation of sketch-based textures by using the proper input image, since there is a significant amount of artwork based on sketches, such as cartoon scenes.

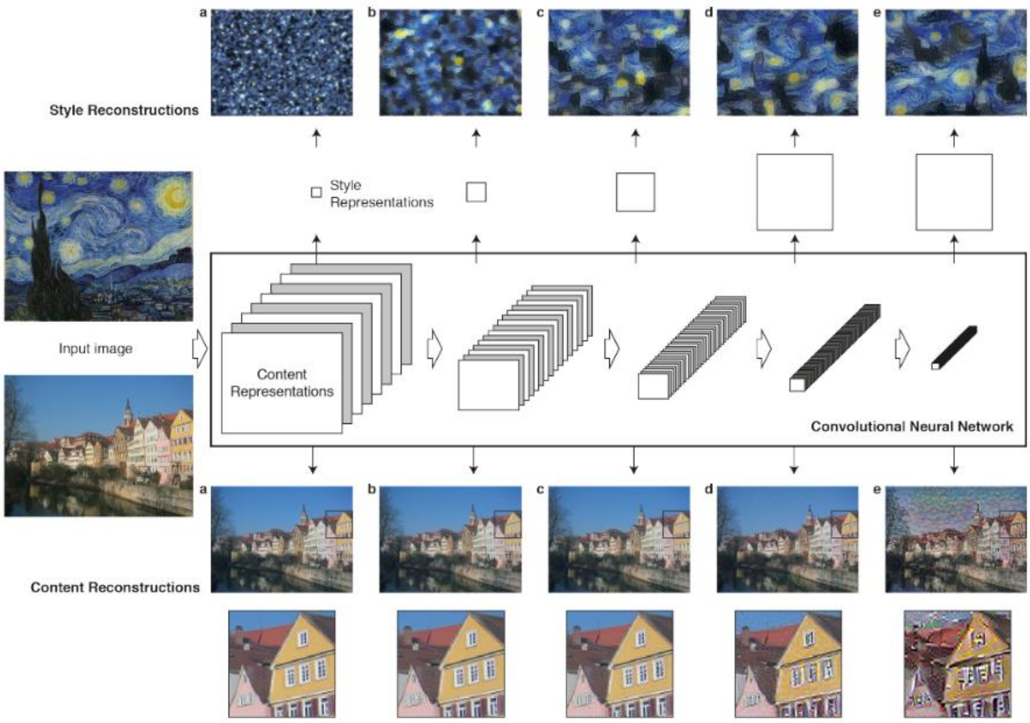


Figure 3-4. Image representations in a Convolutional Neural Network (Gatys et al., 2016).

3.2. Cesium JS

3.2.1. Introduction

The fundamental engine visualisation in this project is based on the virtual globe Cesiumjs. In its website Cesium is defined as an open-source JavaScript library for world-class 3D globes and maps, it is based on HTML5 technologies (Abdalla, 2017), which uses WebGL technologies to provide hardware acceleration and plugin independence to provide cross-platform and cross browser functionality. Additionally, Cesium supports the presentation of 3D contents where the user can dynamically change between 3D globe visualisation and 2D map projection (Chaturvedi, Yao, & Kolbe, 2015).

Within the benefits, it is stated that Cesium provides a higher level of abstraction and its architecture includes four main layers (Figure 3-5).

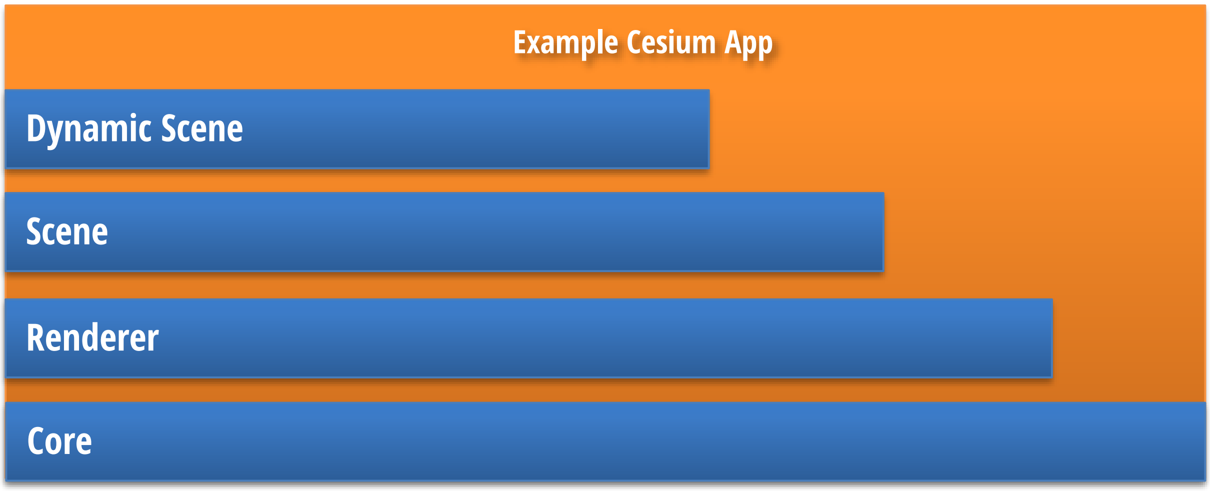


Figure 3-5. Cesium architecture redrawn from Chaturvedi et al., 2015.

The different layers of the architecture are responsible to add specific functionality and to raise the level of abstraction. The details are (Chaturvedi et al., 2015):

- Core – This is the lowest layer in Cesium and includes mainly low-level functions. These functions majorly include computations and calculations such as mathematical conversions, transformations and projections.
- Renderer – This layer is a thin abstraction over WebGL. It includes already available GLSL functions to provide shader programs, textures and buffers.
- Scene – This layer is mainly responsible to provide overall functionality of the globe. It includes high-level globe and map constructs such as 3D globe or map, handling layer imageries from multiple sources, creation of geometries and materials, camera control and animation.
- Dynamic Scene – This is the top-most layer of Cesium, which provides dynamic visualization of the data with the help of its in-built language CZML. It allows to store the data in dynamic objects, loads and renders the dynamic objects all together instead of rendering every frame.

Cesium provides a diverse number of examples and although they are categorised in different fields, each approach claims its own innovation. Furthermore, these implementations make use of the widen amount of methods offered by the library. It is worth mentioning that a recurring application is the 3D city models. The implementation of 3D city models based on Cesium have gained interest by a wider audience of both of the consumer and the scientific community (Murshed, Al-Hyari, Wendel, & Ansart, 2018). Among the examples which apply this tool, is CityGML, as well as the large number of examples in the Cesium website (Figure 3-6).

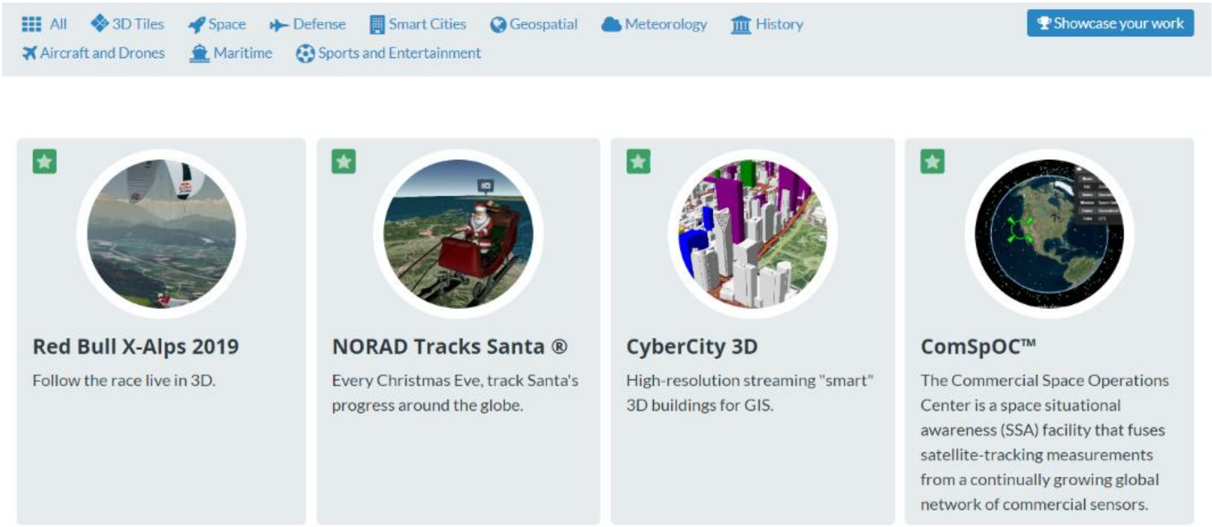


Figure 3-6. CesiumJS Demos (https://cesiumjs.org/demos/).

3.2.2. glTF format

Cesium accepts different input 3D sources; nevertheless, the persistent employment of its native format (glTF) is a well-known data source, yet it is not the only format that Cesium admits, within the different sources are KML and COLLADA. glTF, which can be obtained by using the transformation tool Collada2gltf & obj2glf developed by Cesium. This last format is the one to be used throughout this work.

In accordance with the Khronos Group website (Khronos Group, 2015), the glTF™ (GL Transmission Format) is a royalty-free specification for the efficient transmission and loading of 3D scenes and models by applications. glTF defines an extensible, common publishing format (Figure 3-7) for 3D content tools and services that streamlines authoring workflows.

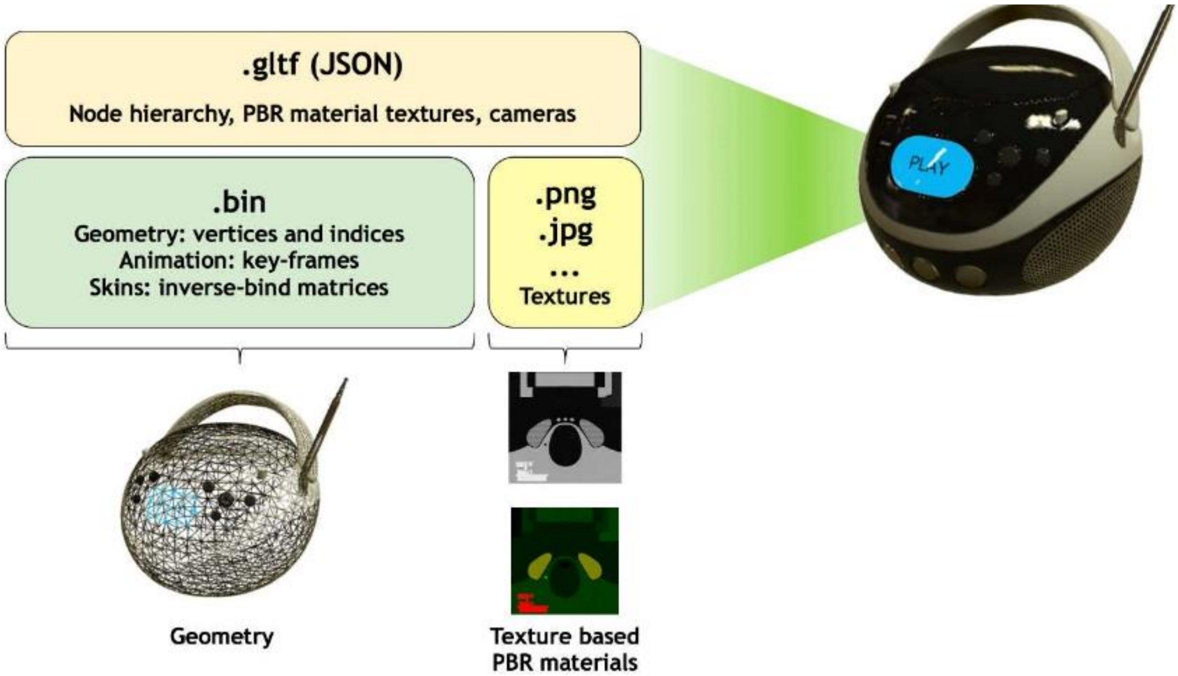


Figure 3-7. Scene Description Structure (https://www.khronos.org/gltf/).

This format has been widely accepted and used as the 3D standard format in multiple software. Moreover, its employment by several companies seeks different purposes of application and users in web-based environments as one of the formats recognised in JavaScript (Figure 3-8). It encourages an extended possibility in the development of this project, giving the opportunity to expand the insight into new interfaces or adaptation to different screen devices. Nevertheless, this topic will be discussed in a further chapter.

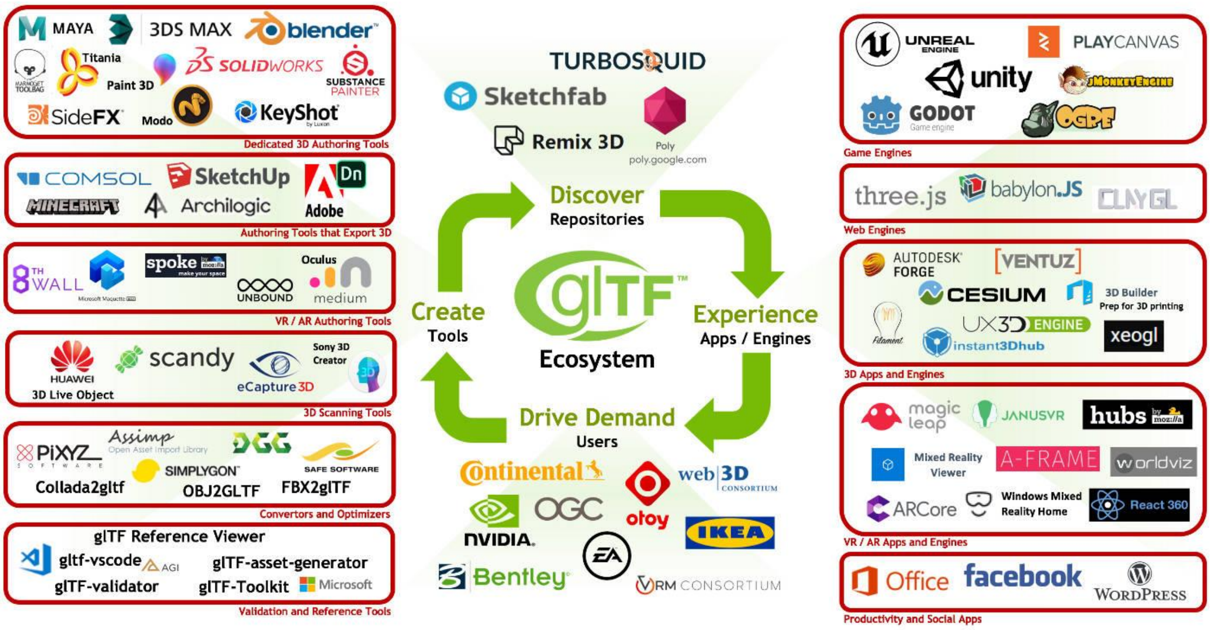


Figure 3-8. COLLADA and glTF 3D Asset Formats (https://www.khronos.org/gltf/).

3.2.3. Limitations of Cesium JS

Cesium has demonstrated collective advantages in its implementation, within the preceding literature a Despite Cesium is a valid platform to start working 3D city models, a discussion about its limitations lies in the study carried out by El Haje, Jessel, Gaildrat, & Sanza in 2016.

A web-based implementation for a 3D city render derived from the use of Cesium carries benefits, such as the support for light maps on the quantized mesh, accordingly to a time of day, among others. However, Cesium inherits JavaScript limitations and performance issues, which greatly vary from browser to browser, due to the different engines implementation. To this reason, current WebGL browsers fail to render huge 3D city models (El Haje, Jessel, Gaildrat, & Sanza, 2016).

To this view is added the fact that the imagery provided by Cesium is inferior compared to the one provided by Google and the evidence that determines a partial similarity in both globes functionalities (Moradi, 2017). Finally, Cesium does not support the direct load of data from GDBs supported by ArcGIS, conversely it has to be converted into Cesium 3D Tiles (Murshed et al., 2018).

3.3. Virtual 3D city model

Data visualisation is indeed an essential task in the cartographic field to stimulate the better understanding of the surrounding environment. Nevertheless, in the past, the existing technology for the Web-based implemented solutions had several limitations in terms of computer's hardware and internet bandwidth (Lin, Chang, Tsai, & Shih, 2017). It is substantial to acknowledge that urban land management undertakes all actions, strategies and planning of the city or community, in order to maintain and develop the balance between the environmental, economic and social needs (Ross, 2010), which this project narrows into the extent of the TUD main campus. This is how, among other solutions, the implementation of 3D city virtual models gains a considerable popularity for its development, as key in the urban modelling, allowing the user the visualization of this complex data in different LoDs, at multiple aggregation scales (Murshed et al., 2018).

The understanding behind a virtual 3D city model lies in the report of the implementation of the Berlin 3D city model (2006), who defined these models as a representation of urban spatial data and geo-referenced data, under a common metaphor, the virtual city. An additional asset to this outcomes is the inclusion of digital terrain models (DTMs), building models, street-space models and green space models, as fundamental components (Döllner, Kolbe, Liecke, Sgouros, & Teichmann, 2006). Hence, the perception process implicates the abstraction of the real elements in the city, which will comprise some of the city properties and attributes (Moradi, 2017).

Preliminary work on the web-based exploration and implementation with large and structured semantic 3D city models indicates that deep cues are needed for the localisation of map information in a 3D space; likewise, the complexity of the derivation of a certain information in the model strongly depends on the context and the characteristics of the visual process (Chaturvedi et al., 2015). As part of this study, the stimuli classes of a display add further complex parameters describing the respective properties of a display, defining the following three groups:

- Physical properties (e.g. element size, position, colour, contrast).
- Gestalt properties (e.g. overlapping elements).
- Cognitive properties (e.g. visual attraction, figure ground relation).

Whilst plenty web-based 3D virtual city implementations focus the rendering of the geometrical component, this project intends to include the complementary texture to the façades and preserving only the prioritised information contained in the images by adding some generalisation techniques in the façade building's textures. This is supported by the study carried out by Janke et al., (2009), which intends to avoid the overlapping. Therefore, the resulted "flat" façade bear a striking resemblance with a map (Jahnke et al., 2009), easing the recognition of the structures thanks to the classification of the architectural information.

Finally, the implementation of these models in a web environment in real-time dynamic visualisation can be accomplished in association with the direct employment of OpenGL, which achieves a highspeed render. It is worth mentioning that most of the implemented solutions are based in the CityGML information, as this data can be managed in popular data based such as PostGIS and Oracle, creating a tile-based approach for visualising the city (El Haje et al., 2016).

4. Methodology

4.1. 3D Townscape model design

The management and integration of the geoinformation encompasses the combination of different tools in the workflow, either for the on-demand integration thematic data with the virtual 3D campus model and the dissemination of this 3D city model through digital media such as internet (Döllner et al., 2006). Although the use of a 3D database allows applications and systems to access contents in the virtual model without the use of other subsystems, in a primary stage in the development of these tool, the access and render of the data is done by declaring different parameters applying the JSON structure, combined with the geo-spatial parameter.

The exploration of the Cesium library, as one of the most suitable to create virtual globes, carried both dynamic 3D visualisation of geospatial data and native discrete temporal data support, this combined with the existing layers, such as widgets, data sources, scene creation, render and core (Figure 4-1) (Murshed et al., 2018). Based on this knowledge, it was necessary to define the following phases in the workflow:

1. **Data preparation**: Cesium supports a diverse number of formats to visualise 3D data each of them owns different capabilities. Some examples are CityGML, JSON, GeoJSON, CZML and 3D tiles (Murshed et al., 2018), this is how the choice for this project depends mainly in the functionalities. Thus, as a requirement, the final format is expected to handle texture images in the 3D geometries, which, at the same time, could be easily treated in Cinema 4D. Therefore, taking in consideration this description, the format employed is glTF.
2. **Implementation**: This step thoroughly described the process from its initial state of preparation iuntil the final UI, adding programmed tools and the description of the aspect design modification of the façade's texture.
3. **Visualization**: As part of the transmission of information a proposed interface will be designed for the application and the different interactions in the scene.
4. **Evaluation**: the design, user impressions, comments over the 3D models and use of the web interactions; moreover, the statistical results in terms of utility of the UI.

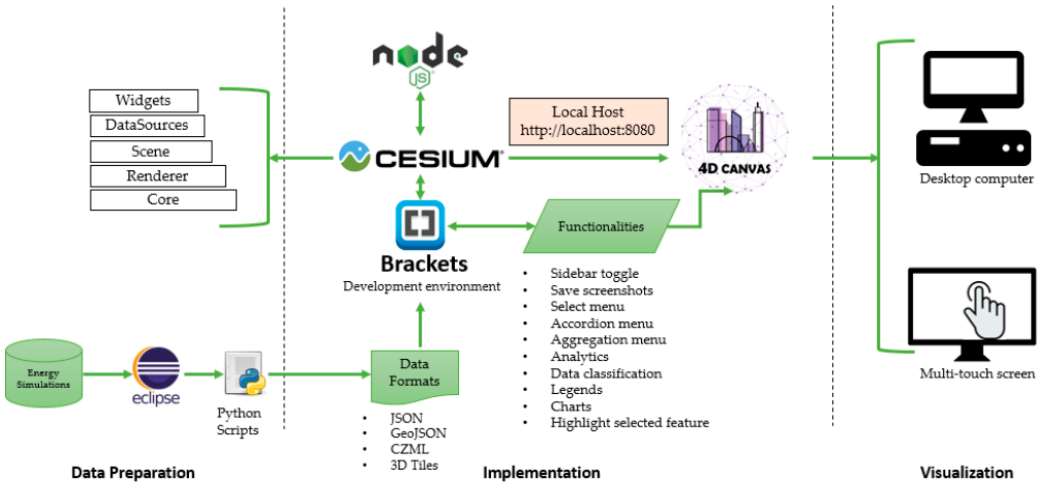


Figure 4-1. Used the proposed system architecture for the dynamic 4D interface in Murshed et al., 2018.

This proposed model holds in its workflow the use of various scripting languages, libraries, API, software and data formats explained in Table 4.1.

Category	Sub-Category	Purpose of Use
Script Language	HTML5	The markup code for displaying pages on the web browser.
	CSS	A method used to add style to the HTML page.
	JavaScript	Program the interaction with the scene.
Library / API	Cesium	Main library on which the 3D is built.
	Deep Art Effects API	API with different designs to apply the neural style transfer in the façade's images.
Software	Cinema 4D	Modelling application to edit and design the 3D campus buildings and texture. Export to COLLADA.
	KronosGroup glTF converter	A command-line tool to convert COLLADA files to glTF.
	Photoshop	Design and filtering of the façades texture aspects.
	SPSS	Statistical tool to analyse the results in the user test.
Data format	c4d	Cinema 4D format, which holds the buildings and its textures.
	COLLADA	3D format for interactive 3D applications.
	glTF	Cesium 3D format which supports materials and source code
	JPEG	Digital images of the façades.
	GeoJSON	Attributes and coordinates of origin of the buildings.

Table 4.1. Technologies, software and data formats deployed in developing the application adapted from Murshed et al., 2018.

This experimental multi-level system set up would allow the interaction of user groups with different GIS backgrounds. The partial initial setup in the use of Cesium as a virtual globe, bears a close resemblance to previous implementations in city models and 3D navigation landscape proposed by D. Moradi (2017) and M. Abdalla (2015). These previous studies have based their implementations on the use of Cesium as the main Virtual Globe to visualise 3D models. Nevertheless, it is important to emphasise the essential differences in these approaches, such as the 3D information source or the central focus of the non-photorealistic implementation, based in skectch techniques.

4.2. Data processing

The data process and modelling in this thesis, has two main areas of development: the 3D object modelling and the façade's aspect design and stylisation. Most strategies in the visualisation of 3D models in web environments, base their solution on the application of 3D tile-sets already available (e.g. CityGML). This information would not initially contain the façade's information that in terms of rendering speed, it would increase the time to visualise the information, carrying with it an overload in the UI, depending on the amount of 3D models in the scene. Hence, a bridge is digitally built through the different tools to visualise such information in a web environment by combining the provided instruments in Cinema $D for editing and Cesium for handling and visualisation.

Our area of study is focused on the main campus of the Technical University Dresden (Figure 4-2), located in the South part of the city of Dresden. Inside this area a small sample of buildings was taken to build the workflow process, to define each aspect and initial setup of the models. Once defined this information will be used to extend the visualising area.

Figure 4-2. Area of study (https://navigator.tu-dresden.de/)

According to the study carried by Prechtel (2016) s large-scale 3D spatial data is available since 2014, with the TU Dresden campus model. The buildings in this representation are modelled in LoD2/LoD3 with fairly precise façade and roof details and with photorealistic textures associated (Figure 4-3). The data source for the development of this project is based on these preliminary outcomes; therefore, the original aspect texture of these models will be transformed by applying software techniques.

Figure 4-3. The campus model from (fictitious) high-rise perspective. University buildings appear as landmarks with photo texture. (Prechtel, 2016).

For each different aspect design different algorithms application were defined following the same methodology process (Figure 4-4). The different approaches define a total of 5 texture aspects. Each technique pursues the reduction of the image details and compression, which graphical density and degree of abstraction can be adapted, especially in monochrome versions (Prechtel, 2016), in order to compare and complement these visualisation techniques, when solving task of recognition. The different models with the designs will be accessed using JavaScript methods, which will allow the user to adapt the façades to its needs or requirements, something that might lead to a more efficient location of objects in the façades.

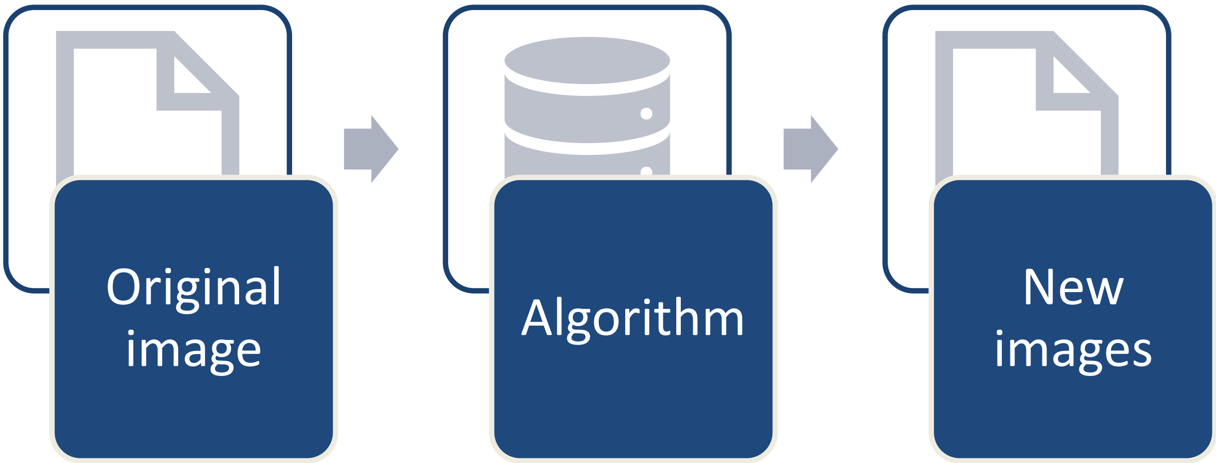


Figure 4-4. Off-line image pre-processing for the non-photorealistic styles.

In his article on design options in digital 3d landscape models, Prechtel (2016) claims that an explicit edge enhanced design is never photorealistic, because edges in reality become recognisable only indirectly through material changes or other exposures to light source. Hence, image genesis in 3D modelling software simulates the interplay of surface properties and illumination, dependant on the surface parameters, such as the surface patterns, roughness, diffuse and specular reflexion of light, lightning and transparency. This is how, each of the style techniques will be affected by these parameters and will, therefore, to depict a higher or lower amount of information in comparison to each other.

Photorealistic is the initial system setup in the UI where neither abstraction, nor stylisation in the image is applied. All the characteristics in the original image are preserved. As the first information source for the user, this style aims to ease the earliest interaction and recognition for the user, to build an initial map recognition of the area.

Artistic is the first NPR technique applied to the building's façades, which although is a NPR technique, would not share the same benefits of the sketch-based techniques. The user will have the opportunity to compare these styles and be able to identify basic objects in the façades (e.g. windows or entrances), yet, the user will not be able to establish a detailed perception of the elements contained in the façades, due to the LoA in the models.

Pencil is the first sketch technique that is based on a set of line drawings in grey scale that mimic the pencil strokes in a paper. The hypothesis behind the use of this style is that, although, pencil strokes might not depict the objects with sharpen lines, the different stroke thickness would provide a categorisation and level of importance of the structures, as well as a recognition of materials in the façades at the same time as it grants an aesthetical visualisation of the model.

Edge enhancement or *Contour* will attempt to fixate the user's attention on particular objects (windows and doors mainly) and not so much on the materials or additional elements on the façades such as vegetation. This might provide a semiology graph connection in the elements to conceive what is contained inside of the building without having a look of each floor plan of the building. However, it does not intend to replace the interior information, which, on the contrary, would complement the model.

The last aspect design is the sketch-based image achieved through the application of NST and *CNN*. Although the multilevel application of software techniques might achieve a functional outcome, the use of NST techniques would help to reduce the time for sketch-based image creation as the image generation becomes faster when applying this technique, once the network is defined for a specific image. However, the perception of the scene is fairly artistic, which might lead to an entertained navigation for the user.

Finally, as part of the landscape view, some other elements are integrated, such as trees and streetlights. These elements are expected to provide an additional context to the area, as they are objects that the user would encounter in the real life. It is stated by Semmo (2016) that only few works deal with the cartography-oriented visualization of green spaces in 3D virtual environments. Furthermore, in general, vegetation objects can be distinguished with and without individual characteristics

4.3. Web environment design

One fundamental part in the 3D virtual implementation is the design of the user's interface, and the extent of interaction given to the user. For that reason, some basic architecture will be defined, in order to achieve an initial setup in the UI, with basic interactions. The aimed interplay will be added as an external menu apart from the wide extent of widgets offered by Cesium of which only the most significant will be preserved.

According to the Cesium website[1] by default, the initial scene in handles default camera controls and the Viewer comes with some widgets:

- *Left click and drag* - Pans the camera over the surface of the globe.
- *Right click and drag* - Zooms the camera in and out.
- *Middle wheel scrolling* - Also zooms the camera in and out.
- *Middle click and drag* - Rotates the camera around the point on the surface of the globe.

[1] www.cesium.org

Apart from the inital camera control, Cesium provides some widgets in the basic setup of the Viewer (Figure 4-5) and its description is provided below.

1. *Geocoder*: A location search tool that flies the camera to queried location. Uses Bing Maps data by default.
2. *HomeButton*: Flies the viewer back to a default view.
3. *SceneModePicker*: Switches between 3D, 2D and Columbus View (CV) modes.
4. *BaseLayerPicker*: Chooses the imagery and terrain to display on the globe.
5. *NavigationHelpButton*: Displays the default camera controls.
6. *Animation*: Controls the play speed for view animation.
7. *CreditsDisplay*: Displays data attributions. Almost always required!
8. *Timeline*: Indicates current time and allows users to jump to a specific time using the scrubber.
9. *FullscreenButton*: Makes the Viewer fullscreen.

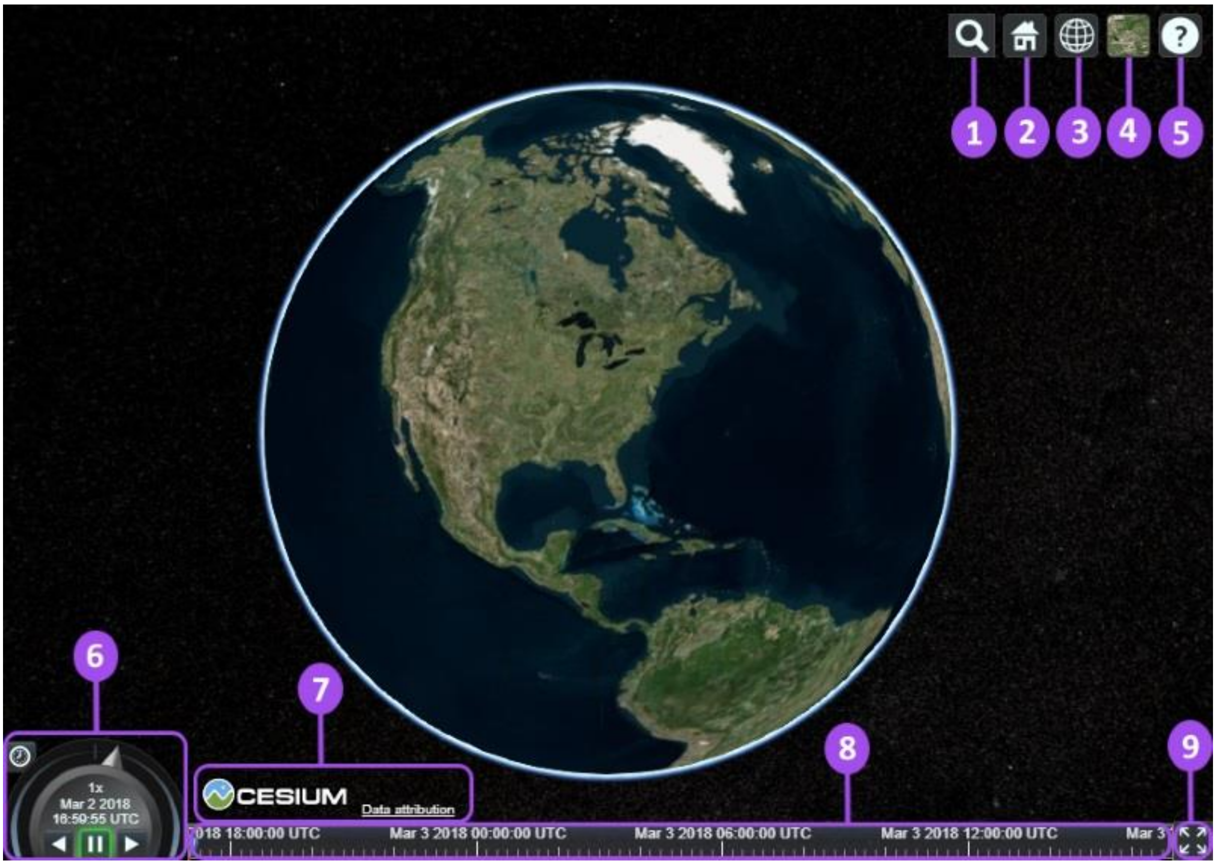


Figure 4-5. Cesium tutorial (cesium.com/docs/tutorials/cesium-workshop/).

Finally, as it was mentioned, an external sidebar menu will be included, so the user can interact with the objects in the scene. Most of these solutions are defined through the different methods and functionalities that Cesium has to access and modify items in the scene.

The Initial interactions defined for this project, according to the objectives, are:

- Day/Night mode, considering that the objects might be perceive differently depending on the day-time.
- Some additional features to the scene (e.g. trees or streetlights), apart from the 3D models of the buildings, which are defined as the main items in the scene.
- A drop-down list with the adaptive textures to apply in the buildings. This tool might have the possibility to change the base map, depending on the aspect texture selected, which will adapt the entire scene to an aesthetical NPR model.

This web-based solution is originally implemented for desktop screens, which means that not further initial implementation with responsive design will be employed. Considering too, that the amount information (3D models and textures) might be too difficult to render in mobile devices.

4.4. Evaluation

The final step in this methodology is the implementation of a user test, as map displays are tools for communication of geographical information to the user, in order to improve the effectiveness and efficiency of this tool(van Elzakker & Delikostidis, 2010). Based on this idea, the use of the UE define a set of basic metrics for evaluating the usability, such as, learnability, memorability, efficiency error, and objective satisfaction (Roth et al., 2017; Roth, Ross, & MacEachren, 2015). However, this is not the only approach to evaluate a UI. According to Roth et al., 2017; Roth, Ross, & MacEachren (2015), the use of the term the tree U's user, utility and usaulity (Figure 4-6), might give a widen overview and introductory evaluation in this project.

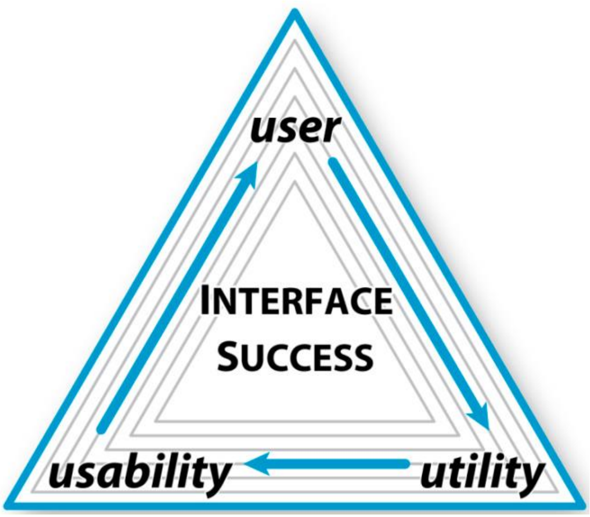


Figure 4-6. The Three U's of Interface Success (Roth et al., 2015).

The primary results will be derived from the use of the term *Utility*, which approaches of evaluation falls into two strategies “benchmarking task” and “analytical products” (Roth et al., 2015). As is stated in this study in 2015, the first strategy employ the following concepts:

- *Cognitive operation*, which defines the identification of one data element for then be compare with two or more data elements.
- *Search target*, encapsulates the information content under consideration, including space, time and attributes.
- *Search level*, which is an additional aspect of information content that considers the percentage of all map items under the idea that along a continuum spatial analysis exists a switch from elementary to general.

In terms of the Analytical Products, this will be derived by the user when employing the interactive map and will vary according to the user’s overall goals. Knowledge will be constructed while using the interface. Consequently, to complete the first iterative, triangular user, utility, usability relationship (Figure 4-7), a plan in to implement is described below.

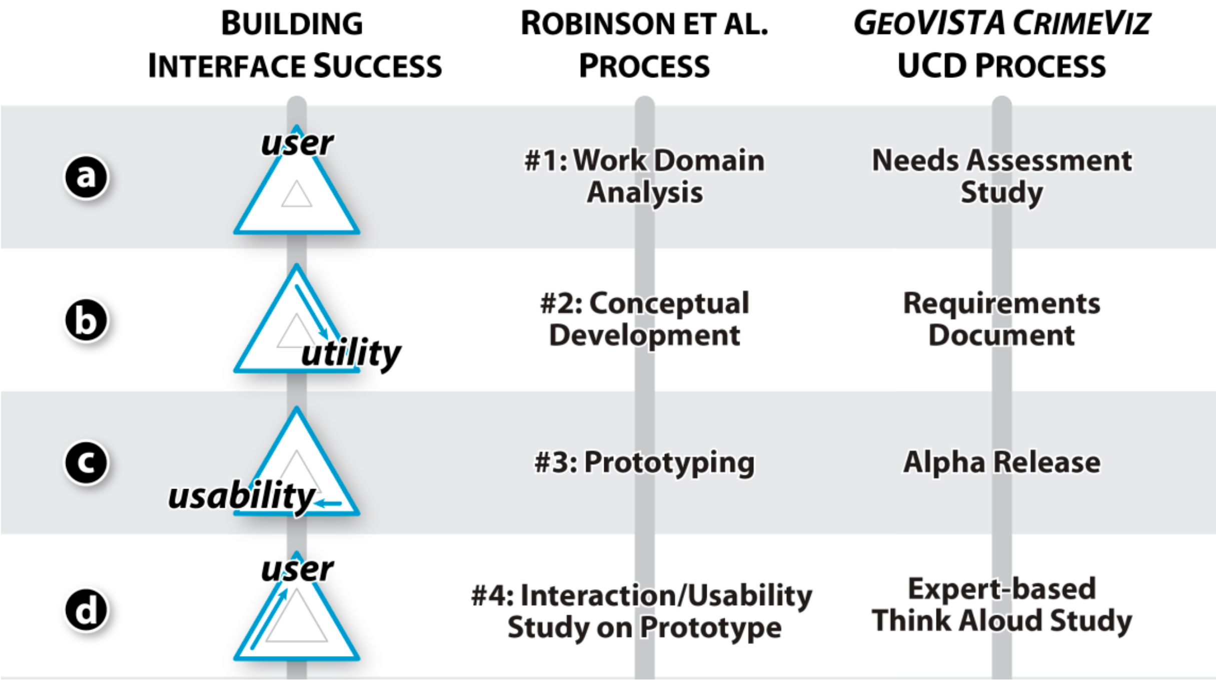


Figure 4-7. User-centred Design as an Iterative Process (Roth et al., 2015).

Based on the documents provided in the Usability.gov website, this section will describe a test plan for conducting a usability test during the development of adaptative texture application. The goals of utility testing include establishing a baseline of user performance and understanding, establishing and validating user performance measures, in order to improve the efficiency, productivity, and end-user satisfaction.

The usability test objectives are:

- To determine cognitive operation and search target, when applying different category of aspects to the façade's buildings in the main campus of the TU Dresden. Also, to locate potential sources of error, which may include:
    - Navigation errors – failure to locate functions, excessive keystrokes to complete a function, failure to follow recommended screen flow.
    - Presentation errors – failure to locate and properly act upon desired information in screens, selection errors due to labelling ambiguities.
    - Control usage problems – improper toolbar or entry field usage.
- Exercise the application or web site under controlled test conditions with representative users. Data will be used to access whether utility goals regarding an effective, efficient, and well-received user interface have been achieved.
- Establish baseline usefulness when applying different aspect textures to façade's buildings for future usability evaluation based on a user centred design.

Since this web-based application main purpose is to help users not only to navigate through the area of study for recognition, but also to locate specific objects located in reduced areas, such as the façade of a building. Therefore, the main target is students and academics which are either familiar or need to search for a specific place without knowing the area. By applying the UE, which encourages the completion of user studies in a rapid, discount manner that involves only a small number of participants (Roth et al., 2017), the number of participants is limited to a number of 10.

Participants

Most participants are expected to have a basic experience in the use of GIS applications and maps. As it is described, they can either be familiar or not with the area of study, since most of the task are based on elements and shapes of the buildings.

The participants' responsibilities will be to attempt to complete a set of representative task scenarios presented to them in as efficient as possible, and to provide feedback regarding the usefulness of the textures. The participants will be directed to provide honest opinions regarding the utility of the application, and to participate in simultaneous-session subjective questionnaire.

Training

The participants will receive and overview of the user test procedure, equipment and software. Also, some deficiencies in the 3D models, which are already located.

Procedure

The participant's interaction with the Web site/Web application will be monitored. The facilitator will brief the participants on the Web site/Web application and instruct the participant that they are evaluating the application, rather than the facilitator evaluating the participant. The facilitator will ask the participant if they have any questions.

After each task, the participant will complete the post-task questionnaire and elaborate on the task session with the facilitator. Participants will complete a pre-test demographic and background information questionnaire.

5. Implementation

This chapter examines the practical application of the aforementioned methodology, from the pipeline definition to achieve the non-photorealistic models, until the digital infrastructure involved in the final web-based UI. The preliminary work follows a similar process application employed in the project called "Concept and implementation of Contextualised Model: The Uch Enmek Example" developed by M. Abdalla (2017), which considers some of the conversion tolls implemented in this project.

As one of the fundamental objectives, it is outlined the design of the different aesthetic sketch models. For this reason, this chapter will also exemplify the various parameters implemented to produce each of the styles, in order to remove the over-cluttered visual cues in the images. As final interaction for the user some pseudo-algorithms to render the information in the scene are reported, which in a final phase contrive to include all the aspects in a Cesiumjs interface.

By the shown subsequent workflow (Figure 5-1) shows the process is divided in three main sections described as well in different levels. The backbone of this development will concentrate on the storage of the 3D building models in their different categories of thematic representations. Eventually this information will be transformed in the next phase, using the program developed by Kronos Group named *COLLADA2glTF.exe*. Once this information is properly, all this data is concentrated in different JavaScript functions to transfer the models into the scene. The input style texture selected will constrain the files added to the loop, making the dynamical change in the UI.

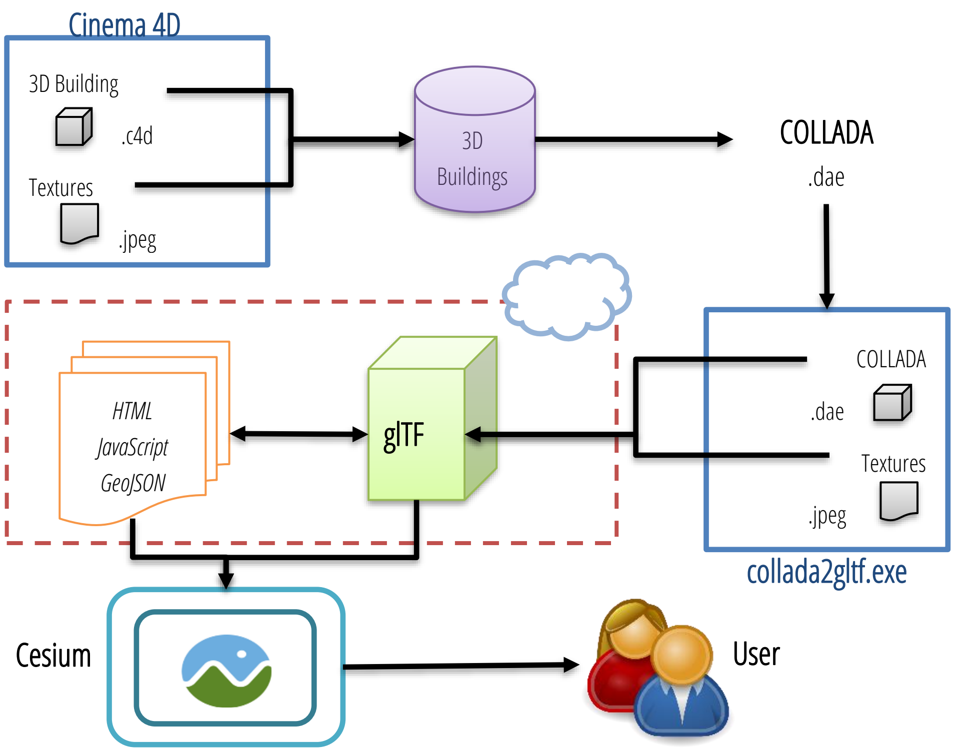


Figure 5-1. General workflow for the web-based application for the project.

5.1. 3D Models

5.1.1. Cinema 4D

The development of this project is set and based on the area of extent represented in Figure 5-2, which coverage was built up by the working group Visualization, Institute of Cartography and Centre for Information Services and High Performance Computing (ZIH) in the TU Dresden (Figure 5-3). Among the buildings available an initial sample of 8 buildings was selected to apply a general workflow. This 3D model is stored in Cinema 4D, which will be the preliminary stage for the further dynamic implementation in the following steps.

Figure 5-2. Technische Universität Dresden- Area of study (Aerial image from Google).

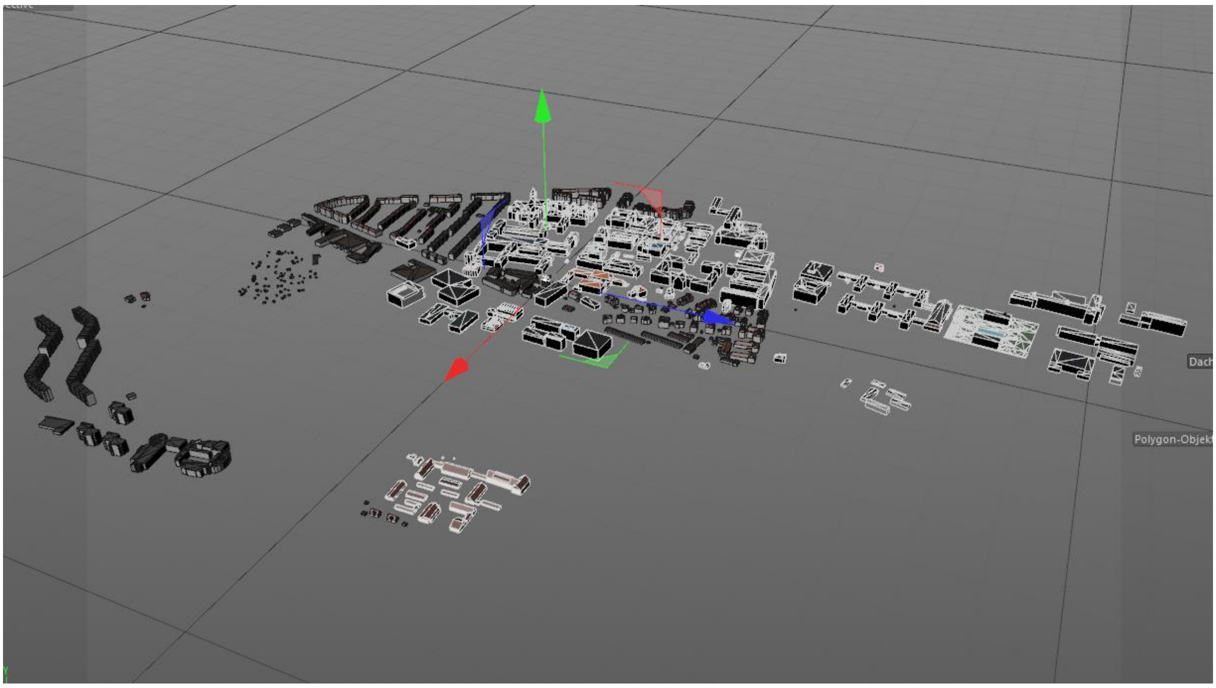


Figure 5-3. Buildings selected, correspond to the Technische Universität Dresden (Project in Cinema 4D TUD ZIH).

Each of the developing procedures is described step by step the process to get the 3D models. In order to follow the workflow model proposed, prior to the final aspect design, some of the most important architectonical elements of the façades were extracted from the first sample of buildings. These selected buildings are expected to contain a representative number of shapes materials and structures to transfer the style to a larger area.

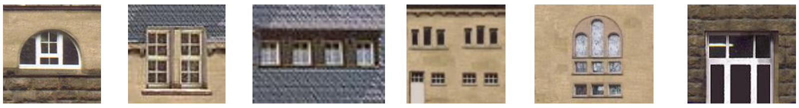
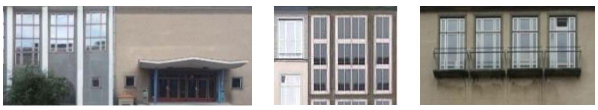
Building	Important Characteristics
Hülsse- Bau	15 images (windows, doors, material and colour) 
Tillich- Bau	26 images (windows, doors, material and colour) 
Georg-Schumann- Bau	50 images (windows, vegetation, material and colour) 
Georg-Schumann-Str.7a	5 images (metallic structures and illumination) 
Von-Mises- Bau	11 images (windows and doors) 
Merkel- Bau	16 images (windows, doors, material and colour) 
WT- Halle	3 images (metallic structures and illumination) 
Barkhausen- Bau*	56 images (windows and doors) 

Table 5.1. First selected sample to structure the final workflow.

*This building contains most of the examples; therefore, will be the testing building.

Once the classification is set, it is possible to start the 3D model design; for this purpose, each of these buildings was exported from the main *c4d* file and separately stored. Subsequently, the geometric element is then associated to their image captions, which are a collection of images. These images are used as materials in the software, which subsequently is related to the different faces of the geometry (Figure 5-4).

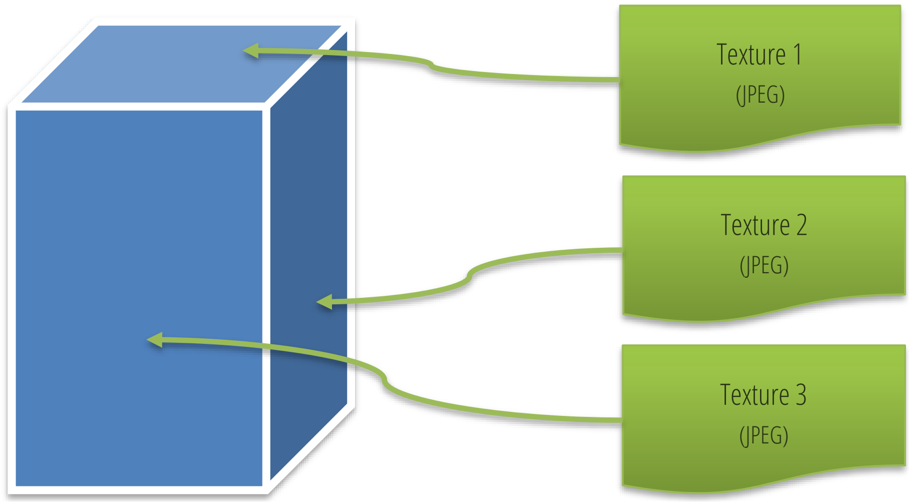


Figure 5-4. Binding of the textures to each façade of the buildings.

As the original project contains its own coordinate system, the exported the information to a new project, the objects will preserve an origin set for each building, which changes the spatial reference and does not match with the root node in the new project. Additionally, all anchor points of the 3D models are defined by the projection of its centroid projected at lowest level of the object. That is when, each object is transported to the *origin point* (Figure 5-5) in the coordinate system, namely coordinates [0,0].

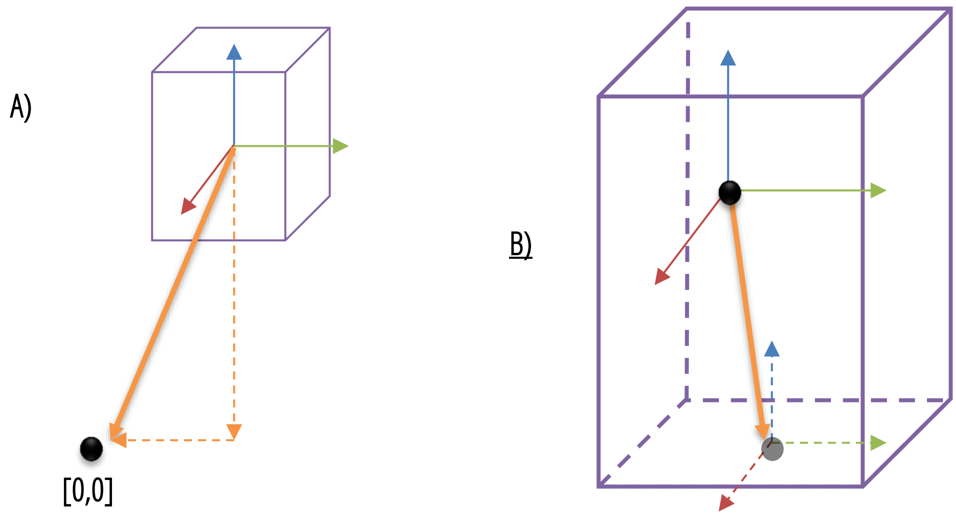


Figure 5-5. a) Translation of the object to the project origin, b) anchor point is set in the centroid projected at the lowest level of the object.

These parameters are necessary to assure the proper location of the models in the screen, as each building will have an associated pair of coordinates of origin in the Cesium display.

As a complementary part of the scene, the design of the additional features, such as the trees and streetlights, become an essential task to provide a wider context about the area for the user. The design of the streetlights (Figure 5-6) was taken from the tutorial provided in the class "Virtual 3D Landscapes" in the TUD during the Winter Semester 2018-2019.

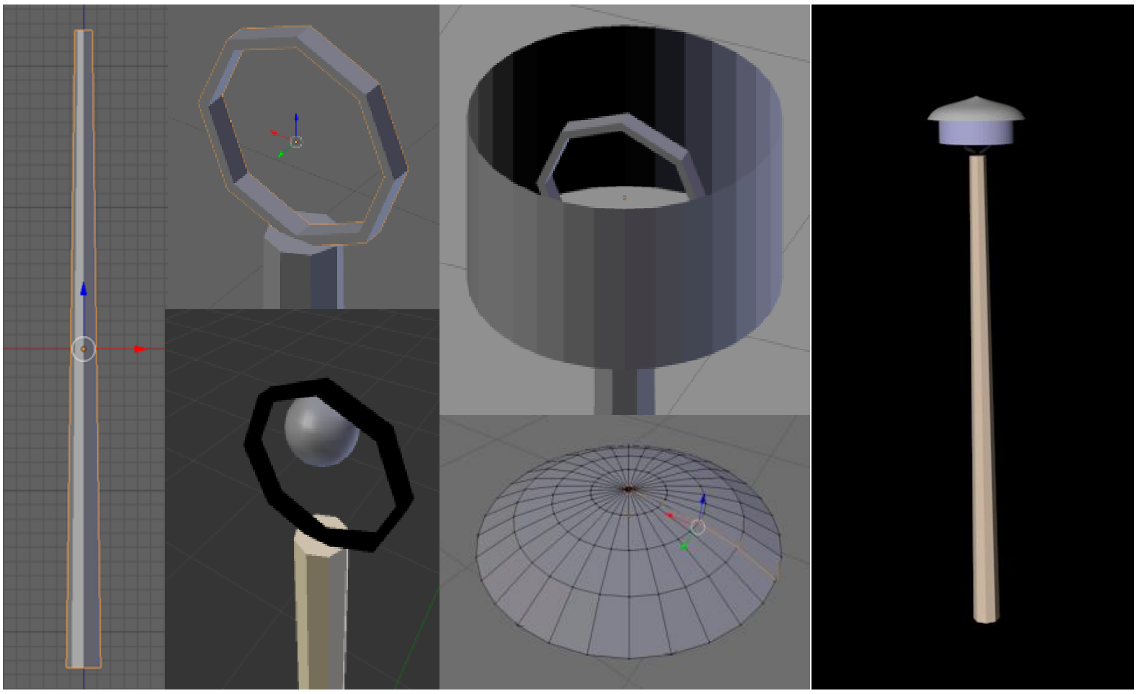


Figure 5-6 Streetlamp design process and final 3D object outcome in Cinema 4D. Information courtesy by Dr.rer.nat. Nikolas Prechtel.

As for the trees, the basic shapes in Cinema 4D were taken and subsequently, modified. It is worth mentioning that the basic structure of the tree trunk contains a higher detail compare to the crown of the tree, which design is based on a sphere. This mentioned detail considers the further possibility to include an additional concept to the scene, as are the seasons; therefore, the visual vegetation will adapt as well to this new conception (Figure 5-7).

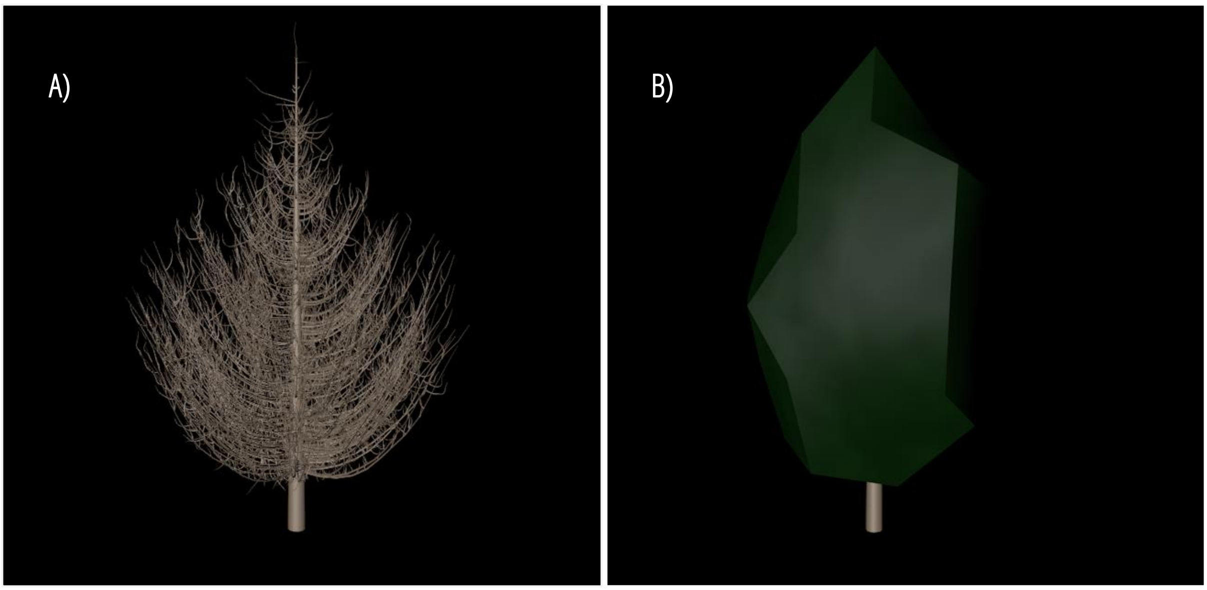


Figure 5-7. A) Trunk designed using the rail spline sweep, then divided in 10 parts to use the hair objects to the sections in Cinema 4D (tutorial: https://bit.ly/2lF9qFI). B) Design of the tree trunk and the addition of the crown.

5.1.2. GeoJSON

As it is defined in the website, GeoJSON[2] is a format for encoding a variety of geographic data structures (Figure 5-8), which supports the following geometry types: Point, LineString, Polygon, MultiPoint, MultiLineString, and MultiPolygon. Geometric objects with additional properties are Feature objects. These sets of features are contained by FeatureCollection objects. Thanks to its properties, there has been an increasing trend in the use of GeoJSON format and enjoys great support in China and Singapur, whose common features are textual representation, simplicity, good readability, small size and simple usability in the majority of programming languages (Orlik & Orlikova, 2014).

```
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [125.6, 10.1]
  },
  "properties": {
    "name": "Dinagat Islands"
  }
}
```

Figure 5-8. GeoJSON structure format.

The use of this format mainly would contain the coordinates of origin, where the 3D objects will be located in the scene, whether the buildings or additional features. These coordinates are directly retrieved from the map in "*Geoportal Sachsenatlas*". However, this implementation not only supports the storage of the coordinates, but also it can store further detailed information about the buildings. In order to create the code source with the attributes the website geojson.io (Figure 5-9) was used, which gives the possibility to edit the attributes like table.

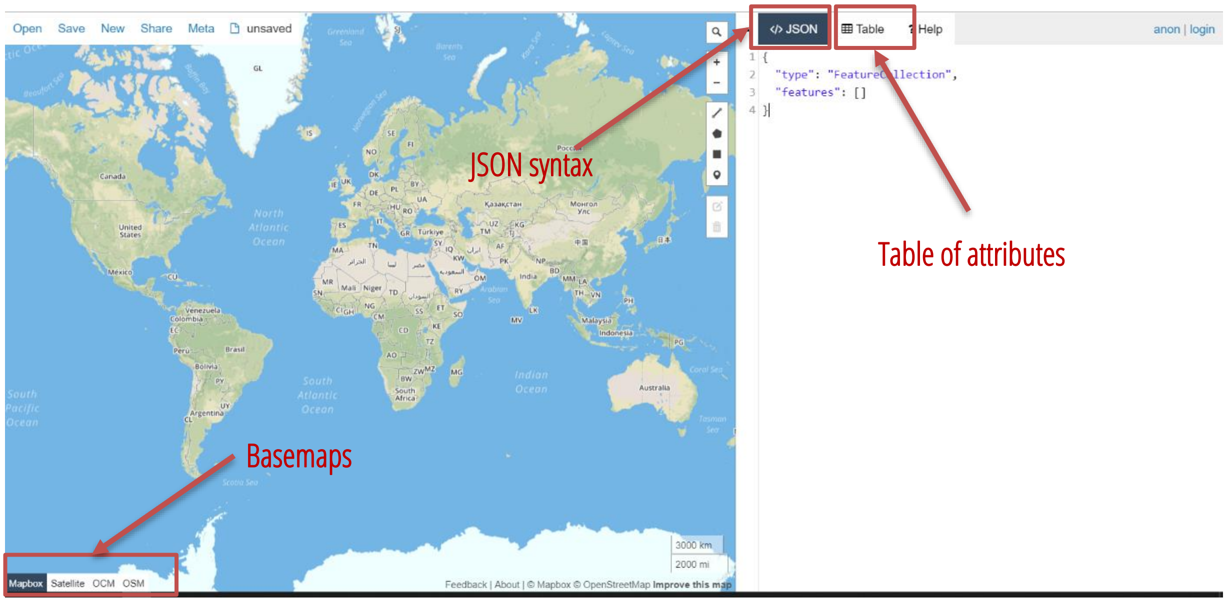


Figure 5-9. Web tool to create GeoJSON files (http://geojson.io).

[2] http://geojson.org

This web tool allows a wide range of formats, such as *csv*, *shapefiles*, which has a table structure; moreover, it has editing tools that support point and polygon geometries. The JSON syntax to all that is created in the map and the table has the possible formats to export the data: GeoJSON, TopoJSON, csv, kml, among others.

5.2. Conversion tools

5.2.1. COLLADA

Just as glTF format, COLLADA is a format managed by Kronos Group (Khronos Group, n.d.), who defines the format as an XML-based schema to make it easy to transport 3D assets between applications - enabling diverse 3D authoring and content processing tools to be combined into a production pipeline. The intermediate language provides comprehensive encoding of visual scenes including geometry, shaders and effects, physics, animation, kinematics, and even multiple version representations of the same asset.

Cinema 4D is a creation tool software for this format, which for this project the version 1.4 is used, as this version is compatible with the conversion tool (*collada2gltf.exe*) for glTF. Such format needs to be stored in the same folder as the images of the façades, since this item contains the local path to this information.

5.2.2. From COLLALDA to glTF

This tool is a web development that converts COLLADA (.*dae*) models or OBJ models to glTF for further use with Cesium[3]. It is possible either to use the all files (. *dae* and JPEG) to create the glTF object or to save them in a zip file. Nevertheless, this web tool has a limit of 10MB for the number of files loaded, still they have the possibility to download the command-line tool to convert the files locally.

Each separated format created is stored in a source folder, which in the command- line of the main JavaScript application will be called later on in Cesium, After creating the first sample of buildings, it has been detected that this tool sometimes presents some glitches when creating the glTF files by generating new nodes in the wireframe of the geometry (Figure 5-10), this makes the textures to be partially rendered.

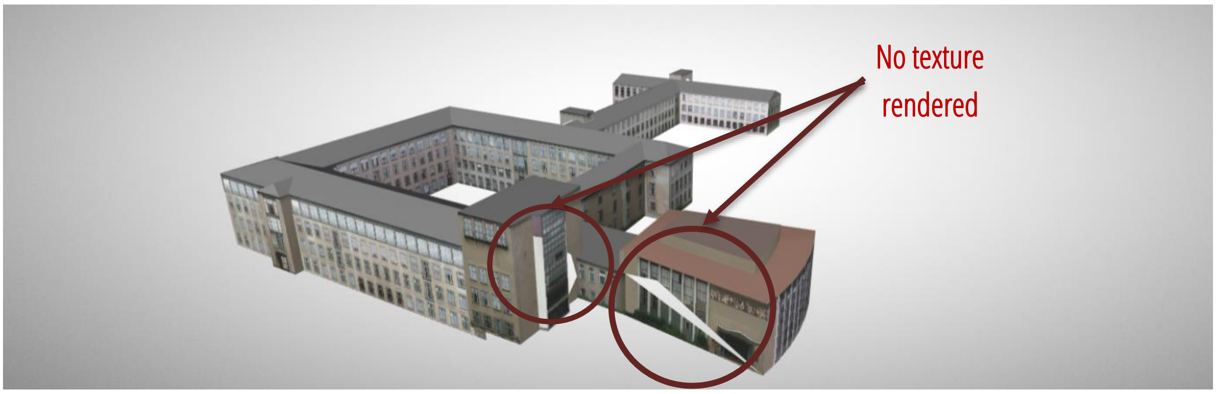


Figure 5-10. Glitches in the final models (Barkhausen-Bau).

[3] http://cesiumjs.org/convertModel

5.3. Aspect texture design

According to Semmo (2016), a *Focus + Context* describes the concept to visually distinguish between important or relevant information from closely related information by enabling the user to interactively change visual representation data for points or regions of interest (details-on-demand). This statement supports the intention of the aforementioned styles in Chapter 3 will be the dynamically rendered on the user's demand.

5.3.1. Photoshop

Although the proper 3D model pre-process is essential to the correct navigation model in the scene, the aspect texture design is the most fundamental part of this project. The following 3 styles' parameters were created in Photoshop, in order to construct a batch script function to run in all the façades. The following steps will be eventually applied to the extend the area of study.

Artistic

The design of this style is based on the Watercolour design created by maps.stamen.com, tile-provider. This idea arose from the different implementations of the NPR examples (Figure 5-10). Among NPR implementations, the watercolour style (Figure 5-11option B-c), is stylistically similar to the basemap designed by maps.stamen, giving the option to implement an aspect texture based on this technique.

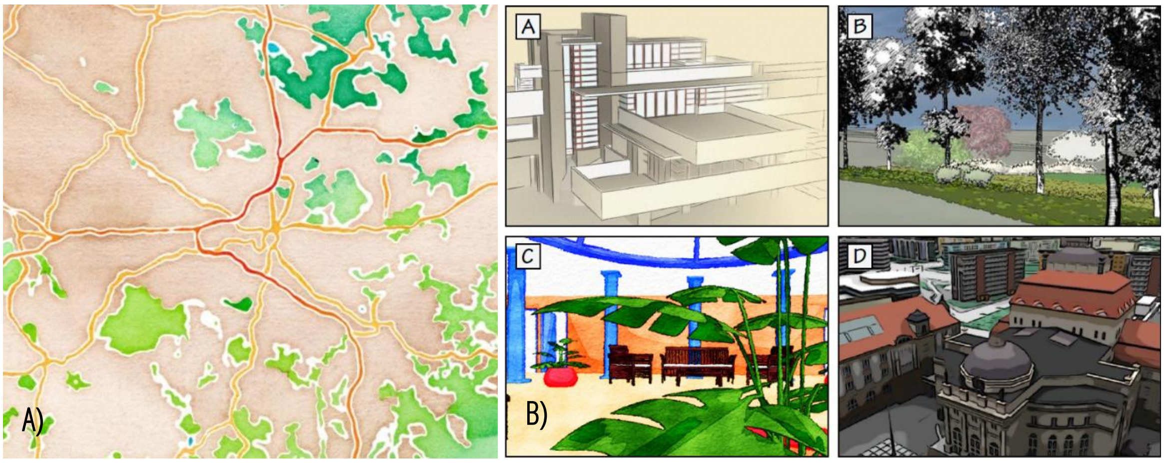


Figure 5-11. A) Watercolour basemap (http://maps.stamen.com). B) Examples of non-photorealistic rendering techniques for 3D geospatial models (Semmo, 2016).

Photoshop provides many different filters to apply in images, among which the watercolour is an option. However, the single application of this filter would not reach the same level of similarity with the basemap, this is mainly caused by the amount of information inside the images (e.g. materials and overcluttered structures). Therefore, a reduction of this information was required, which at the same time gave a brush effect, just like a painting. This action was achieved by using the following steps in photoshop (Figure 5-12, Figure 5-13). The first step of simplification not only reduces the amount of information of the architectonical structures, but also creates the feeling of brush strokes in a painting. Additional to this process comes the application of the watercolour effect and the complete change of colour, to obtain an outcome that finally matches with the basemap.

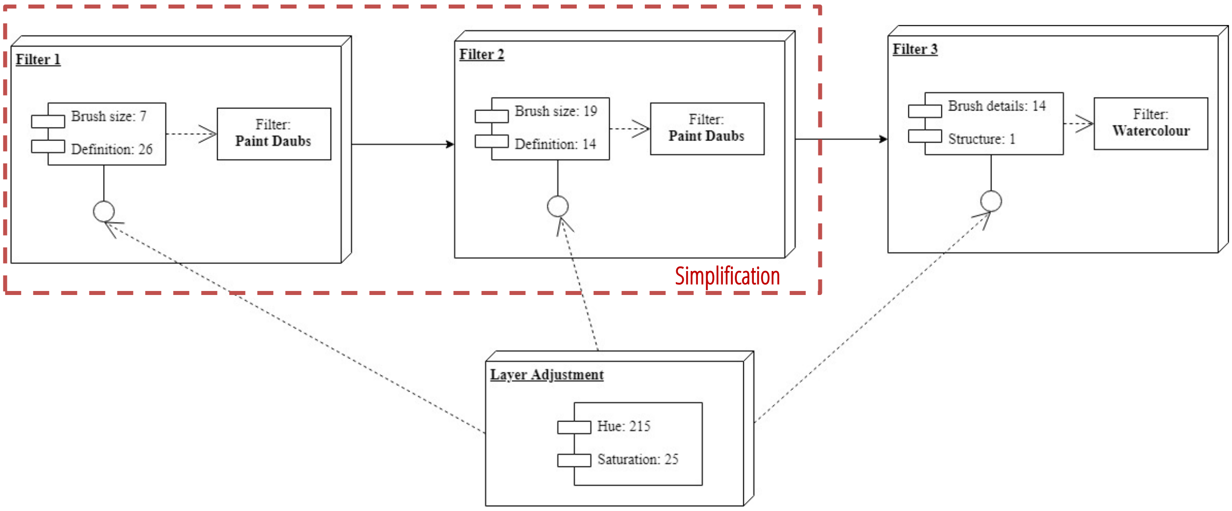


Figure 5-12. Workflow to create the Artistic Style.

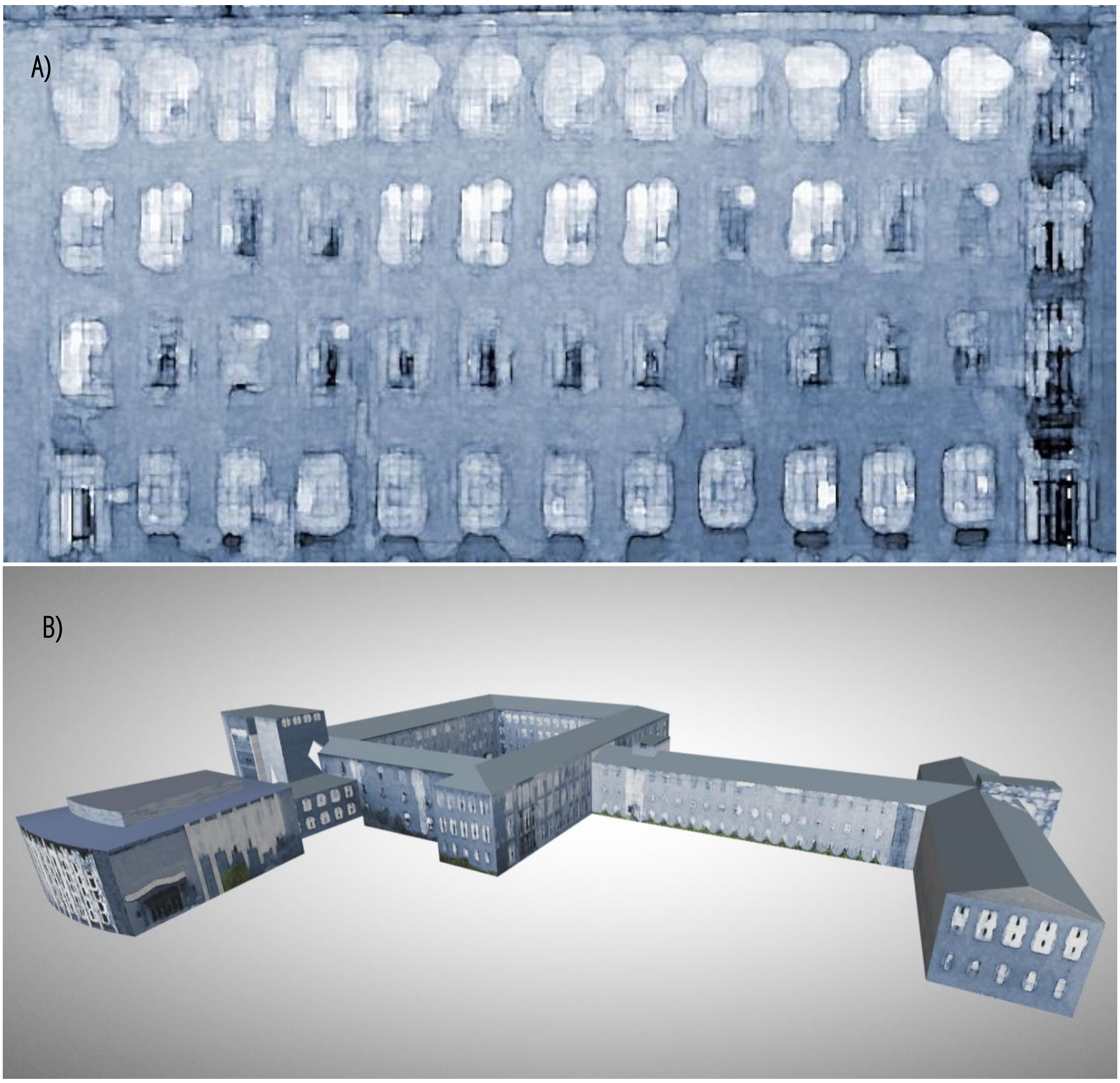


Figure 5-13. 3D model with the Artistic Style (Barkhausen-Bau). A) resulting façade's image, B) resulting building.

Pencil

Many algorithms have been developed to create the visual effect of a pencil-based drawing, every time with a more and more realistic effect in computer vision (Figure 5-14). The employment of this technique has been mainly applied to static images, more than 3D models. The engaging side is the development in the effects applied in long and short lines in the drawings (AlMeraj, Wyvill, Isenberg, Gooch, & Guy, 2009), as well as some basic considerations in the image processing to execute the line extraction in grey scale images (Wang, Bao, Zhou, Peng, & Yingqing, 2002).

The study result shown in Figure 5-14A has closely analysed different outcomes from real line drawings and its change in thickness and hue, depending on their length, whereas the result in Figure 5-14B is more focused on the reconstruction of the stroke paths applied in monochromatic images, to give the effect of real pencil strokes.

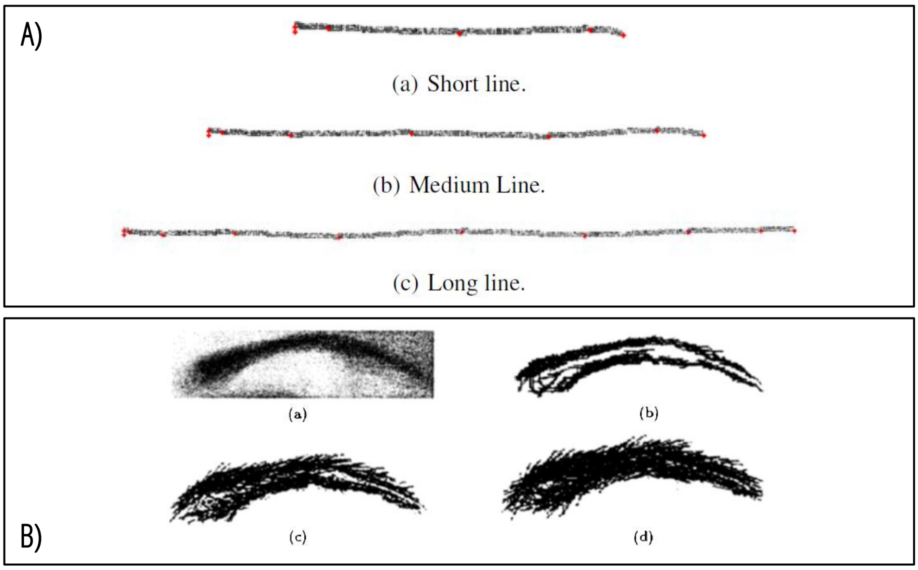


Figure 5-14. Computer generated pencil effect. A) Generated lines and control point positions for varying lengths (AlMeraj et al., 2009), B) Reconstructed stroke paths(Wang et al., 2002).

Some implementations in 3D models have given outstanding results, using a real-time rendering, using more complex and sophisticated algorithms to generate this effect. Nevertheless, these techniques are lead in a direction towards an object-based detection (Semmo, 2016), rather than an image-based approach (Figure 5-15).

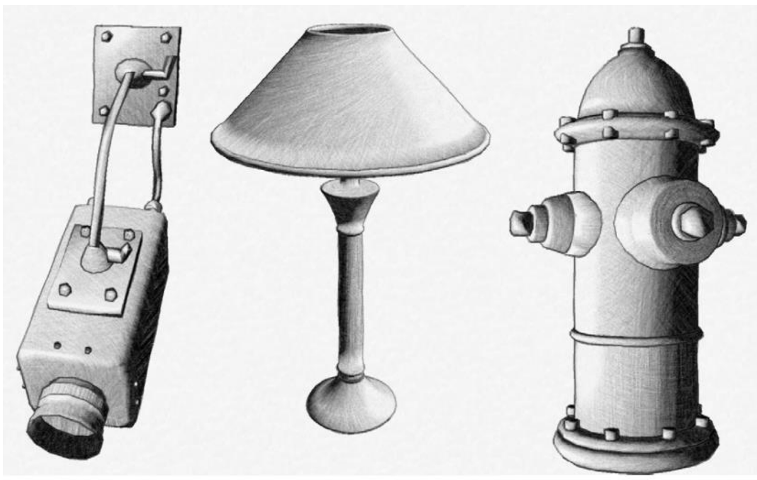


Figure 5-15. Pencil real-time-rendering results (Lee, Kwon, & Lee, 2006).

Unlike the artistic style, the steps to accomplish a convincing effect of pencil with photoshop requires more tools and functionalities from the software. In consequence, a more complex workflow (Figure 5-16) is constructed to create the batch script to automate the image generation. This approximation is based on the website *Photoshopessentials*[4] (Figure 5-17).

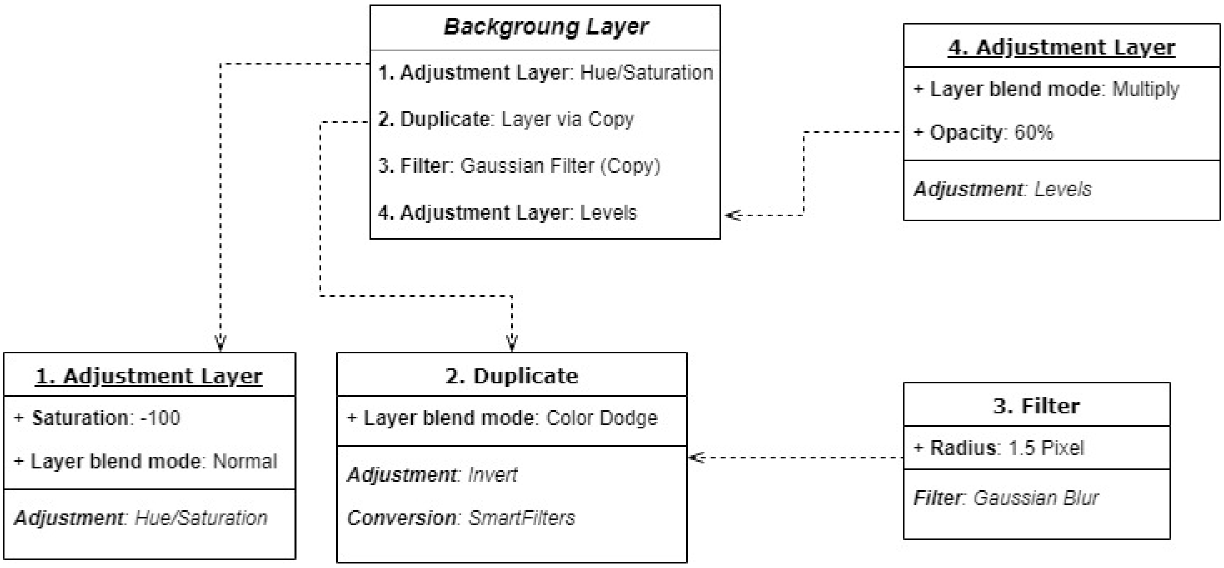


Figure 5-16. Workflow to create the Pencil Style.

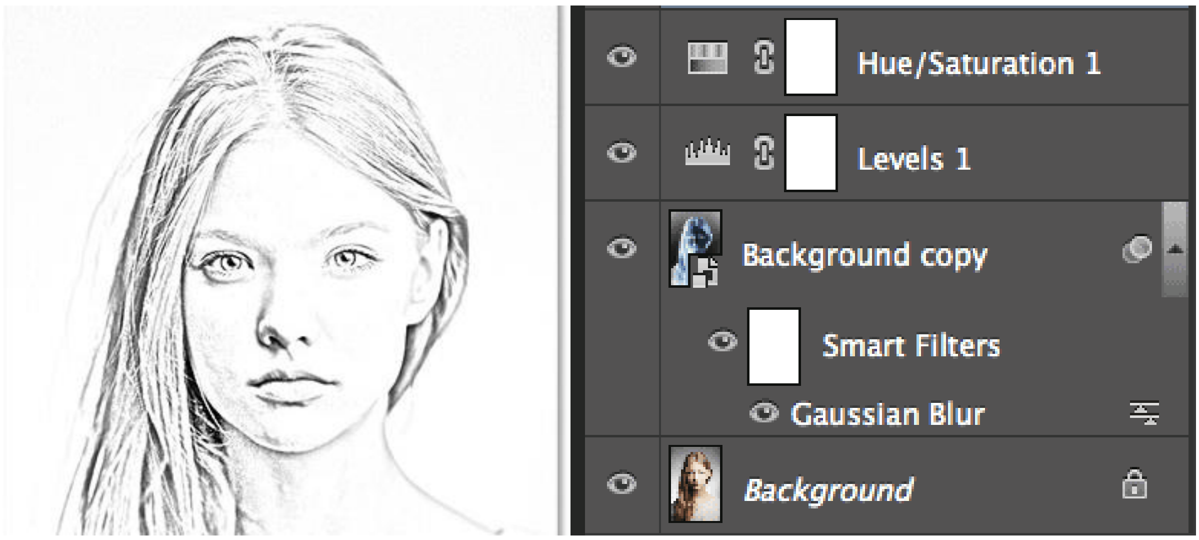


Figure 5-17. Outcome image by applying the methodology in the website *Photoshopessentials* and layer order in Photoshop (https://bit.ly/2Zle7rc).

The following images show on one hand the result in one of the images of the façades (Figure 5-18), which applied to the complete 3D model, the final result when binding all the images is presented in Figure 5-19. It is important to mention that the images that does not contain any structure to detect, that means, it only represents a plain wall, sometimes present some noise depending on the roughness of the material or illumination of the image (if perceivable).

[4] https://www.photoshopessentials.com/

Figure 5-18. Façade image generated by applying the Pencil Style.

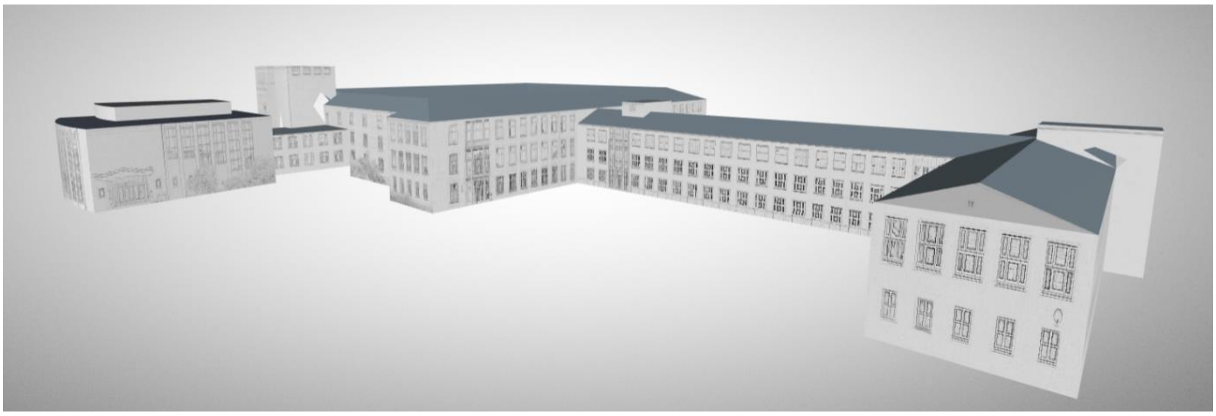


Figure 5-19. 3D model in the Pencil Style.

Contour

The edge enhancement technique is a well-researched topic in NPR. For this reason, in order to detect edges, the study carried out by Semmo (2016) distinguishes between gradient-based edge detection that thresholds the gradient magnitude of an image and Laplacian-based edge detection that identifies zero-crossings in the second derivative, his study discusses the use of use G-buffer information to detect discontinuities in depth to extract contour lines (Figure 5-20). The results of this technique produce a convincing and clean effect to clearly identify objects in the scene.

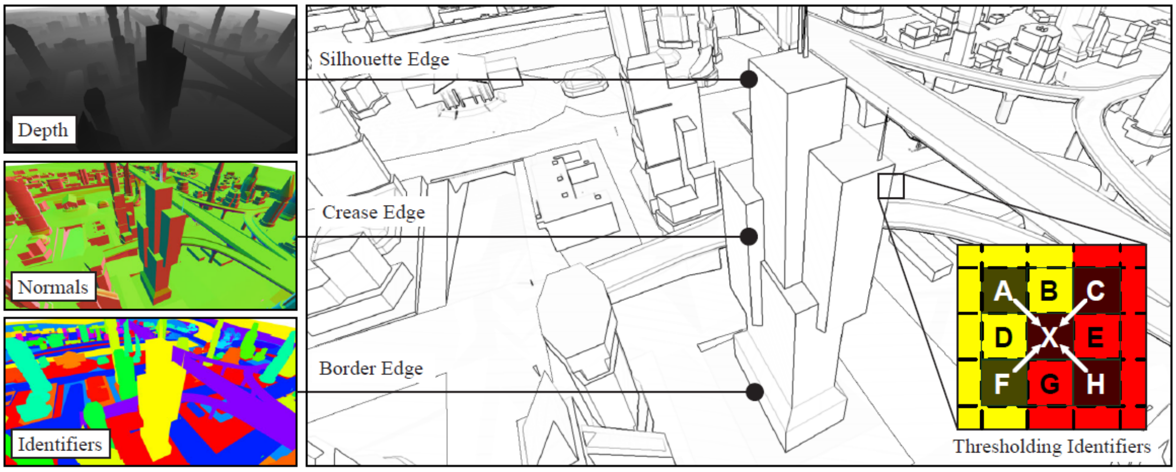


Figure 5-20. Edge enhancement in image (Semmo, 2016).

The last style designed using the tools in photoshop is the edge enhancement, which purpose, unlike the pencil-based style, is to represent the structures in the images using well-defined lines. In order to carry out this task, one specific tool in the software was used "*Find edges*", which gave good results without using any other layer or method to refine the resulting image (Figure 5-21). According to Adobe website[5], the "*Find edges*" filter identifies the areas of the image with significant transitions and emphasises the edges.

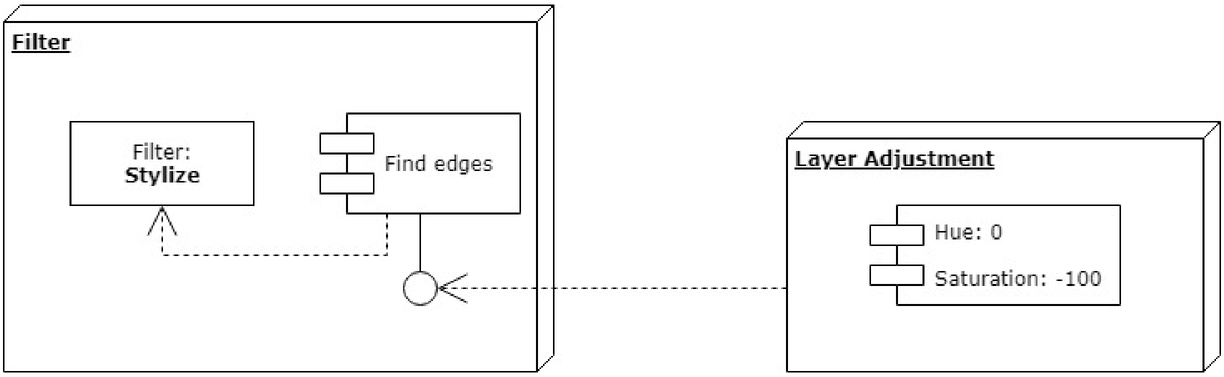


Figure 5-21. Workflow to create the Edge Enhanced Style.

The implementation of this method provides a good overview of the elements in the scene, giving the possibility to distinguish some elements in the façades (Figure 5-22). However, some of the elements, which are only extra information might capture the user's, deviating the attention from the important information. The resulting model (Figure 5-23) grant one more approach of sketch-based design.

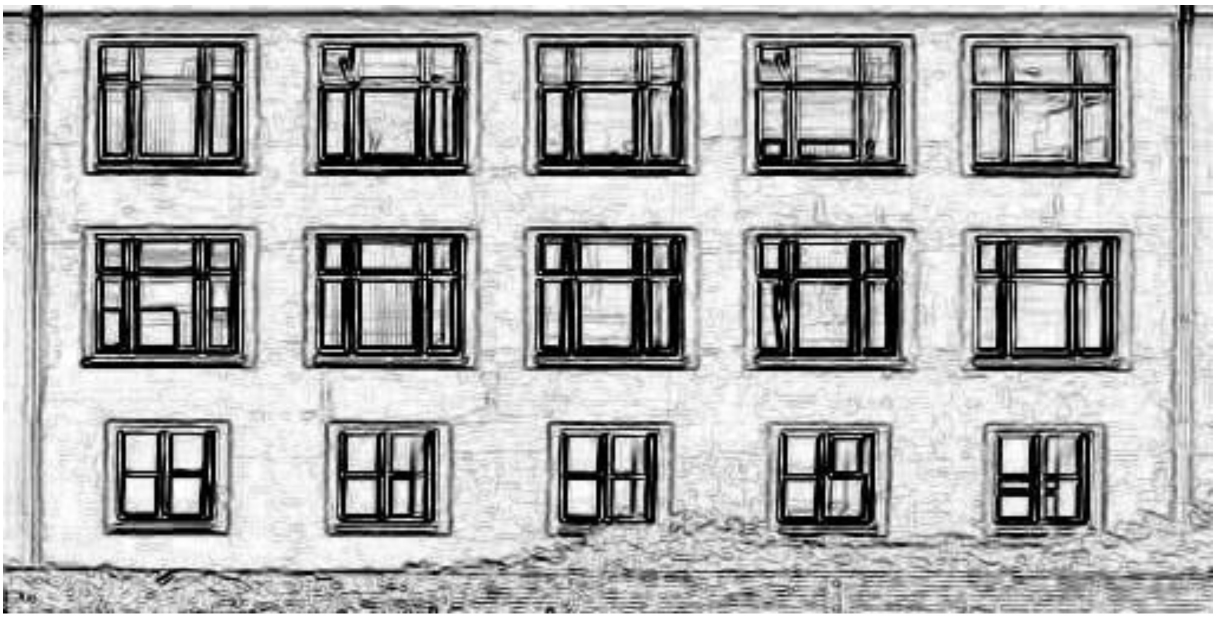


Figure 5-22. Façade image generated by applying the Edge Enhanced Style.

[5] https://adobe.ly/2DDhGte

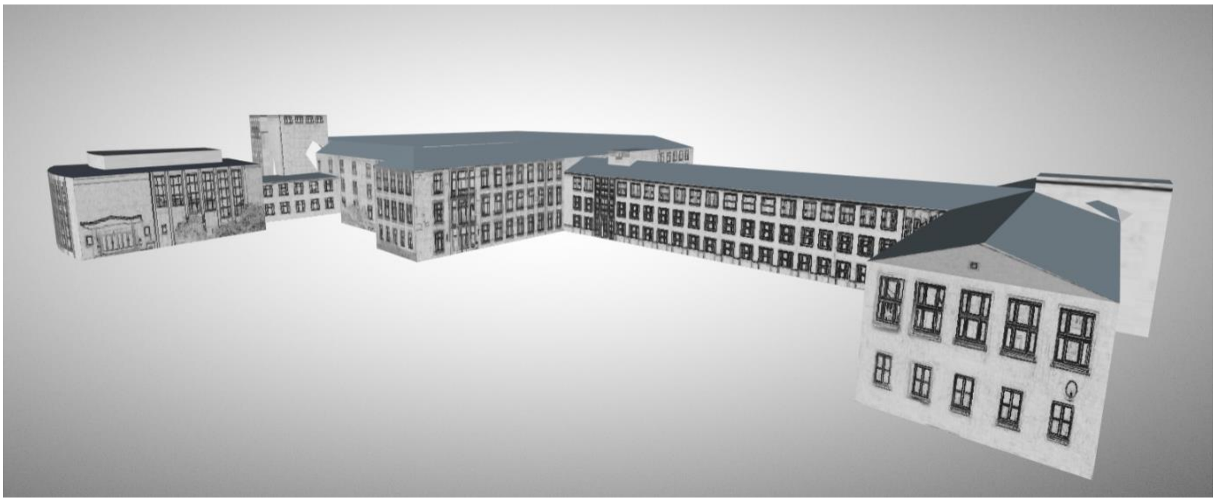


Figure 5-23. 3D model in the Edge Enhanced Style.

5.3.2. Neural Style transfer

The latest inclusion in the sketch-based styles is the implementation of the Neural Style transfer. Chapter 3 provides the background behind the use of this technique to transfer the the style of an input image into a target image, in which case, the target image will be the façades of the buildings. The application of this concept was computed using an approach published in GitHub called "*Deep Art Effects API Example For JavaScript*" (Figure 5-24), which is an example HTML-JavaScript webpage and web app for neural style transfer developed by Deep Art Effects Developer[6].

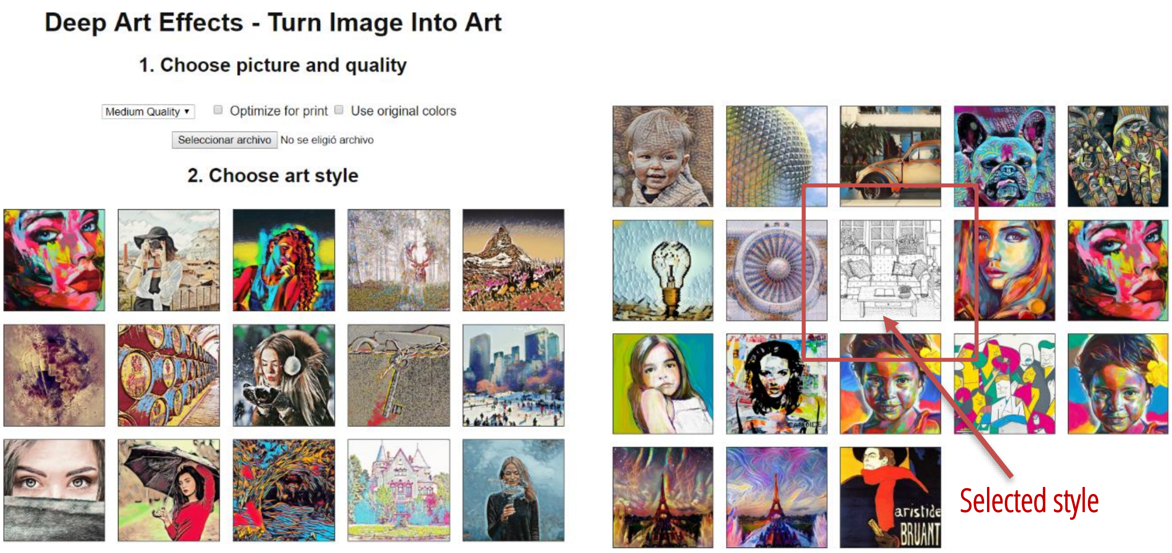


Figure 5-24. Showcase Website from Deep Art Effects Developer (http://showcase.deeparteffects.com/).

[6] https://developer.deeparteffects.com/

This company grants a downloadable folder with this web solution example, which handles one image at a time. It is possible to register, in order to use a trial API key for 14 days. Among the options offered in the list of styles, the white and black sketch was selected (Figure 5-25).

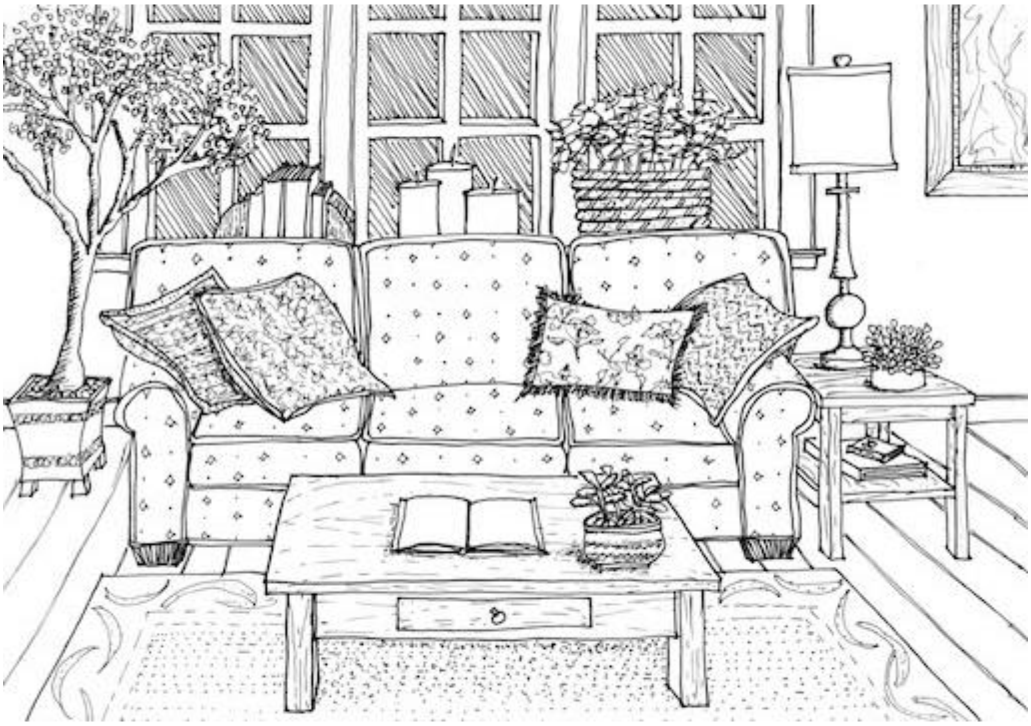


Figure 5-25. Showcase Website from Deep Art Effects Developer. List of styles (sketch style) (http://showcase.deeparteffects.com/).

The resulting images represent the structures with sharp lines (Figure 5-26), yet it stills contain the noise generated by the different materials, as well as it generalises most of the details of the elements in the façades. Each image analysed separately might not provide much information about the elements; however, the combination of all of the in the 3D model (Figure 5-27) might help the user to navigate through the scene.

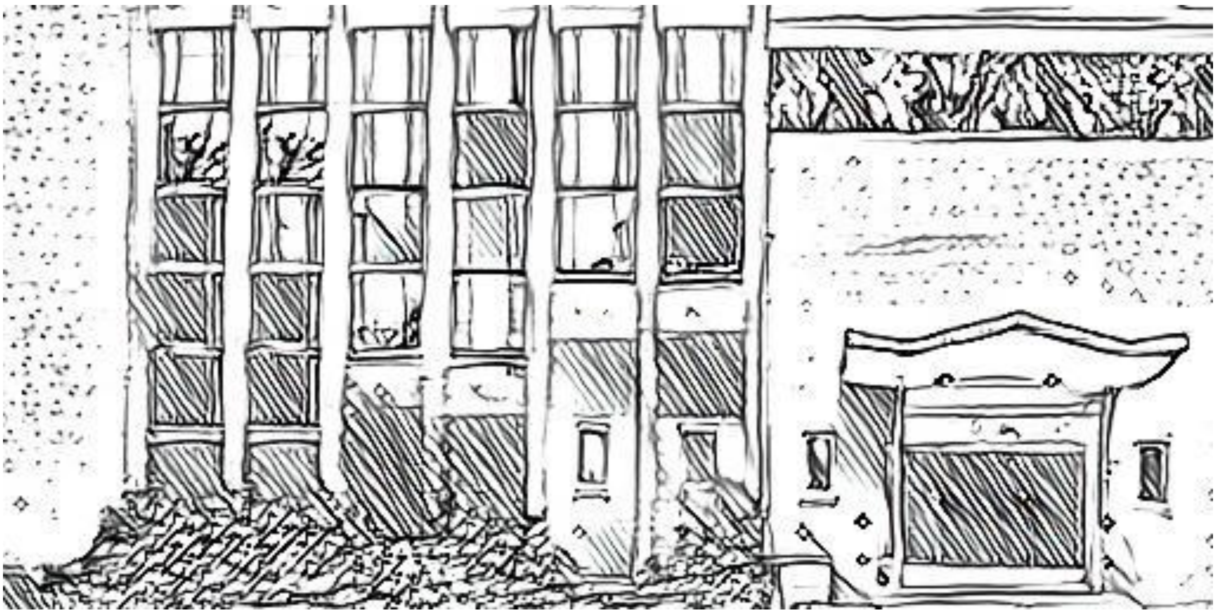


Figure 5-26. Façade image generated by applying the NST Style.

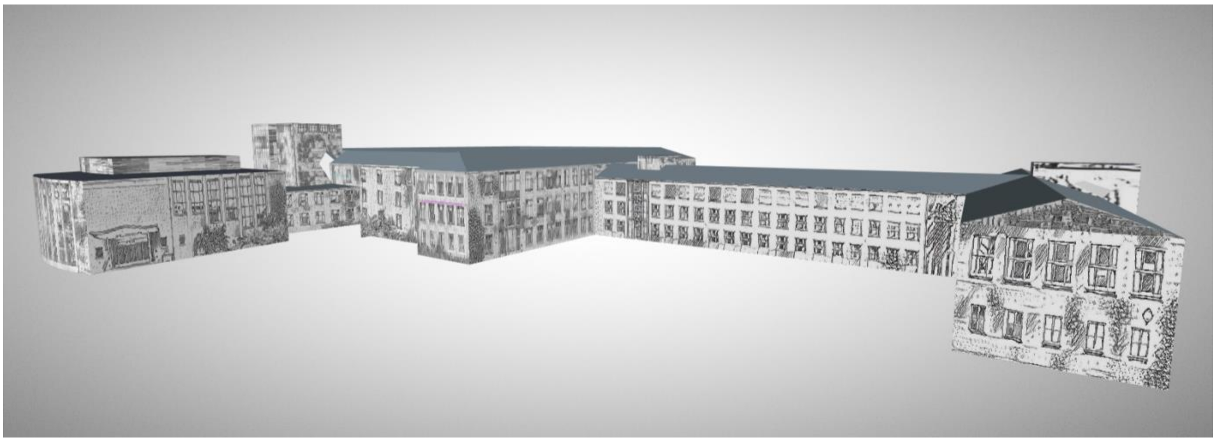


Figure 5-27. 3D model in the NST Style.

5.4. Initial setup visualization

After designing the workflow for the 3D models and each of the texture style, the area of study was extended to a number of 32 buildings. A further step in order to render the all information in the scene properly is the setup framework. This section describes the initial parameters (Table 5.2), which description in the website of Cesium is:

- *Viewer setup*. A base widget for building applications. It composites all of the standard Cesium widgets into one reusable package. The widget can always be extended to add functionality, useful for a variety of applications
- *Initial Position*. The position of the camera.
- *Initial Orientation*. The orientation forms an orthonormal basis with a view, up and right = view x up unit vectors.
  - Heading is the rotation about the negative z axis
  - Pitch is the rotation about the negative y axis
  - Roll is the rotation about the positive x axis

Parameter	Property	Value
Initial Position	Latitude	51.0215°
	Longitude	13.72055556°
	Height	200 m
Initial Orientation	Heading	25°
	Pitch	-13°
	Roll	0°

Table 5.2. Definition of the initial parameter's values

The first information visualised in the globe are the 3D models in the photorealistic styles, which can be dynamically modified by the user in the *Menu* button. The load of the data into the scene is executed in a For-Loop, by accessing the glTF files and associating their names to the corresponding coordinates (Figure 5-28):

Load 3D Building's models

Function CreateModel (Texture Style, Wireframe)
Input: The texture style and the initial state of the wireframe
Output: 3D models rendered as primitives into the scene.
promise API access to the GeoJSON file with the coordinates
then data source is fragmented in the different attributes
sourceFolder ⟵ Complete the path to the folder with the selected texture
    for each feature in the GeoJSON file **do**
        origin ⟵ retrieve the coordinates stored
        modelMatrix ⟵ Transform origin to eastNorthUpToFixedFrame
        glTF ⟵ assign parameters: origin, texture and wireframe
    return *glTF*
    end for
end function

Figure 5-28. Algorithm to load 3D Building's models

In addition to the 3D models of the buildings in the initial setup of the scene (Figure 5-29), the 3D model of the additional features (trees and streetlights) are added to the scene using the previous algorithm but using only the coordinates of origin stored in an instance array. The scene is set in the *Day Mode* at the time 10:00 h. The last tool included into the scene is the "*Clock interaction*" tool, which allows the user to smoothly change the time hour and illumination in the scene.

Figure 5-29. Initial setup of the Web App.

6. User test results

6.1. Methodology and data collected

The evaluation of the web-based application was conducted in the facilities of the TU Dresden. During this evaluation, 10 participants, matching the user profiles described in Chapter 4, were asked to spend 30 minutes with the web site. During this 30 min, the participants:

- Completed a user background questionnaire
- Performed hypothetical task on the site
- Answered questions about the perception of the different aspect texture styles
- Gave some feedback on some of the additional tools in the App.

The ten participants have the following profile characteristics:

Age

24-27	6	60%
28-31	1	10%
32-35	1	10%
Above 35	2	20%
TOTAL (participants)	**10**	**100%**

Gender

Women	5	50%
Men	5	50%
TOTAL (participants)	**10**	**100%**

Education level

Bachelors	6	60%
Masters	4	40%
TOTAL (participants)	**10**	**100%**

Confidence navigating in the TUD

Very	1	10%
More or less	4	40%
Not very	3	30%
Not familiar	2	20%
TOTAL (participants)	**10**	**100%**

3D City Model

Familiar.	6	60%
Not familiar	4	40%
TOTAL (participants)	**10**	**100%**

Use of 3D map

Have used them.	7	70%
Have not used them.	3	30%
TOTAL (participants)	**10**	**100%**

During the evaluation, participants were asked to complete four tasks (Table 6.1) on the site. After the following tasks, data from the users was collected about the ease to perform the task using the app.

#	Task
1	Find the provided information using all the tools in the App
2	Find the provided information using the “Artistic” and “Pencil” styles
3	Find the provided information using the “Edge Enhancement” and “CNN” style
4	Infer information from the 3D models using all the tools in the App

Table 6.1. List of tasks in the evaluation test

The computing environment to carry out the four tasks is summarised in the next table (Table 6.2)

URL of tested website:	http://cintymd.github.io
Computer platform:	Fujitsu ESPRIMO P957/E90+
Browser tested:	Mozilla Firefox Explorer 68.0.2
Screen resolution	1920 X 1080
Operating system	Windows 10

Table 6.2. Computing environment for the evaluation

6.2. Results

6.2.1. Task 1 Gerhart-Potthoff-Bau

During the first task the participants were asked to find a particular entrance in the building (Figure 6-1) using all the tools available in the App. The building was described with a picture of its photorealistic 3D model, so the user could find the building using the initial setup.

Figure 6-1. Entrance number 9 Gerhart-Potthoff Building (https://navigator.tu-dresden.de).

According to the statistical results (Figure 6-2), 80% of the participants found the object in the building, the participants were asked to point at their chosen area, to verify their answer. Among the total of the participants 80% expressed that in a scale from Hard to Easy, the convenience of locating the area in the building was between *Easy* and *Okay*. From these participants 50% were not familiar with the campus or were not very confident navigating in the area.

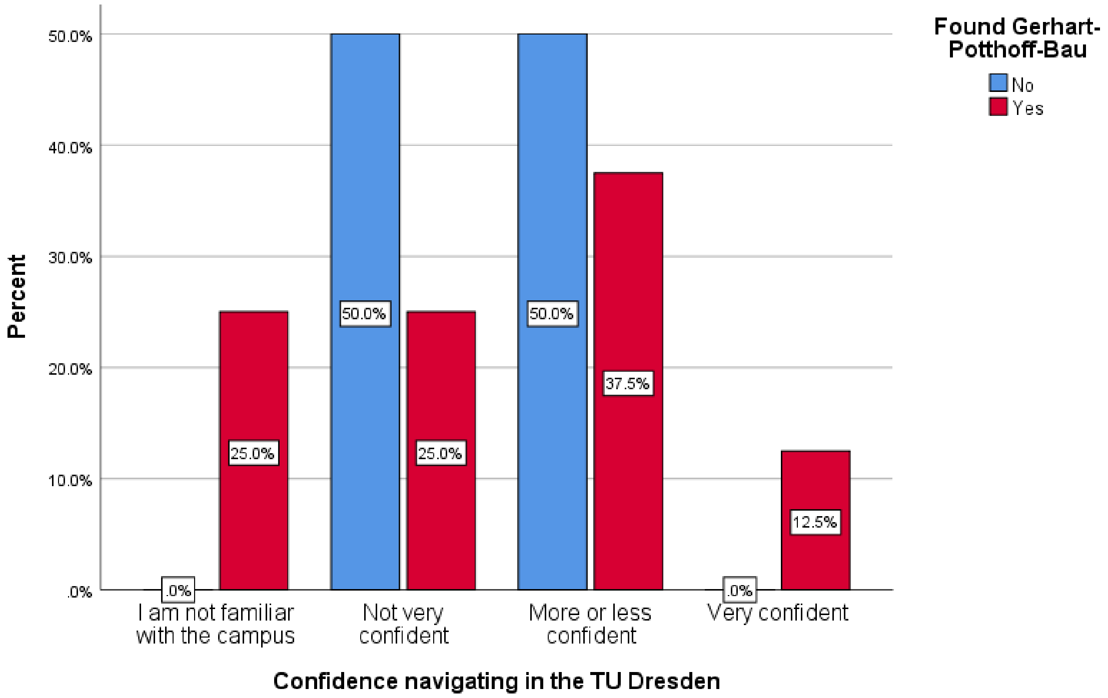


Figure 6-2. Level of confidence in navigating in the TU Dresden and correct the location of the area (Gerhart-Potthoff Building).

The participants were asked to change the different textures, once they have located the area, to verify if there was any other information in the different textures that might help them to be sure about their answer. Considering that some occluded areas would be better perceived in sketch models, according to the results 80% said that there was indeed additional information to be sure about their answer. Furthermore, the participants mentioned that the most helpful textures were *photorealistic* (30%), *artistic* style (20%) *pencil-based* style (20%), *edge enhanced* (20%) and *NST* (10%), this means, that 70% of the participants chose a NPR style as the most helpful texture.

6.2.2. Task 2 Mollier-Bau

The second task bears a close resemblance to the first task, as well as the previous task, the participants have to identify an area (Figure 6-4) in the building Mollier-Bau (Figure 6-3). Unlike the first one, the users are restricted in the texture aspects. The participants are allowed to use only two texture styles to carry out the task, "*Artistic*" and "*Pencil*".

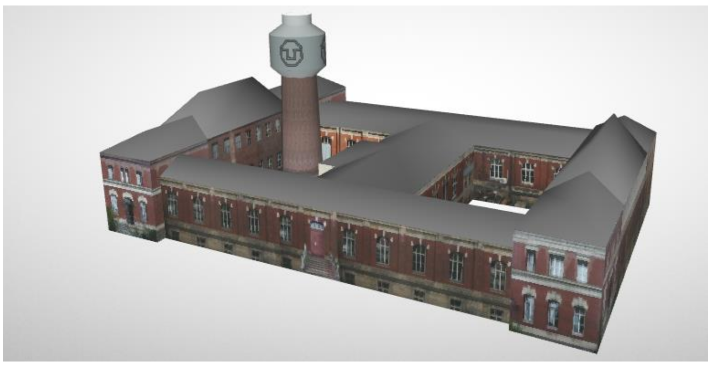


Figure 6-3. Entrance number 9 Gerhart-Potthoff Building.

Figure 6-4. Entrance number 5 Mollier Building (https://navigator.tu-dresden.de).

As it can be seen in Figure 6-5, from the total participants, 90% of them found *Easy* or *Okay* to locate the object, only 10% mentioned that was *hard* to identify the area. From the 90% of participants that reported *Easy* or *Okay*, 66.67% feel very confident about their answer (Figure 6-6).

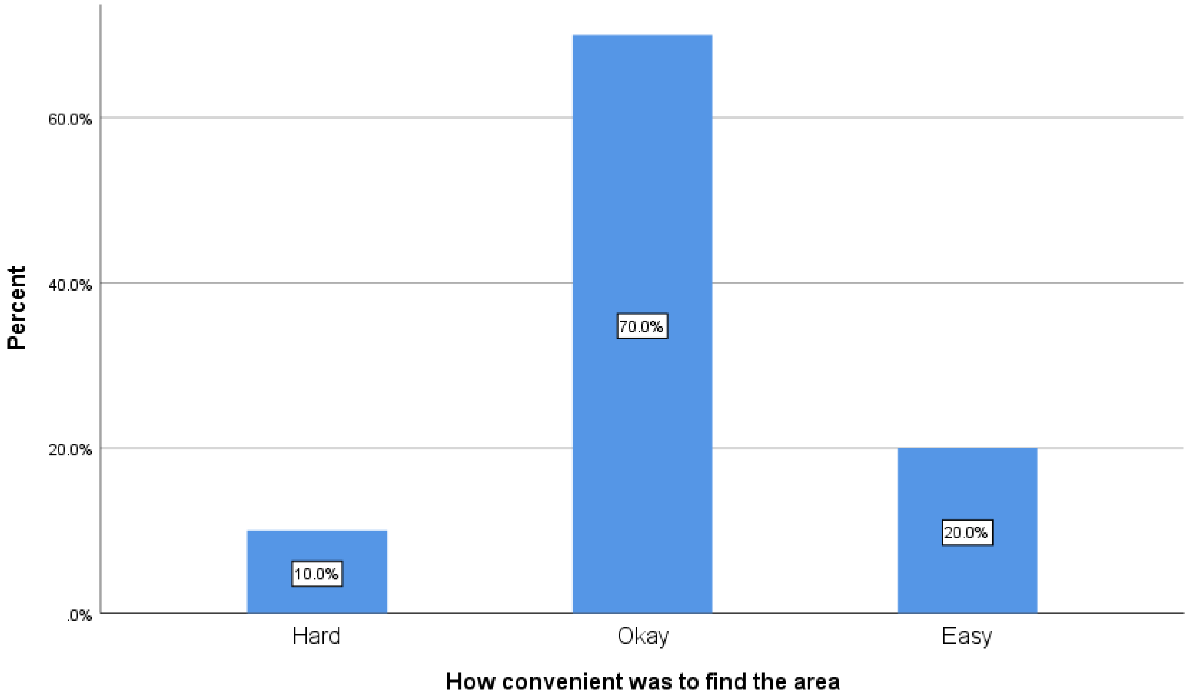


Figure 6-5. Level of convenience when locating the areas (Mollier Building).

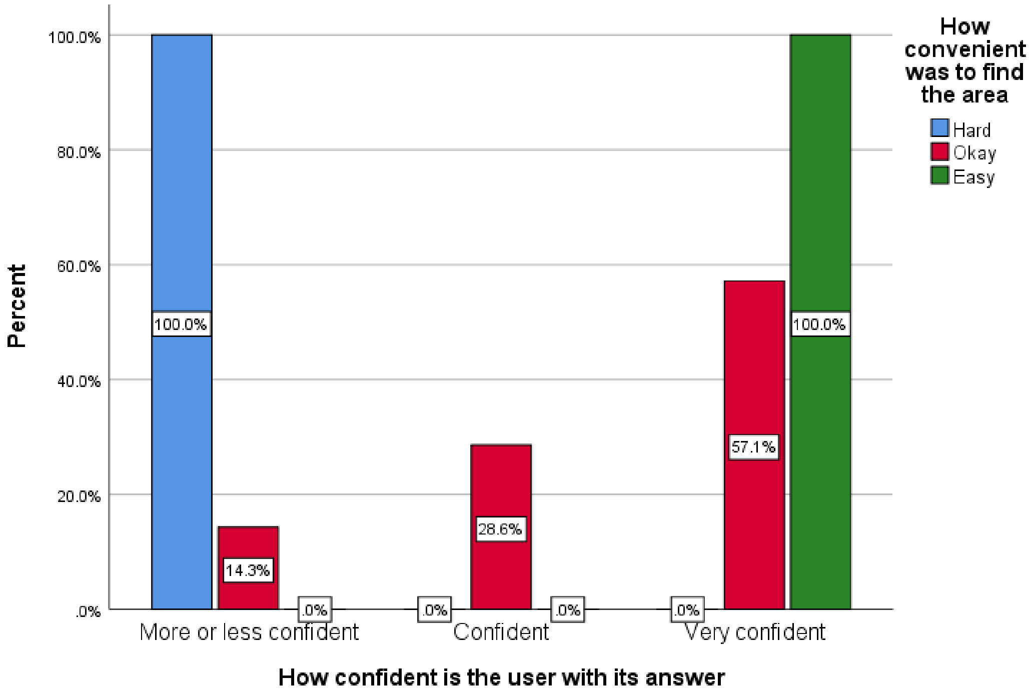


Figure 6-6. Level of confidence in the user's answer and the convenience when locating the area (Mollier Building).

6.2.3. Task 3 Barkhausen-Bau

On the third task the participants are restricted to use only the "*Edge Enhanced*" and "*CNN*" styles to look for the area required in the building Barkhausen-Bau (Figure 6-7). In this case both texture aspects belong to the sketch-based style, which might ease the look for the element in the scene.

Figure 6-7. Entrance number 14 Barkhausen Building (https://navigator.tu-dresden.de).

The resulting data estimates that 70% of the participants mentions that the convenience in finding such area is between *Easy* and *Okay* (Figure 6-8). Besides, 85% of these users feel very confident about their answer (Figure 6-9).

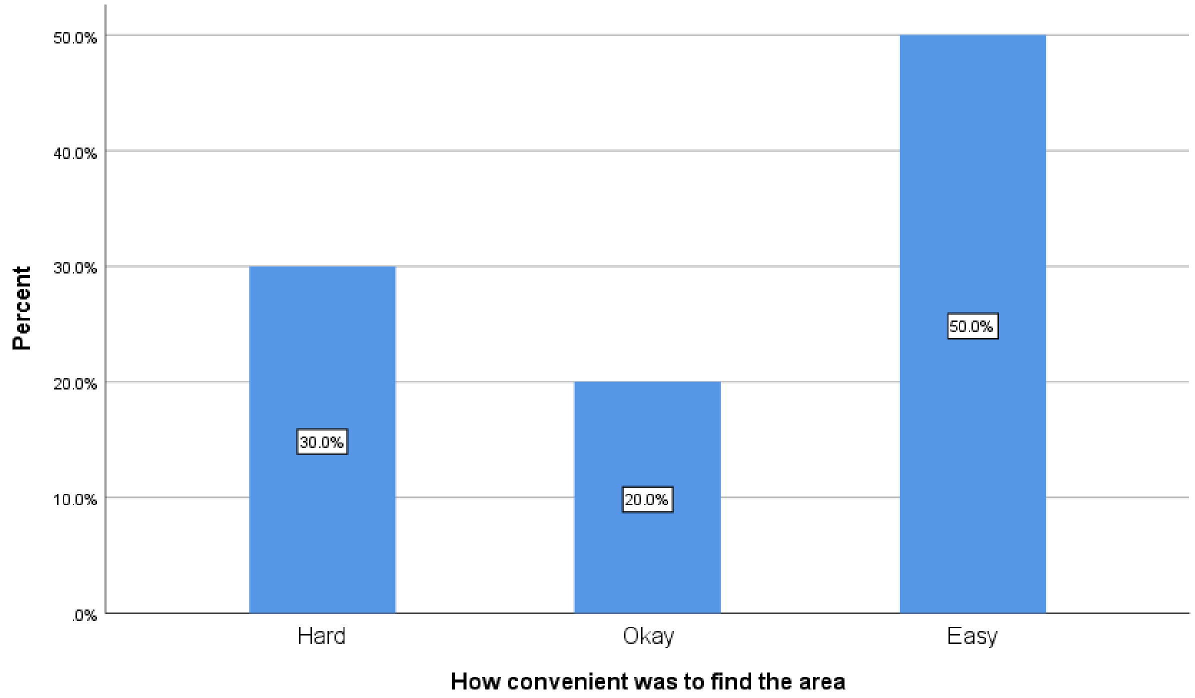


Figure 6-8. Level of convenience when locating the areas (Barkhausen Building).

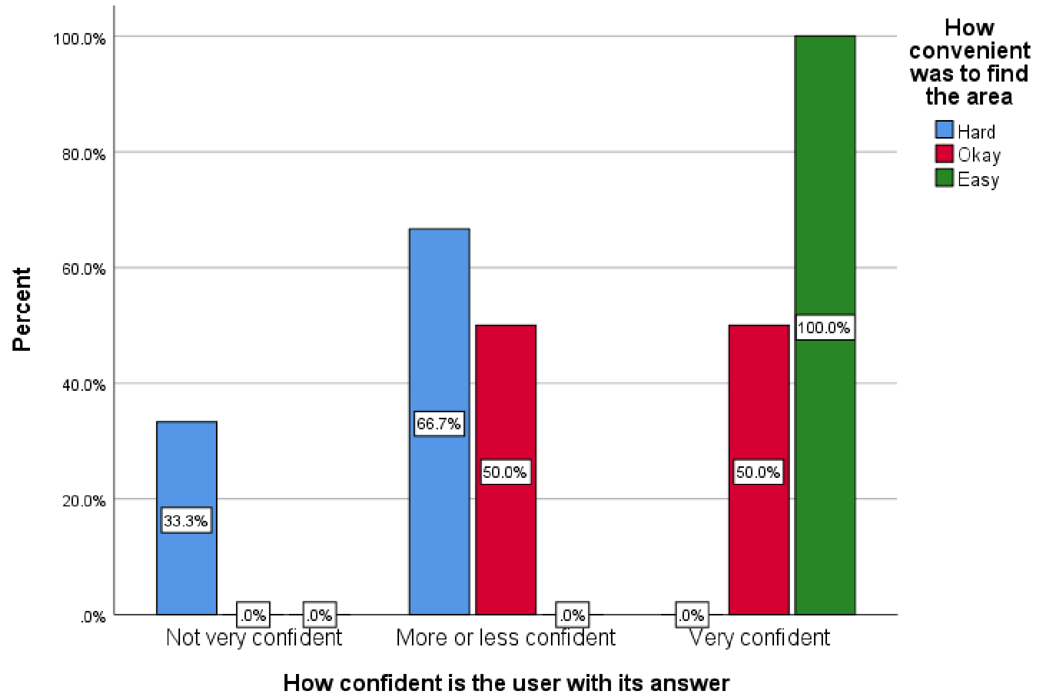


Figure 6-9. Level of confidence in the user's answer and the convenience when locating the area (Barkhausen Building).

6.2.4. Task 4 Görges-Bau

At this point of the evaluation, the participants are more familiar with the tools in the scene. For this reason, the last task is to infer internal information of the buildings (Figure 6-10) from the elements that exist in exterior. In order to carry out the task the participants are allowed to use all the texture aspects and additional tools in the menu.

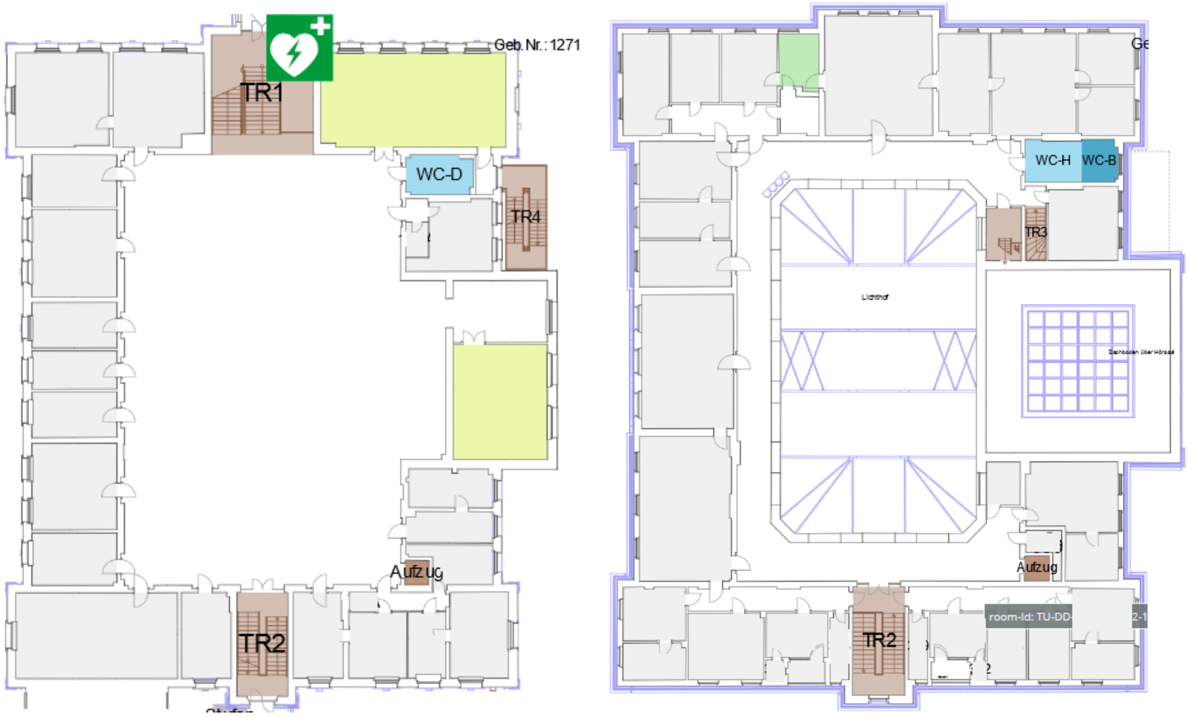


Figure 6-10. Ground floor and second floor, Görges Building's floor plans (https://navigator.tu-dresden.de).

According to the resulting data (Figure 6-11), only 30% of the users determined correctly both floors, whereas 60% determined only one of them.

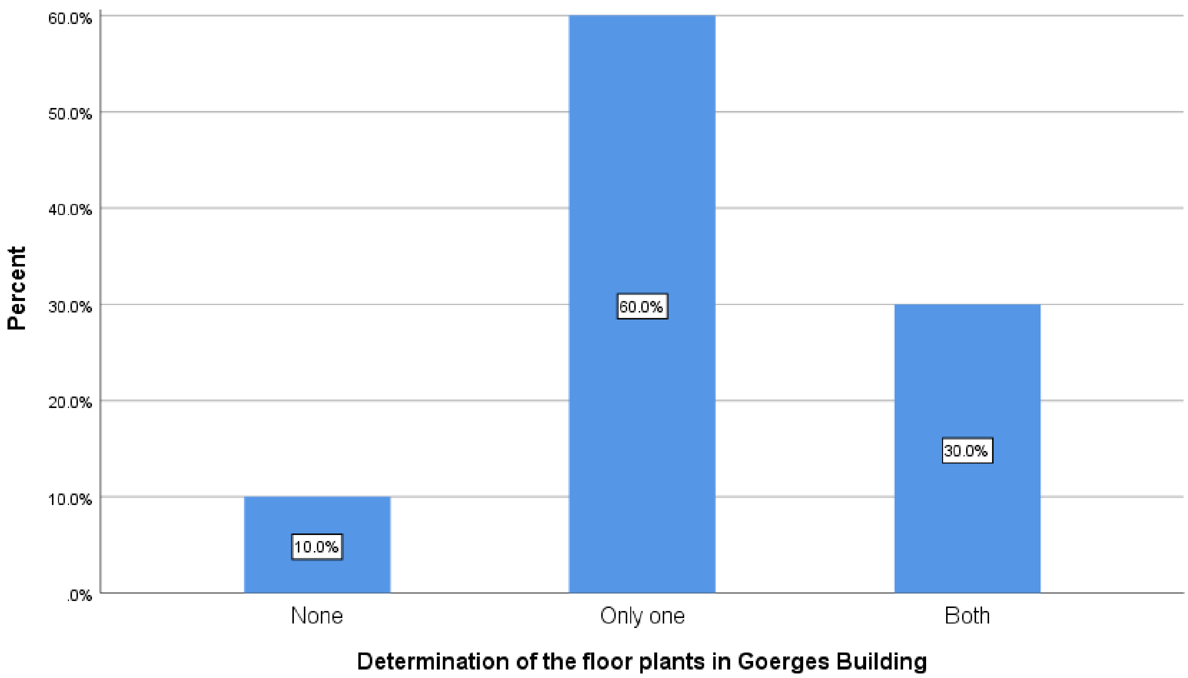


Figure 6-11. Determination of the floor plants, Görges Building.

Based on the perception of the participants (Figure 6-12), 60% mentions that the convenience in determining this information out of the scene was in a level from *Easy* to *Okay*. Although the duration in solving the tasks was not recorded, as for the time allocated by the participants, it existed an increased.

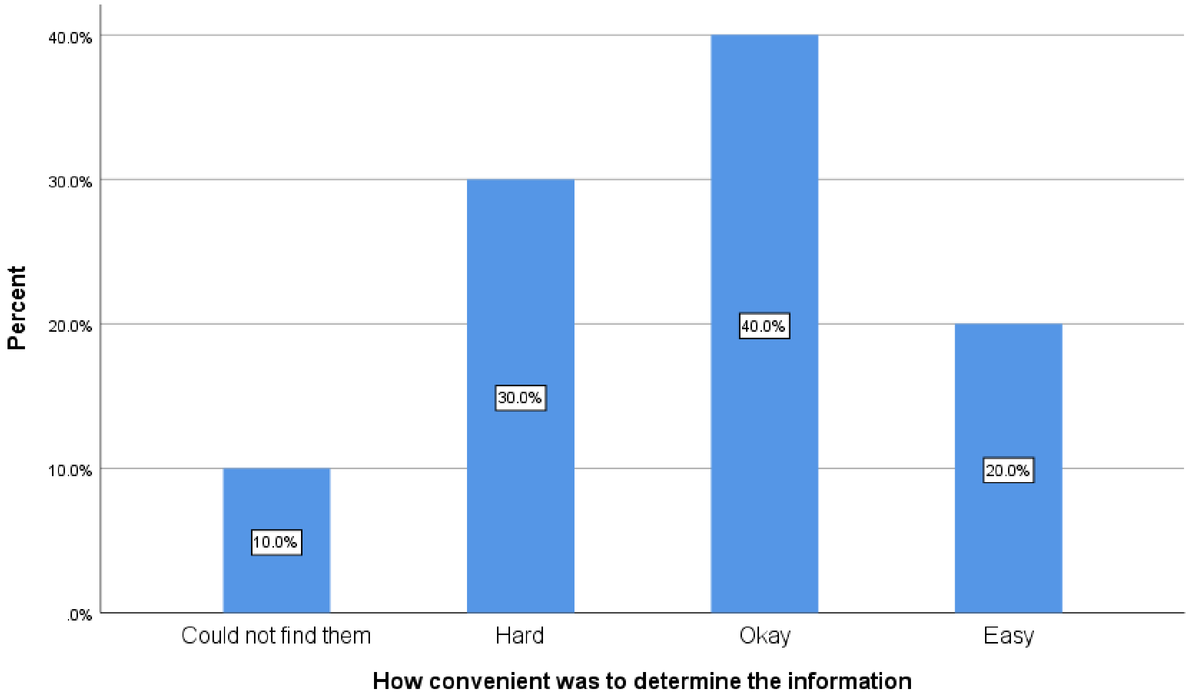


Figure 6-12. Level of convenience when locating the areas (Görges Building).

Additionally, out of the 60% of the participants that found *Easy* to *Okay*, in terms of how confident the participants are about their answers, only 30% feel very confident (Figure 6-13).

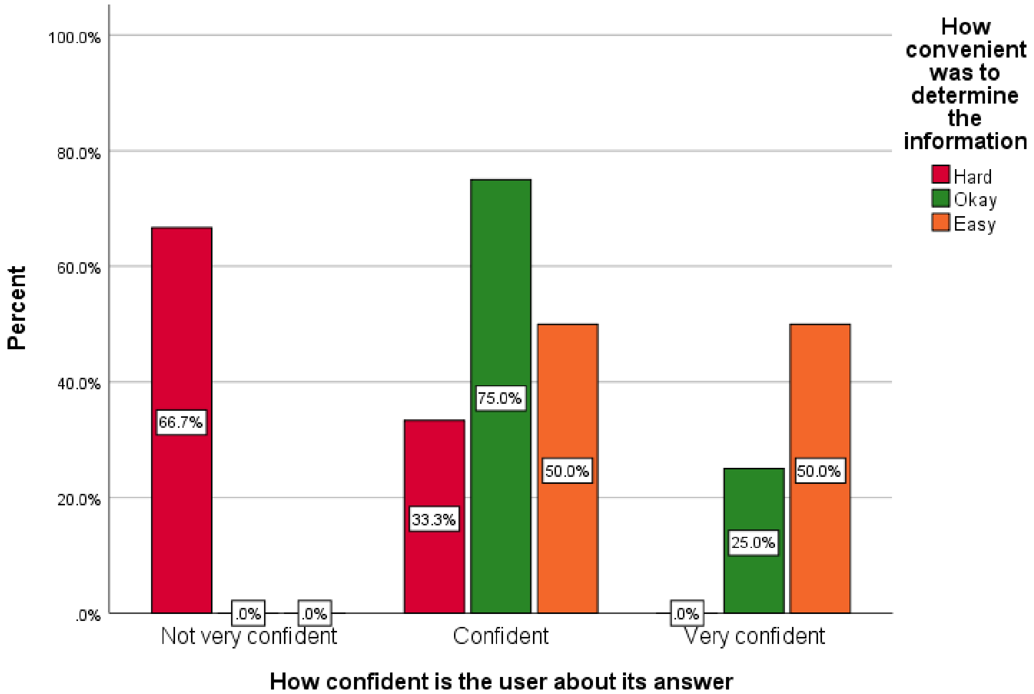


Figure 6-13. Level of confidence in the user's answer and the convenience when locating the area (Barkhausen Building).

6.2.5. User Impressions and suggestions

Once finished the four tasks, the participants were asked about their impression about the different texture aspect styles, in a scale from 1 to 5, in terms of:

1. Usefulness of the styles to carry out the tasks.
2. Difficulty to identify elements in the façades.
3. Communication of information in the façades.
4. Aesthetic design and expression of emotions.

The results reveal that the *photorealistic* view of the buildings is the most useful texture for the user, in order to carry out the tasks. By counting the perception of useful to very useful, the *edge enhanced* style is in second place, followed by the *pencil-based* style and *NST* style. As it can be seen in the Figure 6-14, the least useful texture aspect for the participants was the *artistic* style.

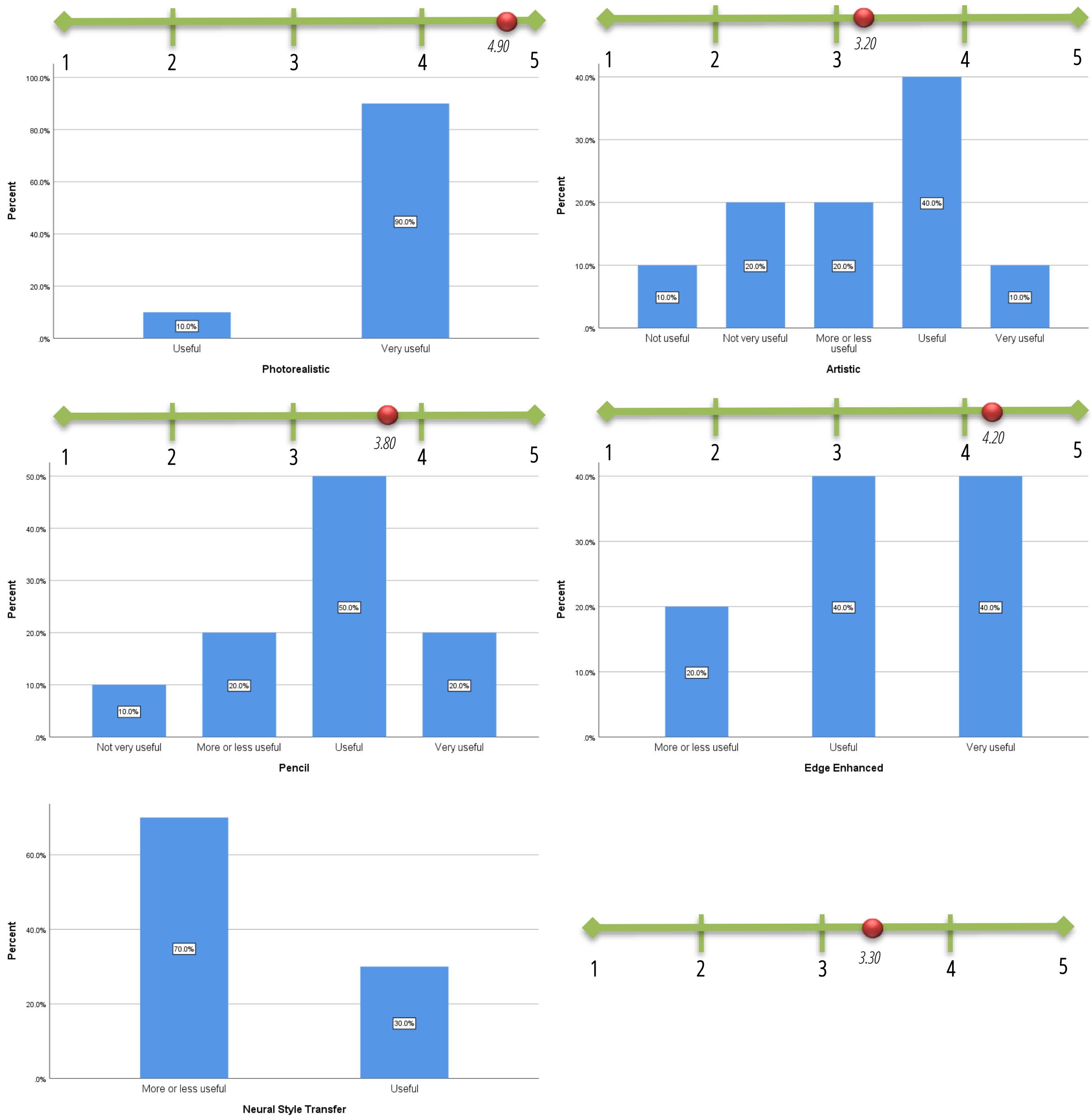


Figure 6-14. User's perception about the usefulness of the texture aspect styles to carry out the tasks.

The performance to solve each of the tasks is also affected by the ease to perceive the element in each of the styles. Therefore, the participants were required to rate each of the aspect styles, according to the difficulty to perceive the structures in the façades (Figure 6-15). Where *photorealistic* and *edge enhanced* styles were once more the highest rated, followed by *NST* and *pencil-based* style.

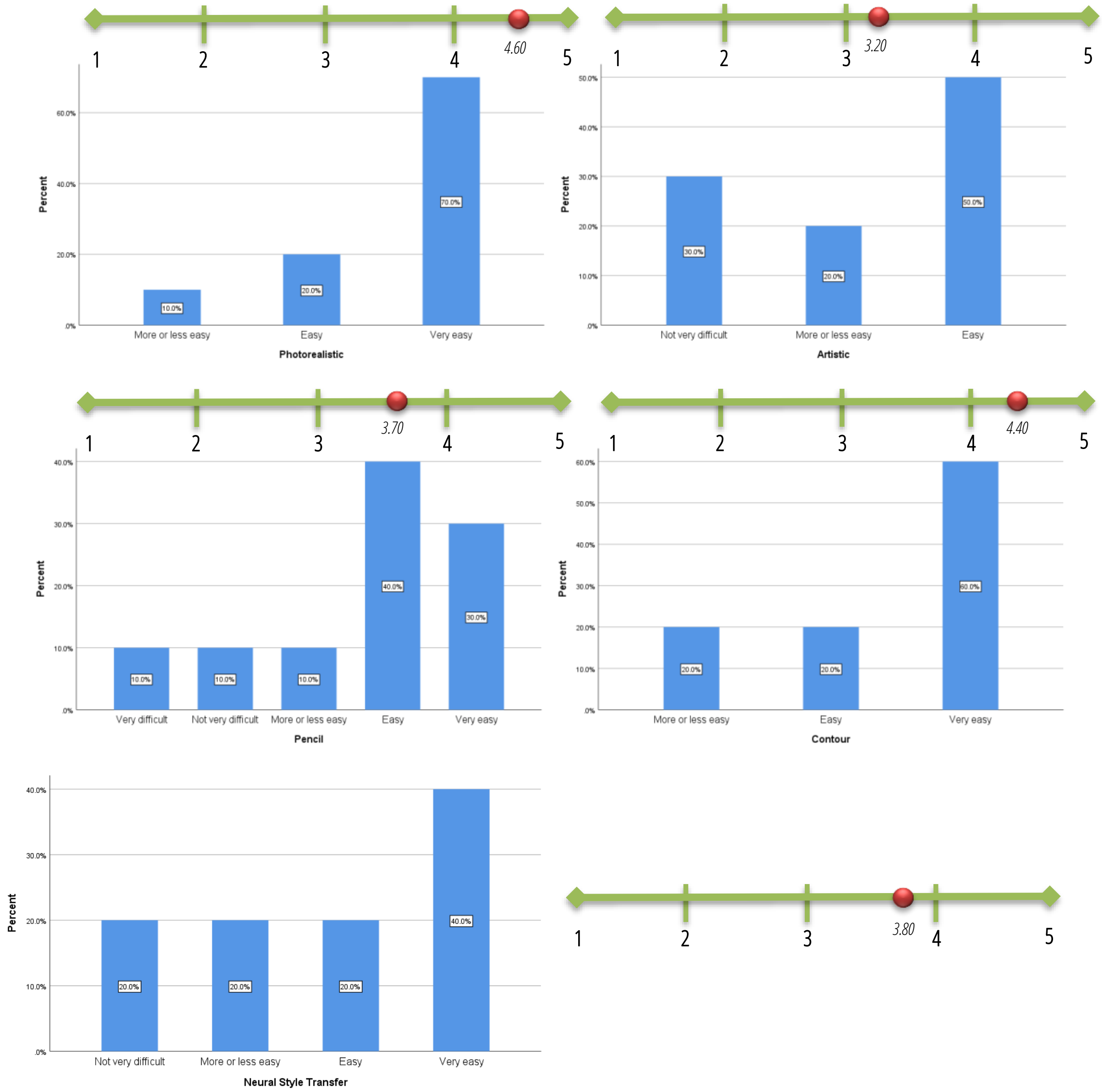


Figure 6-15. User's perception about the difficulty to perceive elements in façades in each of the texture aspect styles.

Although the ease to perceive an element is fundamental, it is also essential to understand which object is being communicated from the buildings (Figure 6-16), in order to be able to identify areas and navigate confidently through the scene. Hence, based on the answers of the participants, it is accurate to mention that the *photorealistic* and *edge enhanced* styles, both communicate well the elements and it is easy to perceive them, whereas, the *artistic* style, which is the lowest rated in the both categories, presents the most difficulty and the poorest communication. The opinion about the *pencil* based and *NST* is fairly neutral for the participants.

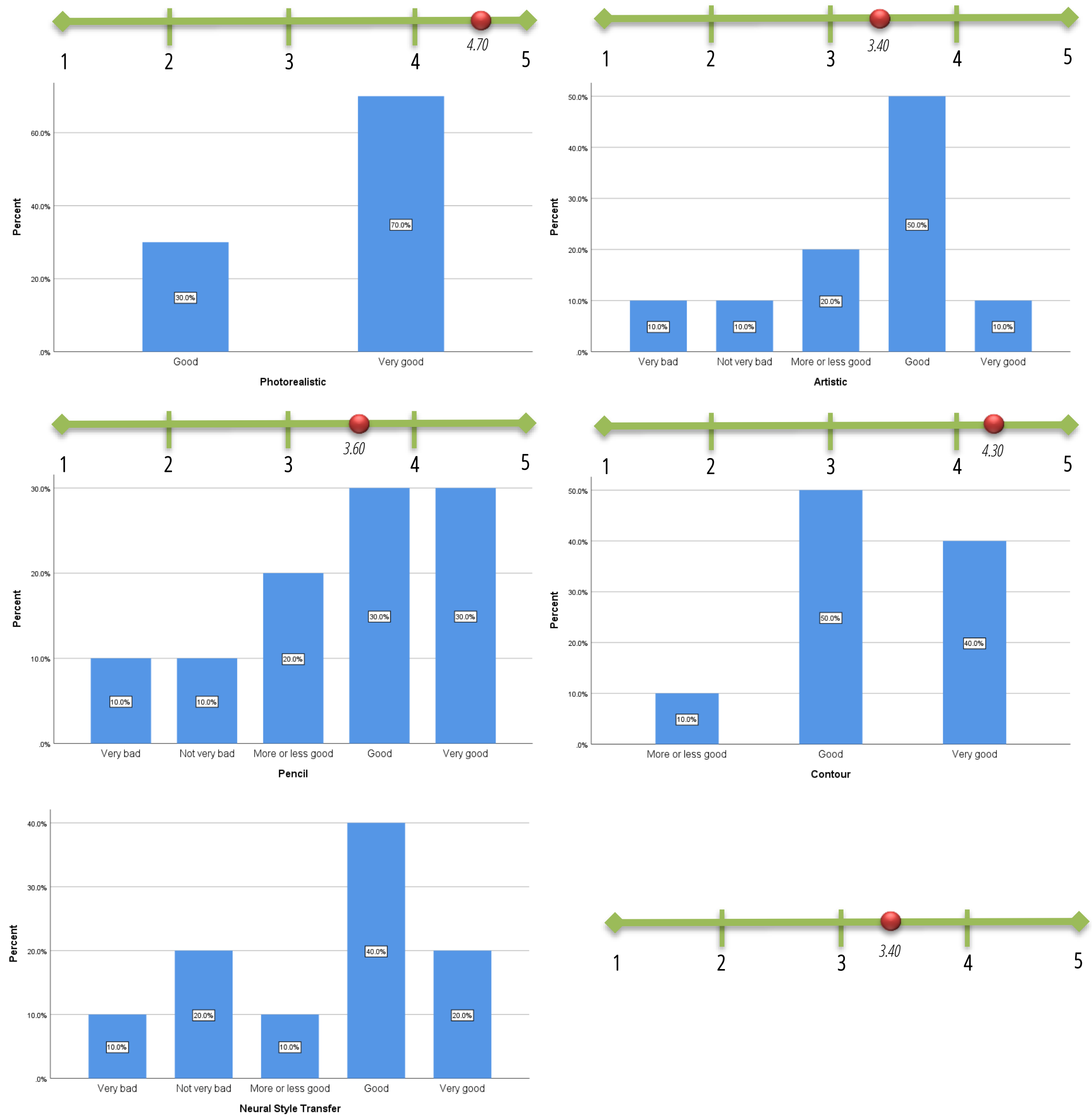


Figure 6-16. User's perception about communication of the elements in façades in each of the texture aspect styles.

The final evaluation about the texture aspect styles, is based on the aesthetic design as an artistic work, which might arouse an emotion in the user, thus, the participants were required to order the NPR styles in a scale from 1 to 4, where 1 represents the least aesthetical style and 5 the most aesthetical (Figure 6-17). On average, the *NTS* technique was the highest ranked as the style that most communicate emotions and transmit a satisfying aesthetic design. In second places is the *pencil-based* technique, followed by the *artistic* style. Finally, the style that transmit the least to the user is the edge enhanced technique, despite its effectiveness to help the user to solve the tasks.

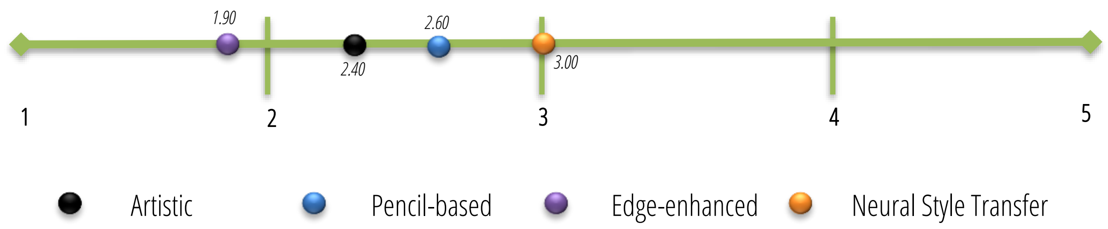


Figure 6-17. User's perception about communication of the elements in façades in each of the texture aspect styles.

After the use of the application, the participants were asked, if available, they would utilise a NPR map for navigation and as reported by them (Figure 6-18), 50% would probably use this tool for navigation. Besides, 90% of the participants considers that this tool might be useful for emergency assistance (Figure 6-19).

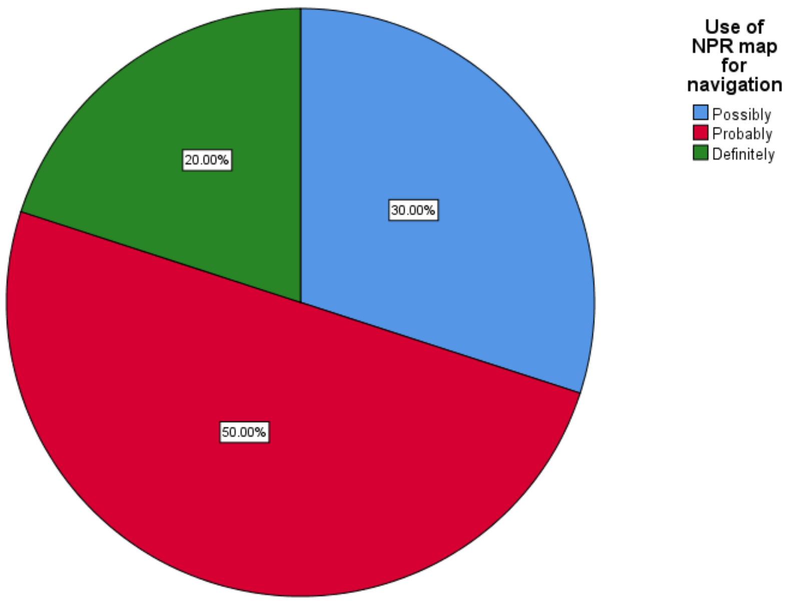


Figure 6-18. User's opinion in the use of NPR 3D model for navigation.

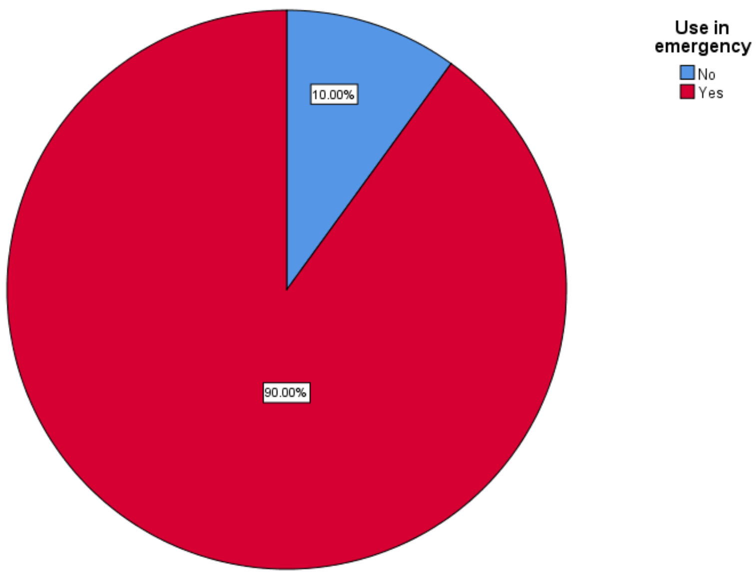


Figure 6-19. User's opinion in the use of NPR 3D model for emergency assistance.

Additional to the evaluation of the different styles in the textures of the façades, the questionnaire contained some evaluations to the supplementary tools and features in the scene. The first interactive tool to be evaluated was the "*DayTime*" mode. 80% of the users agrees that the object recognition might be affected by the illumination (Figure 6-20), depending on the time of the day, thus, 60% of them express that the tool had an impact in the recognition of the areas (Figure 6-21).

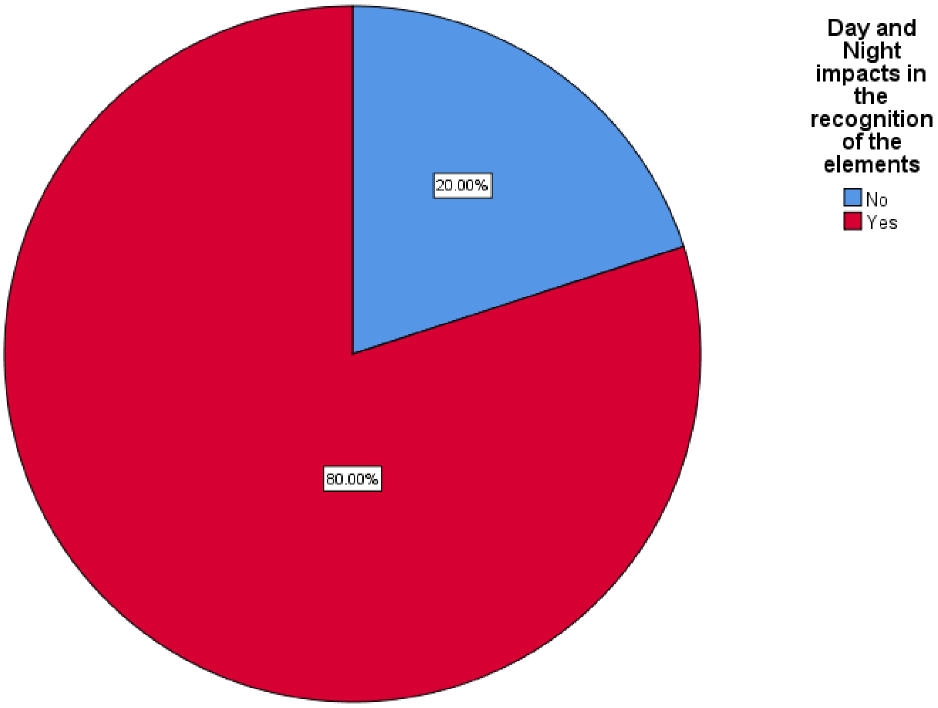


Figure 6-20. User's opinion about the impact of the illumination of the time of day to recognise objects in the scene.

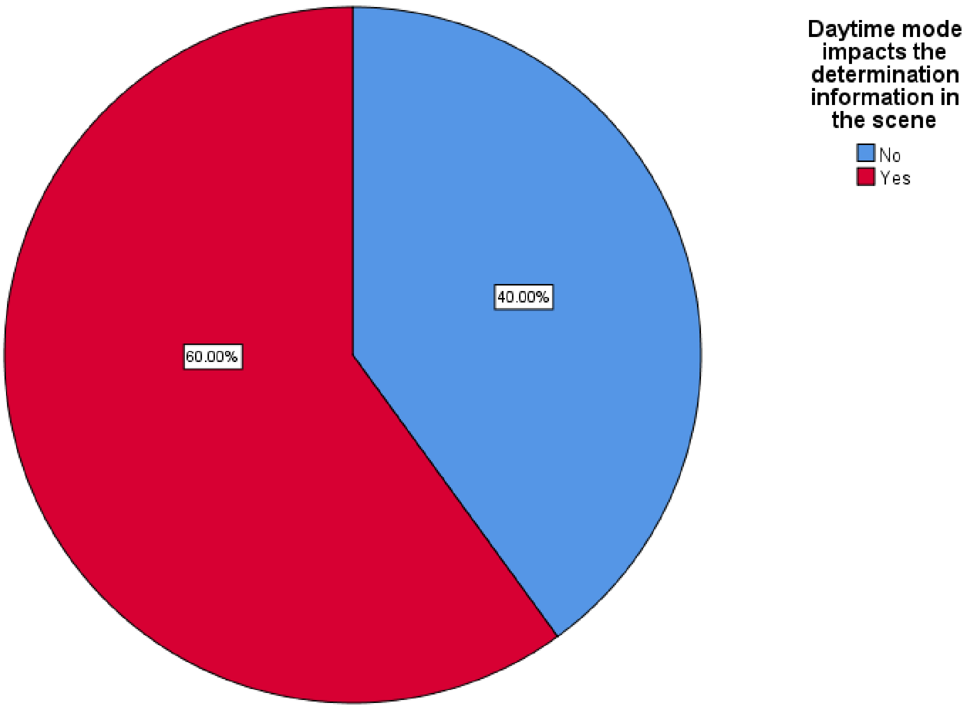


Figure 6-21. User's opinion the impact of the *DayTime* mode in the App.

Subsequently, the additional features in the scene like the trees and streetlamps were graded in their influence on the location of objects or areas in the scene. The estimation says that 70% of the participants declared that these objects have an impact in the scene (Figure 6-22).

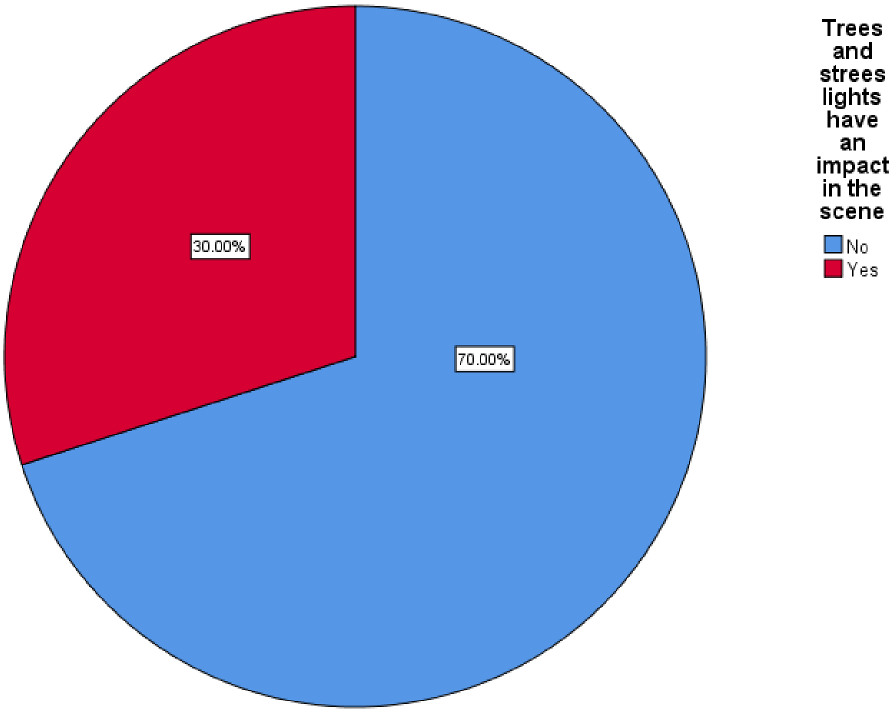


Figure 6-22. User's opinion the impact of the *Trees* and *Streetlamps* in the scene.

Among the comments and feedbacks from the participants, they consider that the tool wireframe might be useful in combination with floor plans, also for the communication of the heights of the buildings and the selection of the complete structure by displaying the silhouette. They consider that this tool might give a support in the urban planning of the area or the building by estimating sizes or volumes, as well as the planning of green areas.

In terms of additional features or information, some of the participants commented the inclusion of bicycle stands, benches, evacuation routes, security zones, emergency exits, the name and information of the buildings and parking spots. This opens the discussion in the further implementation of NPR, as this tool was well accepted by the participants. Nevertheless, some users without a strong background in GIS presented some difficulties while solving the tasks.

7. Conclusion and Future work

This project reviewed a further scope in the use in NPR techniques, as are adaptive models in a 3D scene, with the support of the implementation of sketch style, directly applied in camera images. The evidences from this study points towards the idea of non-photorealistic 3D models as complementary information for the user who has the need of 3D implementations, whether for retrieval of information or leisure navigation.

We have devised a methodology which supports the creation of NPR based on Cesium, as a virtual globe, and although this is not a completely new approach, it might set a further basis in the development of non-photorealistic and/or adaptive townscape models. Additionally, we have provided further evidence in the understanding of the virtual environments, when combining the adaptive non-photorealistic models and the additional features in the scene, such as the trees, streetlamps and wireframe of the buildings. Likewise, we tried to identify meaningful interactions and to define a simple design for the user, which might ease the familiarisation with the web application.

The strong point of our work lies in the affirmation that an additional implementation of sketch models might ensure that 3D virtual models would provide for the user a better awareness and memorability of the real environment. Although the current study is based on a small sample of participants, the comments provided by them suggest a further development in the adaptive sketch models, which not only ends in the addition of more features to the scene, but also in the broadening of more design techniques that employ colour, completely or partially, to the different structures in the images. These new approaches might widen the comprehension in the scene for the users.

Finally, a number of important limitations need to be considered. First, the current study has only examined the operation of 3D models in the TUD area, which does not include all the buildings that belong to the university. Second, the scope of this study was limited in terms of cartographic-oriented design aspects and granting additional information for the user, because of the limited interactive tools and visualisation principles (e.g. filtering, mapping symbolisation, transparencies and colour schemes) in the UI. Third, the size of the participant's sample might give a in initial information in terms of simplicity and usefulness of the application for the users. However, with a small sample size, caution must be applied, as the findings might not be transferable to wider population that includes users with no background in GIS.

It would be interesting to assess the different perception of the user when applying more design techniques, developing the monochrome sketch approach from this work. Similarly, it would be thought-provoking to compare experiences of individuals within the different backgrounds that the final users might have in the use of 3D models; for this reason, future trials should assess the impact of cartographic design aspects in adaptive sketch 3D models. The information provided in this work can be used to develop new style techniques and other types of formats can be tested, in order to implement a more efficient render of the scene.

References

Abdalla, M. M. A. (2017). *Concept and Implementation of a Contextualized Navigable 3D Landscape Model: The Uch Enmek Example (Altai Republic, Siberia).* Technical University Dresden.

AlMeraj, Z., Wyvill, B., Isenberg, T., Gooch, A. A., & Guy, R. (2009). Automatically mimicking unique hand-drawn pencil lines. *Computers & Graphics*, *33*(4), 496–508. https://doi.org/https://doi.org/10.1016/j.cag.2009.04.004

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., Çöltekin, A., Biljecki, F., … Çöltekin, A. (2015). Applications of 3D City Models: State of the Art Review. *ISPRS International Journal of Geo-Information*, *4*(4), 2842–2889. https://doi.org/10.3390/ijgi4042842

Buchroithner, M., & Knust, C. (2013). The third dimension in cartography: recent and future developments. *Kartographische Nachrichten*, *63*(3), 127–132.

Chaturvedi, K., Yao, Z., & Kolbe, T. H. (2015). Web-based Exploration of and interaction with large and deeply structured semantic 3D city models using HTML5 and WebGL. In *Bridging Scales-Skalenübergreifende Nah-und Fernerkundungsmethoden, 35. Wissenschaftlich-Technische Jahrestagung der DGPF*.

DeCarlo, D., & Santella, A. (2002). Stylization and abstraction of photographs. In *ACM transactions on graphics (TOG)* (Vol. 21, pp. 769–776).

Döllner, J., Kolbe, T. H., Liecke, F., Sgouros, T., & Teichmann, K. (2006). The Virtual 3D City Model of Berlin-Managing, Integrating, and Communicating Complex Urban Information. In *Proceedings of the 25th International Symposium on Urban Data Management UDMS 2006 in Aalborg, Denmark, 15-17 May 2006*. Retrieved from http://www.citygml.org/fileadmin/citygml/docs/udms_berlin3d_2006.pdf;

El Haje, N., Jessel, J.-P., Gaildrat, V., & Sanza, C. (2016). 3D Cities Rendering and Visualisation: A Web-Based Solution. In *Eurographics Workshop on Urban Data Modelling and Visualisation (UDMV 2016)* (pp. 95–100). Liege, Belgium. Retrieved from https://hal.archives-ouvertes.fr/hal-01787406

Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). A neural algorithm of artistic style. *ArXiv Preprint ArXiv:1508.06576*.

Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2414–2423).

Gooch, B., & Gooch, A. (2001). *Non-photorealistic rendering*. AK Peters/CRC Press.

Herold, Martin; Liu, XiaoHang; Clarke, K. C. (2003). Spatial Metrics and Image Texture for Mapping Urban Land Use. *Photogrammetric Engineering & Remote Sensing*, *69*(9), 991–1001. https://doi.org/doi:10.14358/PERS.69.9.991

Jahnke, M. (2013). *Nicht-Photorealismus in der Stadtmodellvisualisierung für mobile Nutzungskontexte*. Technische Universität München, München.

Jahnke, M., Fan, H., & Meng, L. (2009). A user perspective on typification of façade features of 3D building models. In *Proceedings of the 24th International Cartographic Conference (ICC), Santiago, Chile* (pp. 15–22).

Khronos Group. (n.d.). COLLADA. Retrieved from https://www.khronos.org/collada/

Khronos Group. (2015). glTF Introduction 'OpenGL Transmission Format.' Retrieved from https://www.khronos.org/gltf/

Klein, A. W., Li, W., Kazhdan, M. M., Corrêa, W. T., Finkelstein, A., & Funkhouser, T. A. (2000). Non-photorealistic virtual environments. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (pp. 527–534).

Lee, H., Kwon, S., & Lee, S. (2006). Real-time Pencil Rendering. In *Proceedings of the 4th International Symposium on Non-photorealistic Animation and Rendering* (pp. 37–45). New York, NY, USA: ACM. https://doi.org/10.1145/1124728.1124735

Lin, F., Chang, W.-Y., Tsai, W.-F., & Shih, C.-C. (2017). Development of 3D Earth Visualization for Taiwan Ocean Environment Demonstration. In Y. Tan, H. Takagi, & Y. Shi (Eds.), *Data Mining and Big Data* (pp. 307–313). Cham: Springer International Publishing.

Liu, Y.-J., Luo, X., Joneja, A., Ma, C.-X., Fu, X.-L., & Song, D. (2013). User-adaptive sketch-based 3-D CAD model retrieval. *IEEE Transactions on Automation Science and Engineering*, *10*(3), 783–795.

Moradi, D. (2017). *3D Thematic Mapping and Visualization in CesiumJS Virtual Globe*. Technische Universität München.

Murshed, S. M., Al-Hyari, A. M., Wendel, J., & Ansart, L. (2018). Design and Implementation of a 4D Web Application for Analytical Visualization of Smart City Applications. *ISPRS International Journal of Geo-Information*, *7*(7). https://doi.org/10.3390/ijgi7070276

Nishida, G., Garcia-Dorado, I., Aliaga, D. G., Benes, B., & Bousseau, A. (2016). Interactive sketching of urban procedural models. *ACM Transactions on Graphics (TOG)*, *35*(4), 130.

Orlik, A., & Orlikova, L. (2014). Current trends in formats and coordinate transformations of geospatial data --- based on MyGeoData Converter. *Central European Journal of Geosciences*, *6*(3), 354–362. https://doi.org/10.2478/s13533-012-0178-8

Prechtel, N. (2015). On strategies and automation in upgrading 2D to 3D landscape representations. *Cartography and Geographic Information Science*, *42*(3), 244–258.

Prechtel, N. (2016). Anmerkungen zu Gestaltungsoptionen in Digitalen 3d-Landschaftsmodellen. *Vom Gelände Zur Karte...*, *40*, 75.

Ross, D.-I. L. (2010). Virtual 3d city models in urban land management. *Technischen Universitat, Berlin*.

Roth, R. E., Çöltekin, A., Delazari, L., Filho, H. F., Griffin, A., Hall, A., … van Elzakker, C. P. J. M. (2017). User studies in cartography: opportunities for empirical research on interactive maps and visualizations. *International Journal of Cartography*, *3*(sup1), 61–89. https://doi.org/10.1080/23729333.2017.1288534

Roth, R. E., Ross, K. S., & MacEachren, A. M. (2015). User-Centered Design for Interactive Maps: A Case Study in Crime Analysis. *ISPRS International Journal of Geo-Information*, *4*(1), 262–301.

https://doi.org/10.3390/ijgi4010262

Semmo, A. (2016). *Design and implementation of non-photorealistic rendering techniques for 3D geospatial data*. Universität Potsdam. Retrieved from https://publishup.uni-potsdam.de/opus4-ubp/frontdoor/index/index/docId/9952

Semmo, A., & Döllner, J. (2015). Interactive image filtering for level-of-abstraction texturing of virtual 3D scenes. *Computers & Graphics*, *52*, 181–198. https://doi.org/10.1016/J.CAG.2015.02.001

van Elzakker, C. P. J. M., & Delikostidis, I. (2010). Use, user, and usability research for mobile geo-applications for personal orientation and navigation. In *2010 IEEE International Professional Comunication Conference* (pp. 121–130). https://doi.org/10.1109/IPCC.2010.5529824

Wang, J., Bao, H., Zhou, W., Peng, Q., & Yingqing, X. (2002). Automatic image-based pencil sketch rendering. *Journal of Computer Science and Technology*, *17*(3), 347–355. https://doi.org/10.1007/BF02947313

Appendix A. Evaluation User Test Survey

Welcome to this survey!

Thank you for agreeing to take part in this survey. This question-based survey has been conducted for evaluating the effectiveness and aesthetic quality of townscape models. The present model uses non-photorealistic rendering (NPR) techniques giving the 3D surfaces a sketch-like appearance. This survey will not take you more than 30 min. All information provided will be strictly confidential and has a single purpose, the test and eventual improvement of the Web design I have created. No personal information will be published. Only statistical figures will be deduced from the answers.

1. What is your gender?

☐ Male ☐ Female

2. How old are you?

☐ Under 20 ☐ 20-23 ☐ 24-27 ☐ 28-31 ☐ 32-35 ☐ Above 35

3. Select your highest education level

☐ High school/less ☐ Bachelors ☐ Masters ☐ PhD ☐ Higher

4. Please select the category that best describes your profession:

☐ Student ☐ Graduated ☐ Employee ☐ Retired ☐ Unemployed ☐ Other (specify):

____

5. Are you familiar with 3D city models?

☐ Yes ☐ No

6. Have you ever used a 3D map?

☐ Yes ☐ No

7. If "YES" please tell us for which purpose?

☐ Virtual tour ☐ Navigation ☐ Analysis ☐ Other (specify):

____

8. How confident do you feel in the navigation inside the campus of the TU Dresden?

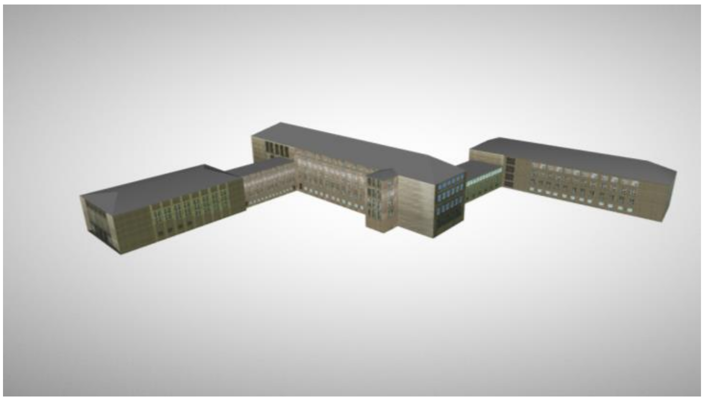
Very confident	Confident	More or less confident	Not very confident	Not confident	I am not familiar with the campus

Now I will give you some minutes to get familiarised with the application…

The next step in this survey is to identify some areas of interest in the buildings.

Task 1

Here is a picture of the **Gerhart-Potthoff-Bau** building representing the photorealistic 3D model:



1. Can you identify the next entrance in the building? (*Feel free to use all the tools contained in the app to carry out the task*).

10. Once you found it, look at the facade using the different textures. Is there extra information that is helpful for you to be sure about your answer?

☐ Yes ☐ No

11. Which texture was the most helpful?

____

12. How convenient was for you to find this entrance in the building?

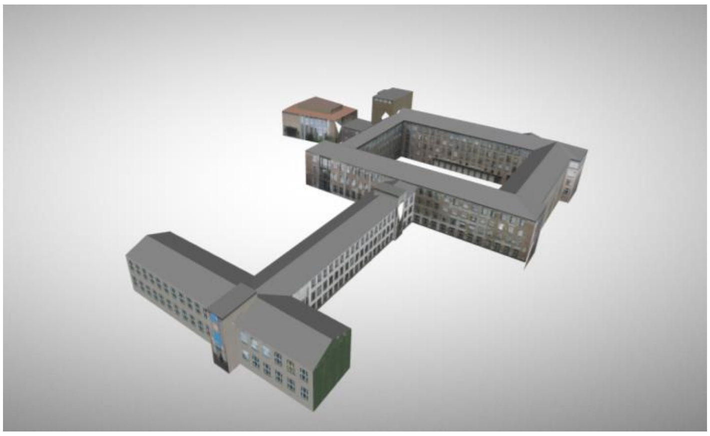
☐ Easy ☐ Okay ☐ Hard ☐ Couldn't find it

13. In case that you find the area, how confident you feel about your answer?

Very confident	Confident	More or less confident	Not very confident

Task 3

Here is a picture of the **Barkhausen-Bau** building representing the photorealistic 3D model:



12. Can you identify the next entrance in the building? Use the "Contour" and "CNN" to carry out the task.

13. How convenient was for you to find this entrance in the building?

☐ Easy ☐ Okay ☐ Hard ☐ Couldn't find it

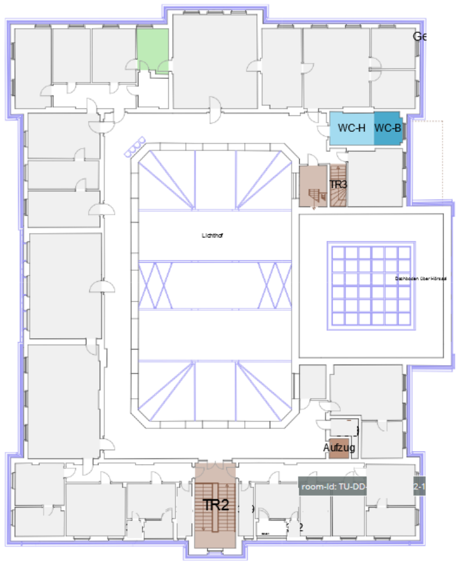
14. In case that you find the area, how confident you feel about your answer?

Very confident	Confident	More or less confident	Not very confident

Task 4

In the 3D model application, go to the **Görges-Bau** building. The location is marked in the next map:

12. Can you identify the floor that corresponds to these floor plans? Feel free to use all the tools contained in the app to carry out the task.



13. How convenient was for you to determine the floor?

☐ Easy ☐ Okay ☐ Hard ☐ Couldn't find them

14. In case that you find the area, how confident you feel about your answer?

Very confident	Confident	More or less confident	Not very confident

12. List the structures or information in the building that helped you to determine the floor:

Now I will ask you to grade the different textures inside the app…

----

13. In a scale from 1 to 5, where **1** is **NOT** useful at all and **5 VERY** useful, grade the textures:

Photorealistic (None)	5	4	3	2	1
Artistic	5	4	3	2	1
Pencil	5	4	3	2	1
Contour enhance	5	4	3	2	1
Convolutional Neural Network (CNN)	5	4	3	2	1

14. In your opinion, how is easy is to identify structural objects in each texture, in a scale from 1 to 5, where **1** is very **DIFFICULT** at all and **5** very **EASY**:

Photorealistic (None)	5	4	3	2	1
Artistic	5	4	3	2	1

Pencil	5	4	3	2	1
Contour enhance	5	4	3	2	1
Convolutional Neural Network (CNN)	5	4	3	2	1

12. In your opinion, how well the textures communicate the information from the facades, in a scale from 1 to 5, where **1** is **VERY BAD** at all and **5 VERY GOOD**:

Photorealistic (None)	5	4	3	2	1
Artistic	5	4	3	2	1
Pencil	5	4	3	2	1
Contour enhance	5	4	3	2	1
Convolutional Neural Network (CNN)	5	4	3	2	1

13. All work represented in an artistic style, should express and transmit emotions to the customer. In this terms, order from 1 to 4 the textures that use NP techniques, where 1 is the most expressive texture and 5 is the least.

1	
2	
3	
4	

12. Will you (if available as an option) use a NPR map for navigation?

☐ Definitely ☐ Probably ☐ Possibly ☐ Probably not ☐ Definitely not

13. In case of emergency, would you find useful the use of NPR visual aspects, for the emergency assistance to locate the entrances and important structures?

☐ Yes ☐ No

Now I will evaluate your opinion about the additional tools.

----

14. Do you think the recognition of objects change in different times during the day (e.g. at night)?

☐ Yes ☐ No

15. Based on your previous answer, does the "Daytime" tool have an impact in your identification of structures?

☐ Yes ☐ No

16. In which cases would you use the "Wireframe" tool?

17. In your opinion, does the external objects like Trees and Street lights have an impact in the recognition of the area?

☐ Yes ☐ No

18. Would you add other elements to the scene?

☐ Yes ☐ No

19. In case of "YES", which other objects would you add?

Appendix B. Cesiumjs Source Code (HTML, CSS and JavaScript syntax)

Index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <script src="https://cesiumjs.org/releases/1.57/Build/Cesium/Cesium.js"></script>
  <link href="https://cesiumjs.org/releases/1.57/Build/Cesium/Widgets/widgets.css" rel="stylesheet">
  <link rel="stylesheet" href="index.css" media="screen">

  <link href="https://fonts.googleapis.com/css?family=Open+Sans&display=swap" rel="stylesheet">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js"></script>
  <script src="https://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.4.0.min.js"></script>
  <script type="text/javascript" src="http://maps.stamen.com/js/tile.stamen.js?v1.3.0"></script>
</head>

<body>

  <div id="cesiumContainer"></div>

  <!-- Some input elements for our app. -->
  <div class="sidenav" id="menu">
    <h1 style="cursor:pointer" onclick="openNav()">&#9776; menu</h1>
  </div>

  <div class="backdrop" id="toolbar">
    <table>
      <tbody data-bind="foreach: layers">

        <a href="javascript:void(0)" class="closebtn" onclick="closeNav()">&times;</a>

        <img class="TUD" src="./Source/Images/TU_Dresden_Logo_weiss.png" alt="TUDresden" width="150"
height="40">
        <h2>Adaptive Textures</h2>
        <p><strong>Daytime <br></strong></p>

        <div class="nowrap">
          <label for="dayMode">Day
            <input id="dayMode" name="source" type="radio" checked />
```

```
        <span class="radious"></span>
    </label>

</div>
<div class="nowrap">
    <label for="nightMode">Night
        <input id="nightMode" name="source" type="radio" />
        <span class="radious"></span>
    </label>
</div>
<br>

<p><strong>Additional Features</strong></p>
<div class="nowrap">

    <label for="trees">Trees
        <input id="trees" type="checkbox" checked />
        <span class="checkmark"></span>
    </label>
</div>

<div class="nowrap">

    <label for="lights">Street Lights
        <input id="lights" type="checkbox" checked />
        <span class="checkmark"></span>
    </label>
</div>
<div class="nowrap">

    <label for="shadows">Shadows
        <input id="shadows" type="checkbox" />
        <span class="checkmark"></span>
    </label>
</div>
<div class="nowrap">

    <label for="wireframe">Wireframe
        <input id="wireframe" type="checkbox" />
```

```
                <span class="checkmark"></span>
              </label>
            </div>
            <br>
            <p><strong>Textures</strong></p>
            <td class="nowrap">
              <select data-bind="visible: $parent.isSelectableLayer($data), options: $parent.baseLayers, optionsText: 'name', value: $parent.selectedLayer"></select>
            </td>
          </tbody>
        </table>
  </div>

  <div class="controlPanel">
    <h5>Change the time: <span id="timeLabel"></span></h5>
    <div>
      <button type="button" class="cesium-button" id="hourMinus">-1 hour</button>
      <button type="button" class="cesium-button" id="hourPlus">+1 hour</button>
    </div>
  </div>

  <div id="loadingIndicator" class="cover">
    <div id="loadingIcon" class="loadingIndicator"></div>
  </div>

  <script src="Source/App.js"></script>

  <script>
    function openNav() {
      document.getElementById("toolbar").style.width = "250px";
    }

    function closeNav() {
      document.getElementById("toolbar").style.width = "0";
    }
  </script>
</body>
</html>
```

index.css

```
.backdrop {
  display: inline-block;
  border-radius: 5px;
  color: #fff;
  line-height: 150%;
  font-size: small;
  font-family: 'Open Sans', sans-serif;
  background: rgba(0, 38, 115, 1);
  border-radius: 4px;
  position: fixed;
  left: 0px;
  top: 0px;
  width: 0;
  z-index: 1;
  overflow-x: hidden;
  transition: 0.5s;

}

.TUD {
  padding: 15px 10px;
}

.sidenav {
  display: inline-block;
  background: rgba(42, 42, 42, 0.7);
  border-radius: 5px;
  border: 1px solid #444;
  padding: 5px 10px;
  color: #fff;
  line-height: 15%;
  font-family: 'Open Sans', sans-serif;
  position: absolute;
}

h1 {
  font-size: 17px;
}
```

```
/*Time control*/
.controlPanel {
  position: absolute;
  bottom: 40px;
  right: 10px;
  background: rgba(42, 42, 42, 0.8);
  color: #edffff;
  white-space: nowrap;
  font-size: 20px;
  padding: 4px 8px;
  border-radius: 4px;
}

.backdrop .closebtn {
  position: absolute;
  top: 0;
  right: 5px;
  font-size: 36px;
  margin-left: 50px;
  color: white;
  text-decoration: none;
}

.loadingIndicator {
  display: none;
  position: absolute;
  top: 50%;
  left: 50%;
  margin-top: -33px;
  margin-left: -33px;
  width: 66px;
  height: 66px;
  background: url(Source/Images/ajax-loader.gif) center no-repeat;
}

.cover {
  display: none;
  position: absolute;
```

```
  background-color: rgba(0, 0, 0, 0.75);
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
}

h2 {
  margin-bottom: 30px;
  padding-left: 10px;
}

p {
  font-size: 15px;
  padding-left: 10px;
}

select {
  width: 180px;
  font-size: 15px;
  cursor: pointer;
  font-family: 'Open Sans', sans-serif;
  margin-left: -10px;
  margin-bottom: 10px;
}

.nowrap {
  vertical-align: middle;
  position: relative;
  padding-left: 35px;
  margin-bottom: 15px;
  cursor: pointer;
  -webkit-user-select: none;
  -moz-user-select: none;
  -ms-user-select: none;
  user-select: none;
  margin-left: 10px;
}
```

```
/* Hide the browser's default radio button */
.nowrap input {
  position: absolute;
  opacity: 0;
  cursor: pointer;
}

/* Create a custom radio button */
.radious {
  position: absolute;
  top: 0;
  left: 0;
  height: 25px;
  width: 25px;
  background-color: #eee;
  border-radius: 50%;


}

/* On mouse-over, add a grey background color */
.nowrap:hover label input ~ .radious {
  background-color: #ccc;
  cursor: pointer;
}

/* When the radio button is checked, add a blue background */
.nowrap label input:checked ~ .radious {
  background-color: #0080ff;
}

/* Create the indicator (the dot/circle - hidden when not checked) */
.radious:after {
  content: "";
  position: absolute;
  display: none;
}

/* Show the indicator (dot/circle) when checked */
.nowrap label input:checked ~ .radious:after {
```

```
  display: block;
}

/* Style the indicator (dot/circle) */
.nowrap .radious:after {
  top: 9px;
  left: 9px;
  width: 8px;
  height: 8px;
  border-radius: 50%;
  background: white;
}

/* Create a custom checkbox */
.checkmark {
  position: absolute;
  top: 0;
  left: 0;
  height: 25px;
  width: 25px;
  background-color: #eee;
}

/* On mouse-over, add a grey background color */
.nowrap:hover label input ~ .checkmark {
  background-color: #ccc;
  cursor: pointer;
}

/* When the checkbox is checked, add a blue background */
.nowrap label input:checked ~ .checkmark {
  background-color: #0080ff;
}

/* Create the checkmark/indicator (hidden when not checked) */
.checkmark:after {
  content: "";
  position: absolute;
  display: none;
```

```
}

/* Show the checkmark when checked */
.nowrap label input:checked ~ .checkmark:after {
   display: block;
}

/* Style the checkmark/indicator */
.nowrap .checkmark:after {
   left: 9px;
   top: 5px;
   width: 5px;
   height: 10px;
   border: solid white;
   border-width: 0 3px 3px 0;
   -webkit-transform: rotate(45deg);
   -ms-transform: rotate(45deg);
   transform: rotate(45deg);
}

html,
body,
#cesiumContainer {
   width: 100%;
   height: 100%;
   margin: 0;
   padding: 0;
   overflow: hidden;
   font-family: sans-serif;
   background: #000;
}

button.cesium-infoBox-camera {
   display: none;
}



#toolbar table tr {
   transform: translateY(0);
```

```
  transition: transform 0.4s ease-out;
}

#menu {
  position: absolute;
  left: 10px;
  top: 10px;
}
```

App.js

```
(function () {
  "use strict";

  // Access token from cesium.com/ion/
  Cesium.Ion.defaultAccessToken = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdG-
kiOiIwOWE1YjQyYi1mNDVlLTQ3NTAtYWRhOC1kNDEwMTJlN2JhZGQiLCJpZCI6OTcyOCwic2NvcGVzIjpbImFzbCIsImFzciI-
sImFzdyIsImdjIl0sImlhdCI6MTU2MDk0NzA4Mn0.jMAyNQnXt6p8imNrPCSTYHi_-9eg6aTxYbiiFBWLipM';

  //////////////////////////////////////////////////////////////////////
  // Creating the Viewer
  //////////////////////////////////////////////////////////////////////



  var currentDate = Cesium.JulianDate.fromDate(new Date(2019, 7, 17));
  var newTime = Cesium.JulianDate.addHours(currentDate, 12, new Cesium.JulianDate());
  var clock = new Cesium.Clock({
    currentTime: newTime
  });

  var viewer = new Cesium.Viewer('cesiumContainer', {
    infoBox: false,
    selectionIndicator: false,
    shadows: false,
    shouldAnimate: true,
    clockViewModel: new Cesium.ClockViewModel(clock),
    navigationHelpButton: true,
    navigationInstructionsInitiallyVisible: true,
    scene3DOnly: true,
    homeButton: false,
  });




  //////////////////////////////////////////////////////////////////////
  // Configuring the Scene
  //////////////////////////////////////////////////////////////////////

  // // Create an initial camera view
```

```
var initialPosition = new Cesium.Cartesian3.fromDegrees(13.72055556, 51.0215, 200);
var initialOrientation = new Cesium.HeadingPitchRoll.fromDegrees(25, -13, 0);
var homeCameraView = {
  destination: initialPosition,
  orientation: {
    heading: initialOrientation.heading,
    pitch: initialOrientation.pitch,
    roll: initialOrientation.roll
  }
};

// // Set the initial view
viewer.scene.camera.setView(homeCameraView);

// // Enable lighting based on sun/moon positions
viewer.scene.globe.enableLighting = true;



//////////////////////////////////////////////////////////////////////////
// Loading and Styling Entity Data
//////////////////////////////////////////////////////////////////////////

// //List of Geometries (function to read JSON file)
function readTextFile(file, callback) {
  var rawFile = new XMLHttpRequest();
  rawFile.overrideMimeType("application/json");
  rawFile.open("GET", file, true);
  rawFile.onreadystatechange = function () {
    if (rawFile.readyState === 4 && rawFile.status !== "200") {
      callback(rawFile.responseText);
      return rawFile.responseText;
    }
  }
  rawFile.send(null);
}

// //Read file and load geometries
var buildingsPromise = Cesium.GeoJsonDataSource.load('./Source/data.geojson');
```

```
function createModel(urlAspect, infoWireframe) {
    viewer.scene.primitives.removeAll();
    buildingsPromise.then(function (dataSource) {

        var buildingEntities = dataSource.entities.values;
        var scene = viewer.scene;
        for (var i = 0; i < buildingEntities.length; i++) {
            var entity = buildingEntities[i];
            //entity.label = undefined;
            entity.name = entity.properties.Name;
            var sourceFolder = './Source/Buildings/';
            var aspectFolder = urlAspect + '/';
            var buildingsNames = entity.properties.Name;
            var origin = entity.position._value;
            var urlBuilding = sourceFolder + aspectFolder + buildingsNames;
            var modelMatrix = Cesium.Transforms.eastNorthUpToFixedFrame(origin)

            var gltfModel = new Cesium.Model.fromGltf({
                name: undefined,
                url: urlBuilding,
                show: true,
                modelMatrix: modelMatrix,
                scale: 1.0,
                debugWireframe: infoWireframe,
                allowPicking: false,
                debugShowBoundingVolume: false

            })

            var model = scene.primitives.add(gltfModel);

        }
        console.log(entity);
    });
}

var treesPromise = Cesium.GeoJsonDataSource.load('./Source/trees.geojson');
var treesInstances = [];
var treeCollection;
```

```
function createTreeModel(showModel) {
    treesPromise.then(function (dataSource) {
        viewer.scene.primitives.remove(treeCollection);

        var treesEntities = dataSource.entities.values;
        var scene = viewer.scene;
        for (var i = 0; i < treesEntities.length; i++) {
            var entity = treesEntities[i];
            var sourceFolder = './Source/TreeModel.gltf';
            var origin = entity.position._value;
            var scale = Math.random() * (4.0 - 2.0) + 2.0;


            var modelMatrix = Cesium.Transforms.eastNorthUpToFixedFrame(origin)
            Cesium.Matrix4.multiplyByUniformScale(modelMatrix, scale, modelMatrix);

            treesInstances.push({
                modelMatrix: modelMatrix
            });

        }

        treeCollection = new Cesium.ModelInstanceCollection({
            url: sourceFolder,
            instances: treesInstances,
            show: showModel
        });

        var model = scene.primitives.add(treeCollection);

    });
}


var lightPromise = Cesium.GeoJsonDataSource.load('./Source/streetlamps.geojson');
var lightInstances = [];
var lightCollection;
```

```
function createLightModel(showModel) {
    lightPromise.then(function (dataSource) {
        viewer.scene.primitives.remove(lightCollection);

        var lightEntities = dataSource.entities.values;
        var scene = viewer.scene;
        for (var i = 0; i < lightEntities.length - 1; i++) {
            var entity = lightEntities[i];
            var sourceFolder = './Source/streetlamp.gltf';
            var origin = entity.position._value;

            var modelMatrix = Cesium.Transforms.eastNorthUpToFixedFrame(origin)
            Cesium.Matrix4.multiplyByUniformScale(modelMatrix, 150.0, modelMatrix);

            lightInstances.push({
                modelMatrix: modelMatrix
            });

        }

        lightCollection = new Cesium.ModelInstanceCollection({
            url: sourceFolder,
            instances: lightInstances,
            show: showModel

        });

        var model = scene.primitives.add(lightCollection);

    });
}

// initial state
createModel('None', false);
createTreeModel(true);
createLightModel(true);

/////////////////////////////////////////////////////////////////////////
// Adaptive viewer (change of textures)
```

```
////////////////////////////////////////////////////////////////////////////



// // Create all the base layers that this example
var imageryLayers = viewer.imageryLayers;

var viewModel = {
  layers: [],
  baseLayers: [],
  selectedLayer: null,
  isSelectableLayer: function (layer) {
    return this.baseLayers.indexOf(layer) >= 0;
  },
  silhouetteColor: 'Red',
  silhouetteAlpha: 1.0,
};
var baseLayers = viewModel.baseLayers;

Cesium.knockout.track(viewModel);

function setupLayers() {

  addBaseLayerOption(
    'None',
    undefined); // the current base layer
  addBaseLayerOption(
    'Artistic',
    Cesium.createOpenStreetMapImageryProvider({
      url: 'https://stamen-tiles.a.ssl.fastly.net/watercolor/',
      fileExtension: 'jpg',
      credit: 'Map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under CC BY SA.'
    }));
  addBaseLayerOption(
    'Pencil',
    Cesium.createOpenStreetMapImageryProvider({
      url: 'https://cartodb-basemaps-{s}.global.ssl.fastly.net/light_all/',
      fileExtension: 'jpg',
      credit: 'Map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under CC BY SA.'
    }));
```

```
  addBaseLayerOption(
    'Contour',
    Cesium.createOpenStreetMapImageryProvider({
      url: 'https://cartodb-basemaps-{s}.global.ssl.fastly.net/light_all/',
      fileExtension: 'jpg',
      credit: 'Map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under CC BY SA.'
    }));
  addBaseLayerOption(
    'CNN',
    Cesium.createOpenStreetMapImageryProvider({
      url: 'https://cartodb-basemaps-{s}.global.ssl.fastly.net/light_all/',
      fileExtension: 'jpg',
      credit: 'Map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under CC BY SA.'
    }));

}

function addBaseLayerOption(name, imageryProvider) {
  var layer;
  if (typeof imageryProvider === 'undefined') {
    layer = imageryLayers.get(0);
    viewModel.selectedLayer = layer;
  } else {
    layer = new Cesium.ImageryLayer(imageryProvider);
  }

  layer.name = name;
  baseLayers.push(layer);
}


function updateLayerList() {
  var numLayers = imageryLayers.length;
  viewModel.layers.splice(0, viewModel.layers.length);
  for (var i = numLayers - 1; i >= 0; --i) {
    viewModel.layers.push(imageryLayers.get(i));
  }
}
```

```
setupLayers();
updateLayerList();

//Bind the viewModel to the DOM elements of the UI that call for it.
var toolbar = document.getElementById('toolbar');
Cesium.knockout.applyBindings(viewModel, toolbar);

// // Define the Basemap according to the texture
var textureFolder = 'none';
var showTree = true;
var showLights = true;
Cesium.knockout.getObservable(viewModel, 'selectedLayer').subscribe(function (baseLayer) {
  // Handle changes to the drop-down base layer selector.
  var activeLayerIndex = 0;
  var numLayers = viewModel.layers.length;
  for (var i = 0; i < numLayers; ++i) {
    if (viewModel.isSelectableLayer(viewModel.layers[i])) {
      activeLayerIndex = i;
      break;
    }
  }
  var activeLayer = viewModel.layers[activeLayerIndex];
  var show = activeLayer.show;
  var alpha = activeLayer.alpha;
  imageryLayers.remove(activeLayer, false);
  imageryLayers.add(baseLayer, numLayers - activeLayerIndex - 1);
  baseLayer.show = show;
  baseLayer.alpha = alpha;
  updateLayerList();

  //Update geometries
  textureFolder = baseLayer.name;
  createModel(textureFolder);
  createTreeModel(showTree);
  createLightModel(showLights);
});
```

```
//////////////////////////////////////////////////////////////////////////
// Clock interaction
//////////////////////////////////////////////////////////////////////////

//general info about the building

viewer.clock.clockRange = Cesium.ClockRange.LOOP_STOP;

var hourPlusButton = document.getElementById('hourPlus');
var hourMinusButton = document.getElementById('hourMinus');
var timeLabel = document.getElementById('timeLabel');

// The clock tick listener gets called every animation frame.
// Keep it fast and try not to allocate any memory if possible.
viewer.clock.onTick.addEventListener(function (clock) {
  var elapsed = Cesium.JulianDate.secondsDifference(
    clock.currentTime, clock.startTime);
  var hours = Math.floor(elapsed / 3600);
  elapsed -= (hours * 3600);
  var minutes = Math.floor(elapsed / 60);
  elapsed -= (minutes * 60);
  //timeLabel.textContent = hours + ' hr ' + minutes + ' min ' +
  //   elapsed.toFixed(1) + ' sec';
});

// Button click callbacks are free to allocate memory.
hourPlusButton.addEventListener('click', function () {
  viewer.clock.currentTime = Cesium.JulianDate.addSeconds(
    viewer.clock.currentTime, 3600, new Cesium.JulianDate());
}, false);

hourMinusButton.addEventListener('click', function () {
  viewer.clock.currentTime = Cesium.JulianDate.addSeconds(
    viewer.clock.currentTime, -3600, new Cesium.JulianDate());
}, false);
```

```
////////////////////////////////////////////////////////////////////////
// Setup Time Mode
////////////////////////////////////////////////////////////////////////

var dayModeElement = document.getElementById('dayMode');
var nightModeElement = document.getElementById('nightMode');

// // Change the day time, from DAY to NIGHT mode
function setTimeMode() {
   if (nightModeElement.checked) {
      var nightTime = Cesium.JulianDate.addHours(currentDate, 25, new Cesium.JulianDate());
      viewer.clockViewModel.currentTime = nightTime;
   } else {
      var dayTime = Cesium.JulianDate.addHours(currentDate, 12, new Cesium.JulianDate());
      viewer.clockViewModel.currentTime = dayTime;
   }
}

nightModeElement.addEventListener('change', setTimeMode);
dayModeElement.addEventListener('change', setTimeMode);



////////////////////////////////////////////////////////////////////////
// Setup Display of Additional Features
////////////////////////////////////////////////////////////////////////

//Shadows
var shadowsElement = document.getElementById('shadows');

shadowsElement.addEventListener('change', function (e) {
   viewer.shadows = e.target.checked;
});


//Trees and Street lignts
var treesElement = document.getElementById('trees');

treesElement.addEventListener('change', function (e) {
```

```
        showTree = e.target.checked;
        createTreeModel(e.target.checked);

    });



    var lightsElement = document.getElementById('lights');

    lightsElement.addEventListener('change', function (e) {
        showLights = e.target.checked;
        createLightModel(e.target.checked);

    });

    //Wireframe
    var wireframeElement = document.getElementById('wireframe');

    wireframeElement.addEventListener('change', function (e) {

        createModel(textureFolder, e.target.checked);
        createTreeModel(showTree);
        createLightModel(showLights);

    });

}());
```