DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Pipeline for Integrating Localization of Outdoor Points of Interest for an Augmented Reality Application

Paul Kehnel

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Pipeline for Integrating Localization of Outdoor Points of Interest for an Augmented Reality Application

Pipeline zur Integration der Lokalisierung von Sehenswürdigkeiten im Freien für eine Augmented-Reality-Anwendung

Author:	Paul Kehnel
Supervisor:	Prof. Gudrun Klinker, Ph.D.
Advisor:	Dipl.-Inf. Univ. David A. Plecher, M.A.
Submission Date:	15.02.2021

I confirm that this master's thesis in informatics is my own work and I have documented all sources and material used.

Munich, 15.02.2021 Paul Kehnel

Acknowledgments

I would like to express my gratitude to Prof. Gudrun Klinker for giving me the opportunity to write my master's thesis. I would also like to thank my advisor Dr. David Plecher for our regular and spontaneous meetings and his continuous feedback that helped me a lot throughout the thesis. Finally, I want to thank Benedikt Wiberg for helping me, when I struggled with the project.

I would also like to express my gratitude to Lou, Seb, Addi, and my family for their moral support and warm encouragement.

Abstract

Context. Many applications exist that combine Augmented Reality and Cultural Heritage, as both fields seem to be made for one another. Most approaches only work indoors, since outdoor conditions are much more challenging for visual approaches. There is a complete absence of open source solutions. Therefore the goal of this project is to build an open-source application that uses augmented reality to enhance sightseeing, by precisely overlaying content over real world objects, e.g. monuments, buildings, gardens, ruins or other sites of interest. **Method**. We use real-time visual localization, enabled through a state-of-the-art structure from motion algorithms to superimpose content over arbitrary objects. For demonstrating and testing our approach, we chose the Sendlinger Gate in Munich, where we overlay historical and contemporary content, in form of images. **Results**. We created a well-functioning application working very reliable and stable even under varying weather or seasonal conditions. Our App *ENSE* (Enhance Sightseeing) is designed to augment the real time experience of sightseeing by superimposing content – previously created by experts – over the object, thus allowing access to former realities, shapes and stories of the site from different timelines[1]. It can be combined with additional text-based information. Due to Covid-19 and the associated curfews, we had to make some cutbacks when testing the app. However, ENSE is set up in a way, that it is easily understandable and reusable. The aim was to develop an open source tool, easy to use and hopefully to be developed further by other researchers, museums, project managers etc. when developing 'new ways of seeing' of our cultural heritage.

[1]Demo Video for the App: https://youtu.be/N2el-QiziO4 (visited on 02/08/2021)

Kurzfassung

Kontext Eine vielzahl an Anwendungen existiert, die Augmented Reality und kulturelles kombinieren, die Bereiche scheinen für einander gemacht zu sein. Die meisten Ansätze funktionieren jedoch nur in Innenräumen, da Außenbedingungen für visuelle Ansätze zu schwierig sind. Auch gibt es einen absoluten Mangel an Open-Source-Lösungen. Ziel dieses Projekts ist es daher, eine Open-Source-Anwendung zu erstellen, die Augmented Reality verwendet, um Sightseeing zu veränden, indem Inhalte über die reale Welt gelegt werden. **Methodik** Wir verwenden visuelle Lokalisierung in Echtzeit, dies wird durch moderene Structur from Motion Algorithmen ermöglicht, um Inhalte über beliebigen Objekten zu legen. Um unseren Ansatz zu demonstrieren und zu testen, haben wir das Sendlinger-Tor in München gewählt, wo wir historische und zeitgenössische Inhalte in Form von Bildern überlagern. **Ergebnisse** Wir haben eine gut funktionierenden App, namens ENCE (enhance sightseeing), entwickelt, die auch bei wechselnden Wetter und Jahreszeiten sehr zuverlässig und stabil arbeitet und es ermöglicht, verschieden Kontent anzuzeigen und zusätzliche Informationen zur verfügung zu stellen[1]. Aufgrund von Corona und den damit verbundenen Ausgangssperren konnten wir unsere Studie nicht wie geplant durchführen. Unsere Arbeit ist so angelegt, dass sie leicht verständlich und wiederverwendbar ist, damit andere Forscher bestimmte Teile oder die gesamte Pipeline in ihren Projekten problemlos einsetzen können.

[1]Demo Video für die App: https://youtu.be/N2el-QiziO4 (visited on 02/08/2021)

Contents

1. Introduction

"Imagine a technology with which you could see more than others see, hear more than others hear, and perhaps even touch, smell and taste things that others can not." [125] Augmented Reality (**AR**) is a fascinating technology, that in recent years is steadily on the rise, currently only hindered by the limitation of computing power and imagination.

The goal of this thesis is building an application that uses augmented reality to enhance sightseeing. You walk through your city, point an AR capable device at a sight, and see altered content, historical, futuristic or artsy. With precise localization in real-time, enabled through state-of-the-art structure from motion algorithms [112], our tool aims to enable others to use AR in Cultural Heritage in their projects and work as a standalone app for sightseeing.

1.1. Motivation

During the last years many projects [20, 61, 129, 133, 4] started combining cultural heritage, tourism, 3D modeling [107] and augmented reality [129]. Today the value [32, 65] and potential of AR for cultural heritage is "very well-known but there is a lack of reliable, precise and flexible solutions, possibly open-source" [104] to build outdoor AR applications with 3D models.

To the best of our knowledge, no out of the box solution exists, that is free and open-source software (**FOSS**), in which you build a 3D model of an outdoor object from images and add AR content that is precisely superimposed onto the real-world object. We think this would be an amazing tool for many projects in cultural heritage.

1.2. Research Question

The purpose of this work is to build a tool, to create and show AR content for outdoor buildings. For testing and evaluating, we apply it to a cultural heritage building and see if the sightseeing experience of a user is enhanced.

> *RQ: Is using our app superior to a classical sightseeing experience in front of a sight, regarding users learning and overall experience?*

1.3. Goal

We want to build a tool, that in a first step uses structure from motion to reconstruct a 3D model from images and then matches future query images against the model and compute

the exact pose of the camera. In a second step AR content, from historical or artistic images, is created and overlaid onto the real object.

In contrast to other projects where AR content is placed over buildings, our solution is open source. Furthermore, we also aim for an approach that is of general nature and not tied to a certain building or object enabling other scientists and developers to use this application. Our last distinctive feature is the usage of state-of-the-art visual localization in an outdoor environment, which leads to an extremely accurate positioning of the user and thus enables an immersive experience.

To achieve this we rely on three principles: **Simplicity**, **Extensibility** and **Reusability**. We want the tool to be easy to use, adjustable and understandable, so the entry barrier is as low as possible.

Finally, we want to showcase the potential of the project with an example. We chose the Sendlinger Gate in Munich to test the possibilities and the limitations of our approach.

1.3.1. Simplicity

For us, simplicity has two meanings. First, we want the process of using the tool to be straightforward. For the user setting up a new sight should consist of as few steps as possible and require a minimum amount of time.

Second, we want to keep the architecture of the application simple. For key features, e.g. the visual localization, we still aim for state of the art algorithms, for secondary functionality, e.g GPS or UI, we focus not on reinventing the wheel, by using existing libraries and solutions.

1.3.2. Extensibility

As mentioned above, there are many projects combining AR and cultural heritage, but hardly any of them explain their architectures, let alone publishing their codes. A rare exception is MauAR [88], a project where you walk along the former border in Berlin and see the Berlin wall, as one the few projects with its codebase published on GitHub[1]. For most projects we researched, a lack of documentation, maintenance, and project structure makes it impossible, to build on top of it, fix errors or reuse it for future projects. By using a well-defined API, software architecture, and good documentation, we circumvent this situation and allow others to build on top of our project.

1.3.3. Reusability

The key to reusability is project planning and project documentation. Following programming principles, e.g. the Single Responsibility Principle [85], design patterns [62], e.g. Encapsulation, and providing comprehensive documentation, helps other people to understand and use the project, while simultaneously facilitating the maintenance of the project. Another aspect of project planning is defining tasks and responsibilities, so modules are easily replaceable at a later date and problems precisely remedied and avoided in the long term.

[1] https://github.com/BerlinerMauAR/MauAR (visited on 01/19/2021)

1.3.4. Technical Requirements

For building the tool successfully three main technical conditions have to be fulfilled:

- Straight forward solution to create and add new sights.
- Simple way to create content.
- Stable and fast localization.

The focus of this thesis, starting from related work down to the evaluation and future work to be done, is set on these three basic and fundamental requirements.

2. Related Work

In this section, we review and summarize works that are important to different parts of our project. Thereby the four main areas are Augmented Reality (**AR**), Computer Vision (**CV**), App Development, and Software Architecture.

2.1. Augmented Reality (AR)

AR's most common definition was created by Azuma et al. as "a system that combines real and virtual content, provides a real-time interactive environment, and registers in 3D" [11]. While this definition is still valid, over the last years the understanding has slightly changed. Now AR is more seen as a technology "that enhances our view of the real world by adding virtual and computer-generated information" [15].

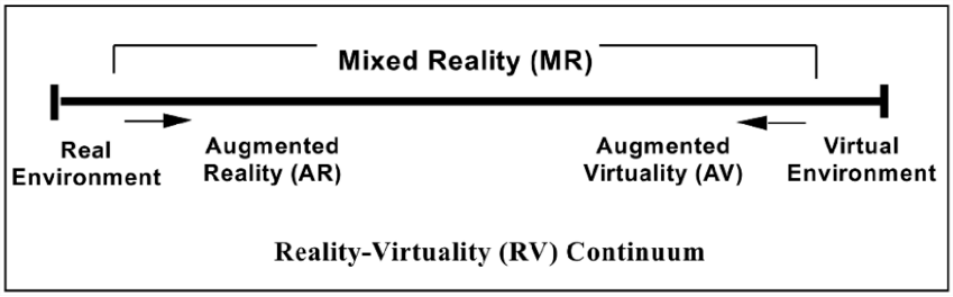


Figure 2.1.: Simplified representation of a RV Continuum From Poul Milgram [93].

Another common way to define AR is with the the Reality-Virtuality Continuum (Fig 2.1). It describes the span between real and virtual environments. At the left it "starts from a real environment and shows the "road" to a totally virtual environment, passing through augmented reality and augmented virtuality" [129].

Over the years many survey and reviews [11, 15, 21, 22, 28, 125] about AR and the history of AR have been published. So we shortly highlight the most important.

In particular, we point to the work of van Krevelen and Poelman "A Survey of Augmented Reality Technologies, Applications and Limitations" [125], which could be the most comprehensive and filling work to date. It gives an introduction to the topic and shows the possibilities and limitations of AR. They use the "reality-virtuality continuum" [93] to derive a definition of AR. Next they summarize the history of AR, picturing and explaining famous milestones like the "Sword of Damocles" from Ivan Sutherland, the mechanical tracking system for the world's first head-mounted display.

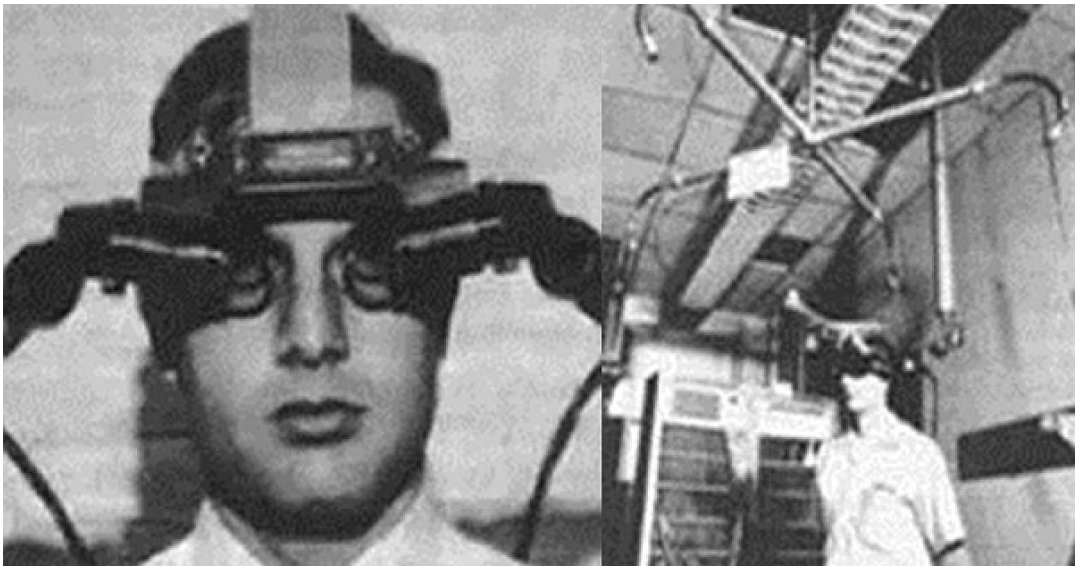


Figure 2.2.: 1986, the world's first head-mounted display, with the "Sword of Damocles" [121].

Then they give an overview of the most common application fields, including personal information systems, assembling, gaming, navigation and education. For each entry they give a short example and show the benefits. Finally, they cover limitations of the technology, e.g portability and outdoor use, depth perception, and social acceptance.

Two papers from Azuma et al. also belong to the most influential works in the area [9, 11]. They establish one of the nowadays most used definitions of an AR System, which was also adopted by Krevelen and Poelman. An AR System [9]:

- combines real and virtual objects in a real environment;
- runs interactively, and in real-time; and
- registers (aligns) real and virtual objects with each other.

The first paper, is a pure survey of the AR field, describing "the medical, manufacturing, visualization, path planning, entertainment, and military applications that have been explored [11]." The second paper is a continuation, providing an overview of the "rapid technological advancements" [9] since the first paper. Through both works, the focus is always on AR as a system.

Last we want to present the "Survey of Augmented, Virtual, and Mixed Reality for Cultural Heritage" by M. Kassahun et al. [15], because they in contrast to the others have a strong focus on AR in the cultural world. They also utilize the Reality-Virtuality Continuum from Fig 2.1 as introduction. Then they cover a lot of technical aspects of AR, like tracking and give an overview over existing toolkits, frameworks, and software development kits **(SDKs)** for developing. Finally, they build an almost exhaustive list of existing AR applications, where they give a summary for each entry, which they then categorize into five main groups:

"Education, Exhibitions, Reconstruction, Virtual Museums and Exhibition Enhancement". Making this survey a good entry point to check for existing projects and get inspiration.

2.2. AR Challenges

AR as amazing a technology it may be, has some pitfalls and obstacles to overcome before it can be used successfully in day-to-day life. Limitations [125] include technical topics like depth perception and tracking but also other aspects as social acceptance. In consideration of the project scope, the most important challenges to consider are **AR in outdoor environments**. "Although there have been a lot of work on using vision-based method to perform registration for indoor AR system, it is very difficult to apply such registration method for outdoor AR systems" [57] due to changing weather and seasonal conditions as well as view restrictions from cars, pedestrians or a construction side this is a challenging task. Another challenge is **running in real-time** with limited computing power and on a mobile device. Out-sourcing heavy computations can be helpful, but does not solve all problem, and even introduces new issues. Other typical problems are virtual objects moving around on the screen, also called drifting [10], due to tracking instabilities or lightning conditions interfering with the device's capability to render objects. Next the whole **development area is extreme volatile**. A current framework can quickly vanish two years from now. E.g. Tango, a phone and tablet-based mobile AR solution, was shut down by Google in order to focus on the more mass-market ARCore product [129]. This development goes even faster for functionality. Trying to run a project created three years ago, goes along with a lot of struggles as functions are depreciated or not supported on modern devices anymore and the support for a specific version might be discontinued. This brings us to the last point, **hardware restrictions**. Not only operating systems **(OS)** but also hardware differs vastly between devices. So it's always a consideration if you are building an application that runs on as many devices as possible or if you use specific properties, like an RGB-D camera, but therefore only a few selected devices.

Through the next sections these limitations are picked up again and possibilities are examined how to handle or circumvent them.

2.3. AR Devices

Figure 2.3.: Hololens a Head-Mounted Display (HMD) from Microsoft [90]

There are three main visualization technologies used in AR: head-mounted displays (**HMD**s), handheld displays (**HHD**s) and spatial displays [21].

HMDs, like the Hololens in Fig 2.3, are worn on the head and overlay virtual environment elements over the user's view of the real world. Handheld displays, as the name suggests, are held in the user's hand, but otherwise, work exactly like HMDs. Spatial displays work with video-projectors, holograms, or similar technology. They display graphical information directly onto physical objects. The advantage being that the user is not required to carry any physical device, in his hands or on his head.

For cultural heritage, in the beginning, HMDs were used, but nowadays almost all solutions work with HHDs, which are mainly smartphones or tablets, or sometimes spatial displays like projectors in museums.

2.4. AR and Cultural Heritage

Maybe it's in the nature of things, that people working in cultural heritage (**CH**) and tourism, are most often not computer scientists and vice versa. Apart from prestige projects, money and resources are normally tight for cultural projects. Nevertheless, there seems to be a natural bound, that AR and CH are meant for each other [127].

The adaptation of AR in CH projects started as early as 1999 with the exploring MARS project [56]. Where the user would not literally explore Mars, but instead with the help of an experimental **m**obile **a**ugmented **r**eality **s**ystem walk through the world and experience it augmented by multimedia material. In the ARCHEOGUIDE Project from 2002 [128], equipped with an HMD, a camera on top of it, a compass, a GPS receiver, and a laptop inside a backpack the user could walk through the archaeological site of the temple of Hera in Greece and see the reconstructed temple rise over the ruins.

During the last years, tons of works and experiments have been done and published in the fields of history, heritage, and tourism in combination with AR. There seems to be a straightforward idea, to enable people to see history and heritage through "different glasses", as "users are able to experience cultural artifacts in a completely new way" [15].

These projects vary extremely in their approaches and goals. They have been conducted from natural immersion in gardens [47] with AR, over the question if historical sites or slavery become less worthy of interpretation if there are no surviving buildings or images and if you can maybe overcome that with AR [4], to tourist guides, by displaying information on the

screen relative to your location [1]. And also in closely related areas, like serious gaming, multiple approaches have been taken to combine CH and AR. Be it a location-based game [49] to explore CH or to learn historical topics like hieroglyphs [105]. But also more theoretical questions have been studied [22]: "Is augmented reality capable of conveying the emotional weight of historical events? Will augmented reality be appropriate for teaching a complex field such as the Holocaust?

When in the early days the user would carry a whole system of equipment (See the MARS Project in Fig 2.4), this rapidly changed and nowadays all that is required is a simple smartphone or tablet [133], allowing the applications to be used by a variety of different people, for example for visitors in a museum. As for all technologies, the childhood is an adventurous time, with a thirst for knowledge, but at a certain point a shift happens and new projects start to rely more on studies, guidelines are introduced and principles established.

These days the advantages of AR are systematically explored and documented, e.g viewing "variable information about an object of interest that is placed immediately in context" [133]. "Several studies demonstrate that the use of new and combined media enhances how culture is experienced" [15] which not leads to an increased number of people having access to knowledge, but to a different type of diffusion of knowledge. In business style the "stakeholders' perceived value regarding the implementation of AR to enhance the museum experience at cultural heritage sites" [32] is evaluated, for example via studies, where the *social value*, through games like a treasure hunt, that could enhance the social aspect of the museum visit, the *educational value*, as visitors can gather information by themselves and *cultural and historical value*, through additional information and more space with a virtual part of the museum are determined. But also from an education point of view, researchers have shown that AR can improve collaborative learning [35], conceptual understanding [24], and spatial abilities [25]. But also aspects as the acceptance of AR and behavioral intentions [109] are examined.

Figure 2.4.: 1999, a user wearing the MARS prototype [56]

Regardless of all of that, there still is a "lack of established guidelines in the application and integration of AR technologies in outdoor heritage sites"[101] and only in a few cases any project code is published and even more rarely this code is maintained.

Another benefit of AR is environmental immersion. This knowledge comes from the historical AR context, where it is also called "Sense of Place" [23]. "Being physically present at the site provides the user with the sense of historical empathy that cannot be achieved

from a classroom with a textbook" [22]. An example, where this occurs very strongly, are Holocaust memorials, where visitors often show strong feelings and emotions as a reaction to the places. In an AR application where a Holocaust story is told from the perspective of a teenager, the experience was so emotional that in the audience "people were moved to tears, were compelled to question how the holocaust - happened" [120].

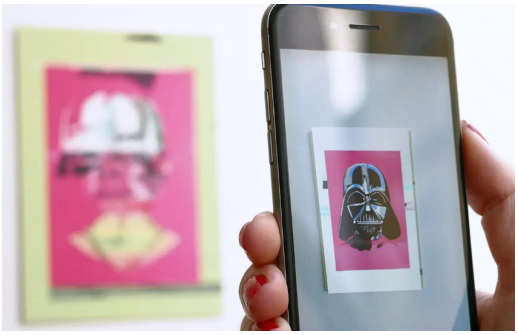


Figure 2.5.: Artivive app – a visualisation tool for AR art [116]

Finally in a museum setting with applications like Artivive [116], "new dimensions of art by linking classical with digital art" can be created, where a digital layer is fused on a classical image (see Fig 2.5). We can transfer this idea to our approach, and keep in mind that we are not limited to historical content, but can also embed artistic content.

2.4.1. Documentation and Digitization

"Cultural heritage structural documentation is of great importance in terms of historical preservation, tourism, educational and spiritual values" [31]. As a result, many computer applications applied to CH are focused on the documentation and digitization of artifacts and sites [103, 129] and 3D reconstruction of real-world objects for heritage preservation is now part of the standard repertoire of scientists. For the longest time methods like laser scanners or modeling software was used. Howeverthis is rather expensive and requires a high level of expertise [107]. With the introduction of multi-image photogrammetry also named structure from motion (**SfM**), as Colmap [115], the whole process was simplified. As a consequence way fewer resources, money and knowledge are required to build a decent 3D model. Consequently, not only professionals, such as archaeologists, architects, and civil engineers but everyone with an interest in this area, can start building 3D models. Today, websites like *sketchfab.com*[1] offer collections of hundreds of thousand free to use 3D models, with many of them originating from CH.

2.4.2. Tourism and Sightseeing

A particular field in CH is tourism and sightseeing. Opposed to museums and excavation sites, a citizen just on his way to work, or when exploring a new city can walk through century-old gates, past ancient palaces and churches. A worker at his daily sprint to the metro may ignore them completely, staring at his phone, while a tourist is fascinated by epochal buildings erected a long time ago, looking at his phone or a guide to get some additional information about the sight. In this area, it comes to no surprise, that "the technology augmented reality" is just on the verge of being implemented in a meaningful way in the tourism industry [50].

[1] https://sketchfab.com/blogs/community/sketchfab-launches-public-domain-dedication-for-3d-cultural-heritage/ (visited on 12/05/2020)

While Yovcheva et al. were maybe the first researchers to see the full potential of AR applications[133] overlaying digital content in their real environments. Nowadays several approaches to enable sightseeing via AR exist. They differ vastly in their goals, their ways of delivering information as well as their functionality.

From a technical point of view, there are two main directions for tracking and registration. **Vision-based** approaches are "best suited for controlled and small environments, but their performance diminishes in wide and outdoor areas" [101], and **sensor-based** approaches, which traditionally perform better in outdoor environments. Tracking via image-recognition [41], marker-based [66] or object-recognition [113], where respectively images, markers or objects are recognized and content content is placed depending on the recognized item, belong all to the first category. Using a geo-location[49, 127], where information is obtained with the global positioning systems (**GPS**) and according to the users position information is selected and shown, is part of the second category.

All of the approaches have different strengths and certain areas where they shine. But also all of them come with severe limitations. Markers are normally not allowed to be placed on public buildings, image-based approaches struggle with weather and seasonal conditions or changes like construction sides. Location-based approaches using GPS can only achieve accuracy up to a certain threshold and object recognition needs to be highly precise and stable to work satisfying and requires way more preliminary work to create a model.

Regardless of the way the approaches work on a technical level, they also have different intentions and goals and there is no strict separation as multiple approaches can be mixed in a project. A location-based approach in Japan from Sasaki et al. tries to give guidance to sightseeing spots and nearby facilities with images and pictograms [113], while also using object-recognition. Another common idea is to overlay a building with images from the past, an engaging way to explore a historic site, where it has shown that "images in addition to the text was clearly the most successful way of attracting attention" [61].

Figure 2.6.: 2011, snapshot from "the House of Olbrich" a building in Darmstadt enhanced with Augmented Reality [67].

Also worth mentioning is the work by Panou et al. "An Architecture for Mobile Outdoors Augmented Reality for Cultural Heritage" [101], as it has many similarities with our approach. They created a mobile tourist guide for CH, where they tried to superimpose 3D models of historical buildings in the real world. In contrast to us, they decided not to use a vision-based approach, as they feared this would not work for outdoor conditions and therefore chose a location-based approach. For interaction, they added gamification elements and also used server-client architecture.

Unfortunately, the code was never published.

Another exciting application in the area "history of architecture", House of Olbrich, was created by Keil et al. [67]. Similar to our work, a user would take a photograph of a building and it would be augmented with a 3D model reconstructed from the original drawings or images. This allows the user to see the original design, of a building that changed its appearances, due to destruction or renovation. As this work is 10 years old now, the technical possibilities were vastly different at that time and so the augmentation only happened for a single image, as seen in Fig 2.6. However, in their future work, they point directly to steps that we implemented in our work, like manipulating the video image and not single images and the possibility to add multiple buildings.

2.5. AR in Games

Unlike in cultural projects there is a lot of money in the gaming industry. Furthermore, the industry is affine to new technologies and experiments. So it comes to no surprise, that the most suitable for everyday uses of AR can be found here as well. With Pokemon Go as the prime example, an AR Game, that broke "all records"[2] at release and still is one of the most downloaded games today.

There are many stories, where an invention was first used in the gaming world and later adapted successfully by the industry. For example, when the US Navy decided to use Xbox3 controllers to navigate their most advanced submarines[3] instead of their previously used special devices costing $38,000. They saved a lot of money, reduced the training time, and got a performance increase. So paying close attention to the gaming industry can pay off heavily. Transferred to AR, it surely is advantageous to try out applications like Pokemon Go and other top performers, when developing an AR application, to see their take on controls, handling, and design.

2.6. AR in the Industry

"While not having gained a substantial foothold on a consumer level compared to VR, AR in support of Industry 4.0 is already being used and implemented" [87]. Here real-time information and the usage of hands-free AR can lead directly to increased efficiency for industrial tasks, like assembly operations. Other than in a museum or on a heritage site, in the industry it is feasibly, to equip and train operators to move around and read information hands-free with the help of an HMD.

Also, the diversity of previously mentioned devices plays a more important role. Gaining increased mobility through a handheld device for certain tasks or allowing the user to work hands-free by using an HMD device [39]. Another difference between CH and the industry

[2]https://www.guinnessworldrecords.com/news/2016/8/pokemon-go-catches-five-world-records-439327 (visited on 12/29/2020

[3]https://bit.ly/39CsiM7 (visited on 12/29/2020)

is, that in the industry the most prominent AR implementation is marker-based, by attaching labels to products, or certain places.

All this is only of limited impact for our work, but we wanted to mention it for the sake of completeness and maybe there is something to learn from it after all.

2.7. Computer Vision (CV)

"Recovering the camera-to-world translation and orientation from an image is one of the fundamental problems in CV. Accurately estimating the absolute pose of the camera is key to applications of augmented reality." [117] In other words, precise 6 degree-of-freedom (**6DoF**) localization of a new camera against a 3D model is a core CV problem, that when solved, enables a number of applications, in the field of AR [70, 91] or for autonomous driving [76, 114].

A subfield of CV is image matching, in fact, one of its oldest tasks. For humans a mostly trivial task, it still remains a difficult problem in computer science once you leave the playing field and tackle real world problems. Outside conditions like weather, daytime, and seasonal change as seen in figure 2.7 cause enormous difficulties. For visual localization algorithms to be applicable to the AR world, they need to not only work indoors and outdoors and be robust under varying conditions, independent of the seasonal changes, the weather or illumination but also "a centimeter-accurate 6-DoF pose is crucial to guarantee reliable and safe operation and fully immersive experiences" [111].

(a) Construction during summer

(b) Cloudy day in Autumn

Figure 2.7.: Easy to spot for the human eye, these images belong to the same building but taken in different positions. However for many computer algorithms, matching these images isn't a trivial task, due to clouds and the ivy with and without leaves.

2.7.1. A Short History of Multiple View Geometry

Some CV problems like perspective projection have already been known and tackled by the ancient Greeks, most prominent by Euclid of Alexandria around 300 B.C. [75, [p.12]

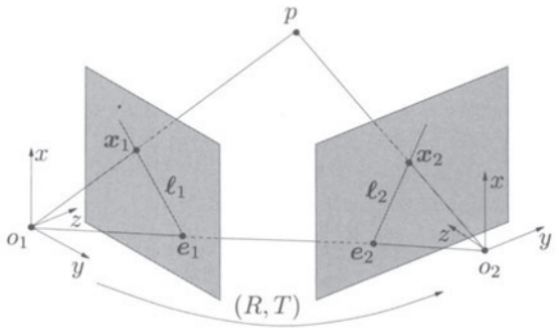


Figure 2.8.: The Euclidean transformation (T) and rotation (R) between two cameras is visualized, $(R, T) \in SE(3)$, the Lie Group. X_1 and X_2 are projections of the point p. [83, p.111]

Closely related multiple view geometry is "[a] basic problem in computer vision [...] to understand the structure of a real-world scene given several images of it." [52]. It was in 1913 that Erwin Kruppa could prove, that in a scenario with two views, only five points are sufficient to determine the transformation and rotation between the two views. [83, p.204] Since then multiple view geometry has been one of the main topics in CV and many algorithms have been discovered to recover structure and motion from two views. With the *f Factorization techniques* by Tomasi and Kanade [80] in 1992 for multiple views and orthogonal projection, being one of the most influential till today.

Finally, the joint estimation of camera motion and 3D location is called visual simultaneous localization and mapping (**SLAM**). Or in other words, SLAM is estimating the camera trajectory while also reconstructing the environment [94]. Until recently it was still unaffordable in regard to computing power to run SLAM in real-time.

2.7.2. Visual Localization

While multiple view geometry is an important fundamental problem, visual localization aims at estimating the exact pose of a query image relative to a 3D model and nowadays is a way more practical task. It enables a wild range of applications, such as autonomous driving or AR applications. "Visual localization approaches need to be robust to a wide variety of viewing condition, including day-night changes, as well as weather and seasonal variations, while providing highly accurate 6 degree-of-freedom (6DoF) camera pose estimates." [114]

Also "visual localization is a key component in computer vision tasks such as SfM or SLAM" [111] visual localization calls for reliable operation both indoors and outdoors, irrespective of the weather, illumination, or seasonal changes.

6-DoF visual localization methods are usually classified as either structure or image-based, both with the aim to estimate the pose of a new query image. The **3D structure based** once use descriptor matching to establish correspondences between a 3D model, made of 3D points in a SfM point cloud and 2D features extracted from a query image[4]. With these matches the

[4]As this is a key component for the project an in-depth explanation can be found in the section 2.7.5 Feature Matching

pose can be computed. [73] A variety of options exist, to speed up the descriptor matching process. This includes prioritization [74], specific search algorithms [82], or hierarchical localization [60]. In huge scenes, for example, outdoor matching, descriptor matches are prone to ambiguity, as locally similar structures can exist in different parts of the scene [73]. The larger the scenes get, as in the project "Building Rome in a day" [2] where 150.000 "internet images" of Rome were used to build a 3D model, the more important it is to take measures to guarantee robustness.

The other approach, **2D image based** localization, also aims at estimating the pose of a query image, but instead of a 3D model a similar photo is used for matching. Place recognition [6] and loop closure [42] are typical fields of application. Of course both methods can be combined.

For many CV tasks such as image classification, object detection, and semantic segmentation the rise of deep learning and Convolutional Neural Networks **(CNNs)** lead to impressive results [117]. So to no surprise researchers also started using deep learning at multiply stages of the SfM process, from computing features to pose estimation. Both these techniques are used by PoseCNN [130] for example.

And so a third approach, that is not part of the classical methods, is using pure machine learning with an end-to-end solution. For example PoseNet [68] uses a "CNN to regress the 6-DOF camera pose from a single RGB image" [68] without any further engineering. As for now these approaches over-fit their training data and are not generalizing well to new scenes. PoseNet for example has a "localization error on indoor and outdoor datasets [...] of magnitude larger, compared to feature-based approaches that are considered state-of-the-art." [117] Even so in the long run, machine learning solutions will most likely overcome these problems and yield top results.

To compare multiple approaches benchmarks [114] are used, e.g. the CMU Season data where for images from divers conditions a 6DoF localization against a known 3D map is computed and evaluated against a ground truth, whereby the conditions include day-night, seasonal and weather changes as shown in Fig 2.9.

Figure 2.9.: Visual localization in changing urban conditions. CMU Seasons Dataset for evaluating 6DoF localization [114]

Over the last years, the current "limitations motivated a surge of deep learning-based methods for absolute pose estimation (APE)." [117] And every year new approaches and with them records are published. But even so the result may be astonishing, it's important to keep in mind that the trade-off in machine learning most of the time comes in form of "the machine's inability to explain its thoughts and actions

to human users" [48] and debugging a "blackbox" is a rather magical task, where you just try the same thing over and over again hope it will work soon.

2.7.3. Tracking and Mapping

For some areas, including AR and autonomous driving, not only the initial localization, but continuous tracking is desired. In AR tracking is required to understand where the user is relative to the world when the user is moving around. In most implementations, a variation of SLAM) is used. In the beginning, a comprehensive map of the environment is not available, so an initial map is created, and with a technique called extensible tracking [70] previously unknown elements are added during the lifetime of the application. An alternative to SLAM is using depth cameras like the KinectFusion [95], where tracking is done with only depth information and a depth map.

For our type of application, concepts like motion tracking, environmental understanding (mapping), and depth understanding are all part of the fundamentals of the AR SDK and handled in the background[5].

2.7.4. 3D Reconstruction

3D reconstructions aim to capture the appearance or geometry of an object. "Developing 3D digital models of heritage assets, monuments, archaeological excavation sites, or natural landscapes is becoming commonplace in areas such as heritage documentation, virtual reconstruction, visualization, inspection of a crime scene, project planning, augmented and virtual reality, serious games, and scientific research." [107]

Building 3D models the conventional geometry-based way, with tools like Blender, is neither easy nor fast. Hardware approaches using laser scanners or light systems are out of reach for non-professional user as the equipment is expensive. Luckily, there is a third approach to reconstruction: SfM. Early experiments with SfM were already conducted in 1976 at the MIT in an Artificial Intelligence Laboratory, regarding the questions: "how the 3-D structure and motion of objects can be inferred from the 2-D transformations of their projected images" [124].

Nowadays SfM is seen as "a pipeline that allows three-dimensional reconstruction starting from a collection of images." [19] The typical building blocks of an SfM pipeline are [19, 115]: *Feature Extraction*, creating local features for each image, *Feature Matching*, finding images that share the same previously extracted features and therefore portray common parts of the scene, *Geometric Verification*, finding a geometric transformation between common points of two images, to verify real correspondences in the scene and if not eliminate outliers, *Reconstruction*, staring the reconstruction process with a pair of geometrically verified images, whereby the common points of them are the initialization points of the reconstructed point cloud. Then in an iterative process add a new image to the reconstruction, using *Triangulation*, defining the

[5]E.G. Document in AR Core of this concepts: https://developers.google.com/ar/discover/concepts (visited on 12/07/2020

Figure 2.10.: Matching example. *Inliners* are points that fit the data model. This image shows 43 inliners (green) and 31 outliers (red). In this case false camera intrinsics where the cause for the high number of outliers.

3D coordinates of new points, and *Bundle Adjustment* preventing propagation of inaccuracies from the camera pose to triangulated points and vice versa.

Multiple FOSS (free and open source software) tools exists to tackle this task, like COLMAP [115], Bundler [119] or Regard3D. Since by and large there are no significant differences between the approaches, the choice is ultimately dependent on project requirements as well as personal preference and background. So for example Regard3D is the easiest to install and requires no programming knowledge, a major plus point for a layman [19].

2.7.5. Feature Matching

In recent years especially the feature matching progress has undergone a lot of changes that lead to huge performance increases in the SfM pipeline. Feature matching is part of the correspondence search between input images, wherein overlapping images, projections of the same points are identified.

Determining good feature points is a challenging task and was first "solved" in 1999 when Lowe et. Al introduced the Scale Invariant Feature Transform, (**SIFT**) [78], as part of an object recognition system. These "features are invariant to image scaling, translation, and rotation, and partially invariant to illumination changes and affine or 3D projection [78]. Until recently SIFT and its derivatives like PCA SIFT [131] have been the gold standard for feature matching.

"The most common approach to sparse feature extraction – the detect-then-describe approach – first performs feature detection and then extracts a feature descriptor from a patch centered around each keypoint." [36] The detector is an algorithm, that selects points from the query image, based on some criteria, e.g. a local maximum of some function. The descriptor, on the other hand, is a vector of values that describes the pixels around the interest point,

e.g. SIFT uses a histogram of gradient orientations from all adjacent pixels. Together the interest point found by the detector and the corresponding descriptor are usually called a local feature. The last step is to match a feature from one image against all features from a different image. The features are compared using a distance measurement like the L2 norm (Euclidean distance). If the value is below a certain threshold, we can assume the features in both images describe the same point and call it a match.

As the name local feature suggests, there are also global features. While local features describe small parts of an image, global features describe an entire image. A very simple version would for example be the average color of an image from a histogram. As global features are compact representations for images, they are often used for image retrieval [64], finding similar images to a query image from a database, or place recognition [7], finding the geolocation of an image. Hence it's common to use global features to select similar images and then perform local feature matching between them.

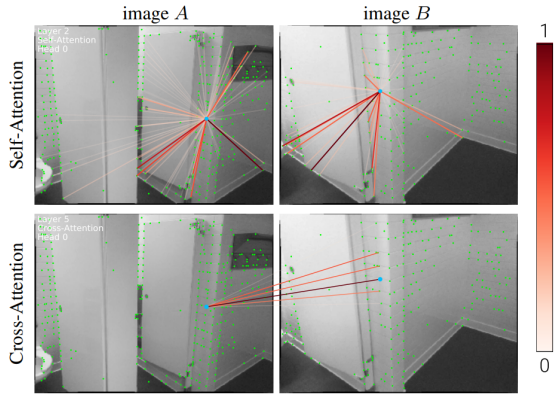


Figure 2.11.: Visualizing self- and cross-attention. Self-attention attends anywhere in the same image, while cross-attention attends to locations in the other image. The weights are shown as rays, with values form $0 - 1$. [112]

Typically methods like SURF [14] and SIFT are also called hand-crafted as they are designed by humans with logic and ideas behind them. Recently researchers started replacing either the descriptor [12], the detector [135] or both with learned alternatives. During the last years, with the rise of deep learning, learned features created with CNNs [29, 36] started to outperform SIFT in terms of "keypoint repeatability and descriptor matching, which are both critical for localization." [111]. Learned features, like LIFT [132], named based on SIFT with the *l* for *learned* or Superpoint [29], which is the current state of the art [117], are not only sparser than traditional handcrafted features, thus reducing the number of keypoints to be matched and speeding up the matching step, but also perform better when comparing pose accuracy (AUC), precision (P), and matching scores (MS). Learned descriptors result in "unrivaled robustness in challenging conditions" [111].

While learned features helped to improve the image-based localization to keep up with the giant 3D models emerging from modern SfM pipelines. Most "approaches are [still] to resource-intensive to run in real-time, let alone to be implemented on mobile devices" [92]. An issue with the structure-based methods is, that as the 3D models grow linearly with the size of the scenery, for the descriptor matching process, the search space is growing as well and as result gets more prone to errors, due to more ambiguous matches and naturally also slower.

The latest development to further improve the matching process and current state of the art is transferring the attention mechanism [126], the reason for the success of NLP [26] and their language models like GPT-3 or Bert [30], to feature learning. SuperGlue, a neural network that matches local features, uses attention "to reason about the underlying 3D scene and feature assignments" [112]. Thereby it uses self-attention, to "boosts the receptive field of local descriptors" and cross-attention, "which enables cross-image communication" as seen in Fig 2.11. A possible interpretation is to compare it to the way humans look back-and-forth between two images when they have to spot differences between them.

For more basic information, there is a well-written article [99] on the OpenCV site, the go-to library for CV algorithms, about feature detection and description including code examples, which we can only recommend.

2.8. App Development

When developing for AR, adjusted to the project requirements, the selection of a suitable development platform and a fitting AR framework is very important. On top of that, every niche in programming has its paradigms and special cases that need to be handled carefully. In the light of developing for mobile devices, for example, compatibility between not only different platforms but also different versions from the operating systems as well as millions of variations in hardware have to be handled.

2.8.1. Development Environment

A possibility to find the most popular development environments is checking for which of them ARCore, the biggest AR Framework (Fig 2.13), provides SDKs[6]: Android, Android NDK, Unity, iOS, and Unreal.

If we first check both solutions that offer support for multiple development frameworks, the two big players are Unity3D[7] and Unreal Engine. "It used to be Unity3D for mobile projects, and Unreal Engine for AAA-games, but things have changed so much since then.[8]

There are small differences in infrastructure between the two, like the assets stores, documentation, and the community. If you prefer C++ over C# as a programming language or require features of an open-source engine, and maybe the application has a heavy focus on graphics then Unreal Engine 4 is the better match. If a bigger community, more out the box solutions and C# are project requirements Unity3D is better fitting.

Finally, if cross-platform compatibility is not desired and your project is planned for Android devices, going for Android Studio or even the Android NDK for more low-level programming is the most obvious solution. This applies analogously for Apple applications, with ARKit and iOS.

[6]Supported SDKs: https://developers.google.com/ar/develop (visited on 12/12/2020)

[7]"53% of top 1,000 grossing mobile games powered by Unity globally" Unity stats: https://unity.com/our-company (visited on 12/12/2020)

[8]"https://circuitstream.com/blog/unity-vs-unreal/ (visited on 12/12/2020)

2.8.2. Development Framework

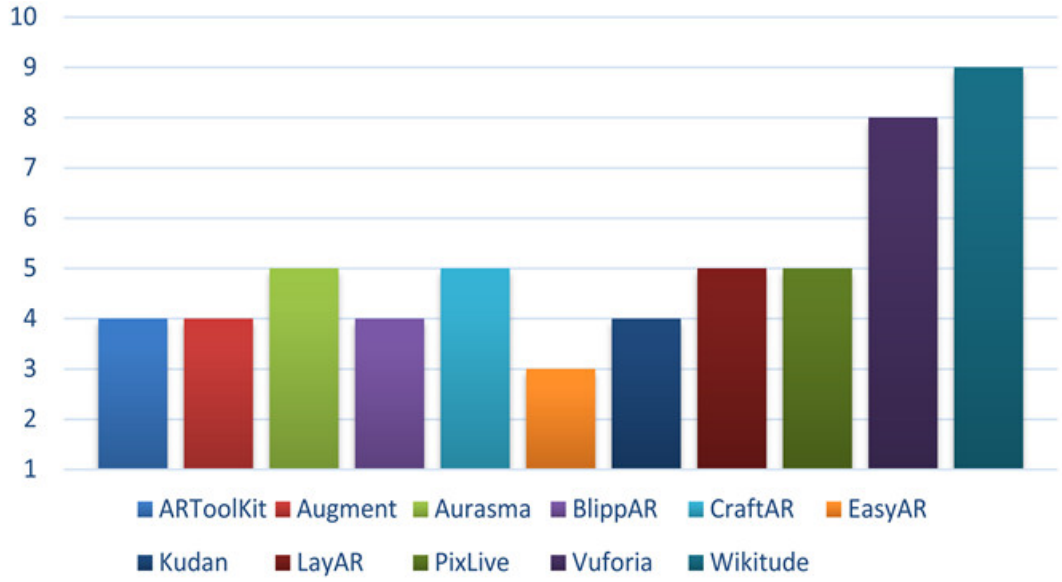


Figure 2.12.: 2017, comparison of different AR Frameworks [54]

As seen in Fig 2.12, which visualizes the results of a three years old study, there is a vast selection of AR frameworks to choose from. Due to the rapid change over the last years and the different application areas like education or entertainment, the results from a 3-year-old study have to be used carefully. A comparison from 2019 claims "the most popular ones are: ARCore, ARKit, ARToolkit, Kudan, MAXST Wikitude" [98], which is similar to an online list, naming Vuforia, Wikitude, ARKit, ARCore and ARToolKit as "5 top ar tools for app development" [123]. Overall they all have similar features as they tackle the same problem. Therefore the application area, the target audience, and the project type are crucial when selecting the framework. Ultimately, we compared their popularity, by doing a search trend analysis, where ARCore was the clear winner as seen in Fig 2.13

Based on the previous information, we decided to briefly compare Vuforia, Wikitude, ARKit, and ArCore.

Vuforia [5] is most likely the solution working best out of the box, with very good customer support and therefore "one of the most popular platforms to help you work with augmented reality" [44, p.14] When using Vuforia, the normal workflow includes scanning the object, as you can not use custom 3D models. This comes with several limitations for the physical object to be a target. It should be opaque, rigid, and contain contrast-based features. Also, it shouldn't contain movable parts. The important thing is that Vuforia Object recognition is optimized for objects that can fit on a tabletop and are found indoors. Lastly, Vuforia is not open source and requires you to purchase a license.

Wikitude is in many ways similar to Vuforia and the newcomer in the group. But with the basic plan costing 2000€ [97] the target group are companies, who want to build an AR

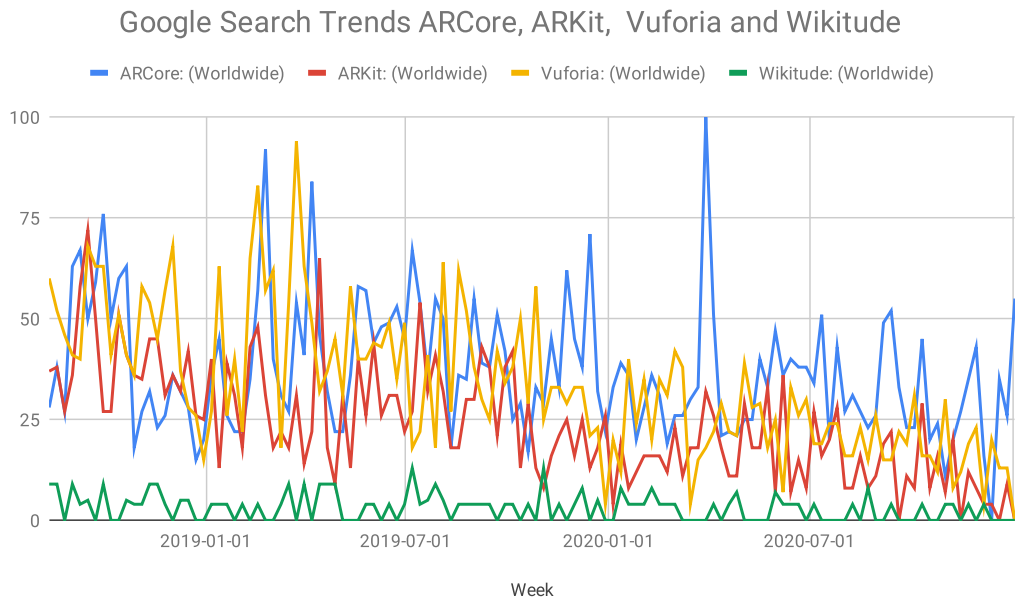


Figure 2.13.: Worldwide interest (Numbers represent search interest relative to the highest point on the chart for the given region and time. A value of 100 is the peak popularity for the term. A value of 50 means that the term is half as popular. The graph was created using google trends: https://trends.google.com/trends/

project and not so much research and custom projects, as opposed to all the other competitors, they don't offer any free plans.

ARKit and **ARCore** are the approaches developed by Apple and Google, both targeted at their own devices and therefore not cross-platform compatible. They are in many ways very similar, as they focus on mobile devices, are free to use, and ship with regular updates. Opposite to ARKit, ARCore is open source.[9]. Both platforms have their strengths and weaknesses, but it is difficult to find technology that would be objectively better. [98] With AR Foundation a Unity high-level, cross-platform API, you can write an application once, and build for both iOS and Android, as AR Foundation "unifies" ARKit and ARCore.

2.8.3. Mobile Development

When developing for mobile devices, there are some special aspects we want to highlight.

1. "Right now we support 5 or 6 different (app) versions only because there are different OS versions" [63] The issue of moving toward fragmentation rather than unification is long existing in the mobile app development market. On top of compatibility issues from OS Versions, for the AR sector, the different types of cameras add even more complexity to this problem.

2. A peculiarity of a high-end smartphone is the camera. And while certain cameras like Stereo or RGB-D can be used to enable depth perception or SLAM [94], special features like this should be used with caution, as the list of cameras or tablets that support this types of functionality is rather short.[10] and leads to even more special cases to be considered when aiming for compatibility.

3. With increased computing power using machine learning on mobile devices is feasible nowadays. Popular architectures like MobileNet [110] are optimized and specifically tailored for mobile devices. Still, mobile devices present resource-constrained environments, even more, if you aim to include the middle and low budget devices. Therefore a lot of state-of-the-art networks require way too many computational resources far beyond the capabilities of most of them.

4. Another important area when developing applications for the consumer market is energy consumption. "Network usage, memory consumption, and low-level programming practices" [72] are categories that help to control the energy drain in smartphones.

2.9. Software Architecture

"The goal of software architecture is to minimize the human resources required to build and maintain the required system." [85] The success of a project depends on many factors. One of

[9] AR Core Github: https://github.com/google-ar/arcore-android-sdk (visited on 12/12/2020)

[10] list of smartphones with depth cameras: https://en.wikipedia.org/wiki/List_of_3D-enabled_mobile_phones (visited on 12/28/2020)

them being software architecture. Oftentimes overlooked or ignored in favor of alleged speed, it will normally come back to haunt you.

One example of this is productivity during the lifetime of a project. "When systems are thrown together in a hurry [...] and when little or no thought is given to the cleanliness of the code or the structure" [85] everything will go amazing and fast in the beginning, but "soon you will find yourself starting to clean up the mess to be able to integrate new features" and wish to start everything from scratch, just to end in the same situation again. This is nicely visualized in the Fig 2.14.

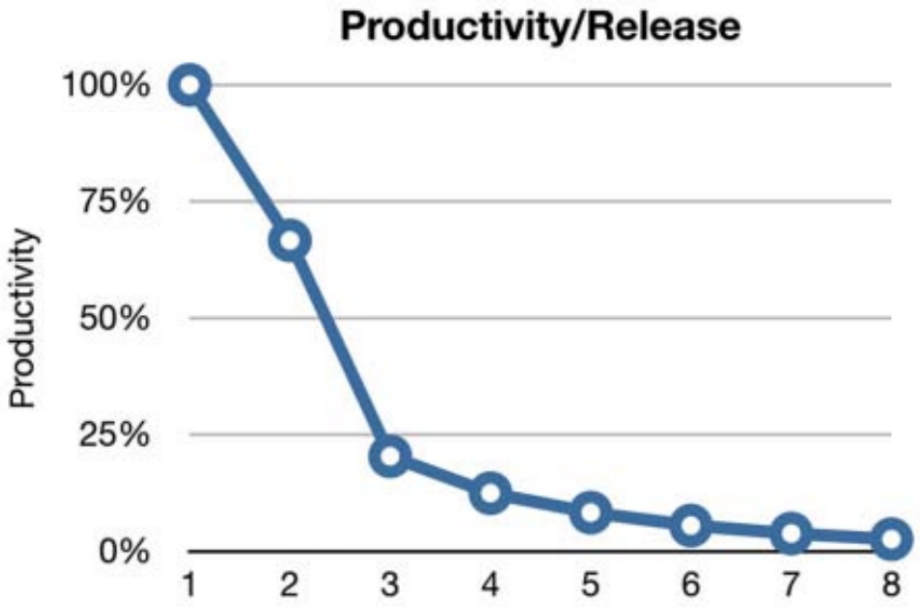


Figure 2.14.: Productivity during the lifecycle of a project [85].

To avoid this, there are principles to impose certain rules on the project, resulting in clean architecture and clean code. This includes structured-, object-oriented- and functional programming.

While the above-mentioned ideas are more focused on the code level, design principles, like "SOLID", help to build a software structure, that tolerates change, is understandable and reusable. This includes concepts like the "Single Responsibility Principle" or the "Open-Closed Principle", [89], meaning that a "software artifact should be open for extension but closed for modification" [85].

3. Project Architecture and Development

The project started of from the work done by P. Tolstoi in his master thesis [122]. So in this chapter, we briefly outline his work and show what lessons we could learn from him. Then we showcase the project architecture, which we build for our approach. As we see software architecture as one of the three main pillars for the success of our project, a considerable amount of time was invested into planning the architecture, sketching for the application, and defining the API.

3.1. Previous Work

"A Framework for location-based Augmented Reality Content on Mobile Devices" [122] by Paul Tolstoi, is the predecessor for this project. In that work, the idea of building a tool for location-based AR content was developed and first tested. During that, a successful prototype was developed and tested at the "Siegestor" in Munich. The paper also includes a rich introduction to AR and gives an exhaustive overview of location-based applications, with and without AR.

We try to extend this idea and build on top of it. While using the knowledge gained in that process, we tried to learn from their mistakes and integrate the ideas mentioned in future work and the findings from the evaluation.

For example in the backend two main problems were described:

- "The SIFT feature detection and matching, that is used by the backend, is patented"
- "The detection and matching could be optimized e.g. by improving the matching or by using different technologies like CUDA"

Both these issues we try to solve by choosing a different matching, the process is shown in detail in the next chapter Project Setup.

Some suggestions and findings were rather simple to integrate or solve. For example concerns regarding the battery life, "At the moment the GPS localization is, used at maximal accuracy available", which drains battery life. As in our approach, we require the location only once, to find out against which sight we are running our matching, we just stop the service as there is no need to query location updates continuously.

Others posed us with problems. The drifting of content for example is a problem throughout many AR applications[1]. The problem lies in the nature of things, would go far beyond the scope of this work to fix and is more on the AR Library sites to solve and so in the

[1] https://github.com/google-ar/arcore-android-sdk/issues/225 (visited on 12/27/2020)

responsibility of Google, Apple, and the industry. As this is already happening [40], it seems to be far less a problem nowadays compared to two years ago, while it still can occur under certain conditions.

The main difference in our work is the approach we chose. While Tolstoi came from a games background, where it seemed to us the highest priority was to build a working prototype, we took a step back and first defined the requirements and abilities of our software.

3.2. Project Architecture and Motivations

When starting the project, we decided to design our architecture before and not after the project, following principles from "clean architecture: a craftsman's guide to software structure and design" [85]. For that we put up goals and figured out, how to reach them:

Backend and **Frontend** We went with a clear separation in frontend (UI) and backend (server). First, we defined the exact jobs for both. The server will have two tasks: Creating SfM models from images and storing them and second computing the pose of given a image and a GPS position. The UI will display AR content, that is created in advance. On top of that text-based information to the AR content must be able to be displayed and a possibility to switch between different AR overlays is required. Now the overall procedure is as following: The frontend will have a single button to initialize the localization process. Thereby it captures an image and sends it to the server. The communication will work over HTTP using a post request. The server expects three parameters for the pose estimation: an image, the rotation of the image, and a GPS location. In response, the server sends the name of the sight and the computed pose in Colmaps data format back to the frontend. In the frontend, the AR content is now placed at the received pose. This enables two additional buttons. One to open a window with textual information and one slider to select the AR content.

The split in *server-based localization* and only minor computations on the frontend , is not uncommon for localization on mobile devices [92].

Extensibility The scenario we had in mind when designing this goal was a new user, who wants to use the tool. We aim to assure that it is at easy as possible for this person to set up the project, understand the ideas and be able to extend it himself.

To enable this, we set a strong focus on high and persistent documentation, making reading and understanding code way easier [86]. Also, we followed the open-closed principle [89], which is the design principle equivalent for ensuring extensibility.

Next, we decided to choose simplicity over complexity if possible. So for example we used the JSON format for our HTTP communication instead of going with google protobuf[2], another method to serialize data, state of the art in modern architectures, but in some ways also an overkill for us. Since the speed we gain is neither the bottleneck nor worth the complexity we would have added.

Also, we want to provide a tutorial video, where one example project is set up and standard questions and errors are discussed.

[2]https://developers.google.com/protocol-buffers (visited on 12/06/2020)

Not to reinvent the wheel Here our focus was on using existing libraries that are maintained. As outlined in the section 2.8.2.8 due to advantages like deploying to multiple systems, integration of ARCore and Co "Unity is the leading platform to develop mixed reality experiences" [45]. So we chose to use Unity. But this also means going with the Unity version 2019.4[3] with long term support instead of the newest Unity Beta which may have some cool new features, but will most likely ensure trouble for someone attempting to run the project in a year from now.

In a weakened form, this also applies to the feature matching process. While this is a core function of our project, we don't build it from scratch, but instead with the help of a selection process, will pick an existing solution, that fulfills our requirements. Also in the previous work by Tolstoi, some features are implemented at high standards and it is no shame to reuse them.

We decided to build our backend using *Python*, instead of sticking with the previous approach off extending Colmap using the programming language *C*. Partly we argue here with the Single Responsibility Principle, also a design principle from Robert C. Martin [84]. In this case, the responsibility of Colmap is building the SfM model. Behaving as a server seems to have little to do with that. It also results in difficulties, when you try to exchange parts.

The backend will consist of two parts. On is the localization, this includes building the model as well as computing a pose, by matching a query image against a previously built model. The other part handles the HTTP services, where we considered using Nginx [108] or Apache [37] server, two of the most used high-performance web servers. We again decided that his might be overkill since for our defined API, a simple HTTP server in Python is well enough suited to do the job and if anytime in the future this might result in a bottleneck, it will be no trouble to exchange this component, thanks to proper encapsulation.

3.2.1. Motivations

The reason why we put such a strong focus on software architecture was first and foremost frustration. Since, when we started working in this field, trying to understand the architecture of the already existing projects cost incredible efforts – in fact, we mostly failed.

There were several reasons for this, the main ones being: zero comments in over 5.000 lines of code, almost no existing documentation, tons of commented-out code, "black-box" functions, where random matrix multiplications would happen, while miraculously some lines got negated.

A stand-out and frustrating example was the conversion from Colmaps quaternion to Unitys quaternion definition. Quaternions are a number system that extends the complex numbers and can be used for rotations, bringing some advantages, like not being susceptible to the "gimbal lock" [51]. At the same time they bring disadvantages, mainly being hard to grasp and debug, as they are four-dimensional or as Oliver Heaviside, a famous mathematician said: "the quaternion was not only not required, but was a positive evil of no inconsiderable magnitude" [53, p.134]. So we almost spend a week debugging and troubleshooting, till

[3] https://unity3d.com/unity/qa/lts-releases (visited on 12/06/2020)

we found this little trivial detail, that for some obscure reason Colmap is using a really unusual ordering, defining a quaternion as $(\mathbf{QW}, QX, QY, QZ)$, where everyone else uses $(QX, QY, QZ, \mathbf{QW})$. But in hindsight, the most annoying thing about all this was, that the whole situation could have been avoided so easily by a single comment as the previous team must have done this conversion somewhere as well. So we wanted to avoid this situation happening again at all costs and decided that therefore simplicity, clarity, and documentation are the most important aspects. And they all can be enforced by good software architecture.

The other reason is why we started from zero was, that we didn't agree with many fundamentals decisions that were made and didn't see it feasible to revert them, e.g extending on top of Colmap as opposed to building an independent backend.

3.2.2. General advantages of our approach

There are three major advantages of our solution.

- Open Source. When we started researching, we always checked the papers for a GitHub repository and often also searched manually if we can find a codebase for these projects. Off all projects mentioned in our Related Work about Cultural Heritage or Sightseeing an AR, only in a hand full off cases we were able to find anything. For example MauAR[88], a non academical project originated from a hackathon and was never maintained. But there is not a single project with a proper GitHub repository, well documented and maintained. Moreover, there is no possibility to reproduce the results, which makes all the work useless for scientific research and deprives us of the chance to built on or learn from their experience. We also utilized a website, where you can search for research with the code: "https://paperswithcode.com"[4], but we didn't find a single entry for search queries containing "Augmented Reality" in combination with "Cultural heritage", "Sightseeing" or similar terms. In contrast, when we build the computer vision part or choose our server, we had plenty of open source implementations available[5].

  So making our work open source and documenting it well, is a huge benefit for other researchers, who want to start a project them self and are looking for inspiration, examples, or parts to take up[6].

- Vision Based. As expounded before, almost all solutions for outdoor AR utilize mainly location based information, due to the limitations and struggles of computer vision algorithms with changing outdoor conditions. Since we were able to overcome these limitations, namely the lack in robustness through seasonal or weather changes, thanks to state of the art algorithms, we can now facilitate the advantage of a vision based solution. Compared to the existing tools, we have an exact pose to disposal and not

[4]https://paperswithcode.com/ (visited on 02/02/2021)

[5]Not publishing code is a problem for almost all areas in computer science, due to various reasons. This includes keeping advantages, commercial reasons or, as infamously claimed by OpenAI with their language model GPT-2, the code is too dangerous to be released. But, the extent to which publishing code is absent in this area, is unique.

[6]Link to our Repository: https://gitlab.com/KehnelP/ense (visited 02/10/2021)

only a maximal 2-meter accuracy, as with GPS solutions. This allows highly precise overlaying of the AR content, leading to a real immersive experience and a better user interaction. This gives us a unique ability for storytelling in front of sights, landmarks or a lieux de memoire[96].

- Content. For demonstrating the application, we decided to overlay images over the building, as this is the simpler solution compared to creating a proper 3D model with texture and colors of a complete building. However, we are not limited to images. Since we work with the pose of the user, without any effort, we can also switch and display a complete 3D model relative to the user, under the condition that we possess such a model.

4. Project Setup

While the previous chapter was a more theoretical part, in this chapter we will go into how we implemented the requirements in practice. Thereby we have once again three main aspects:

1. Choosing and adjusting a visual localization tool.
2. The development of the frontend.
3. The sight we choose and the stories we want to tell in our demo project.

4.1. Choosing an approach for visual localization

As explained in the section 2.7.2, Visual Localization estimates the exact pose of a query image relative to a 3D model.

In the previous work, a 3D model was created with Colmap and the standard and patented SIFT features. For the matching at run-time, the new image would be added to the Colmap model.

To improve the matching process and the model in regards to stability, performance, licensing, and run-time, we decided to inspect the top performers of the annual indoor/outdoor localization challenge at CVPR 2020[1]. It's a well-known project with state-of-the-art competition. The challenge is based on multiple datasets, where the reconstructed pose accuracy is evaluated. The evaluation process is based on a paper: "Benchmarking 6DOF Outdoor Visual Localization in Changing Condition" [114]

Visual localization for handheld devices challenge

Method	Aachen		InLoc	
	day	night	duc1	duc2
Hierarchical-Localization + SuperGlue	89.6 / 96.1 / 98.8	44.9 / 71.4 / 88.8	49.0 / 69.2 / 79.8	53.4 / 77.1 / 80.9
ONavi	85.7 / 93.7 / 98.9	48.0 / 71.4 / 88.8	41.9 / 68.2 / 84.3	50.4 / 76.3 / 80.2
Visual Localization Using Dense Semantic 3D Map And Hybrid Features	90.3 / 95.5 / 97.9	44.9 / 67.3 / 87.8	48.0 / 62.6 / 79.3	53.4 / 64.1 / 74.8
KAPTURE-R2D2-APGeM	88.7 / 95.8 / 98.8	44.9 / 62.2 / 85.7	21.7 / 37.4 / 54.5	23.7 / 41.2 / 54.2

Figure 4.1.: 2020, CVPR Leaderboard [16].

The long-term outdoor challenge seemed to be a perfect fit since we work with outdoor sights and want to ensure that a created model, can be used for more than a season. We inspected the leaderboard[2] of the subcategory "Visual localization for handheld devices

[1]https://sites.google.com/view/vislocslamcvpr2020/home (visited on 12/27/2020)
[2]https://www.visuallocalization.net/workshop/cvpr/2020/ (visited on 12/08/2020)

challenge" and checked out the papers and git repositories of the best performing projects.

To choose one of the approaches, we based the selection loosely on the following criteria:

1. The ranking in the challenge.
2. The number of issues in the GitHub repository and the average answer time, to see if the project is maintained.
3. A subjective impression of the project, gained by inspecting the codebase (comments, structure) and the read.me to see how easy it would later be to understand and integrate the project.
4. Also subjective, reading the associated papers if they exist.

We evaluated the top 5 entries, where only 3 actually had a GitHub repository, and overall the Hierarchical Localization Toolbox, by Paul-Edouard Sarlin made the best impression. By now it has over 30 issues, with a fast response time, new features are developed, the codebase is well documented and it was the winner of the challenge[3]. And it also claimed that the matching can run in real-time, on a modern GPU and so is suitable for real-time applications, which would be a huge advantage. The closest competition was the 4.th place Kapture project [58] from the Naver team, an open-source group from Korea.

4.2. Hierarchical Localization Toolbox

The work of Sarlin is based on three corner stones:

HFNet [111], a hierarchical localization approach [60] based on a CNN that simultaneously predicts local features and global descriptors for accurate 6-DoF localization. With this they achieve *fast runtime*, suitable for real-time applications, by using "the coarse-to-fine localization paradigm" [34, 102]. This means that the first "perform a global retrieval to obtain location hypotheses and only later match local features within those candidate places." [111]. Also, great *localization robustness* is attained via leveraging learned descriptors, a method they newly proposed.

Superpoint [29] are learned features, like LIFT [132], which for descriptor matching outperform popular hand-tuned representations [27], like SIFT [79]. Superpoint excels in terms of keypoint repeatability and descriptor matching, while also generating a sparse number of keypoints, thus reducing the number of keypoints that are matched, later on, further increasing the overall speed.

SuperGlue [112], a neural network builds on top of this, responsible for the feature matching. The important part, that makes the overall performance so impressive is the use of attention: "flexible context aggregation mechanism based on attention, enabling SuperGlue to reason about the underlying 3D scene and feature assignments jointly" [112] This results in

[3] In hindsight this was a smart choice. As 5 months after this decision was made, now while writing the thesis, the project has almost 1000 stars on GitHub and worked very well for our approach (See 8.Conclusion)

outperforming traditional heuristics and techniques and state of the art results, with matching in real-time on a modern GPU.

These three components lay the foundation of the Hierarchical Localization Toolbox and are used in this work as the backbone of the architecture. The general steps of the toolbox for building and localization are:[4]:

- Extract SuperPoint local features for all database and query images
- Build a reference 3D SfM model
  - Find covisible database images, with retrieval or a prior SfM model
  - Match these database pairs with SuperGlue
  - Triangulate a new SfM model with COLMAP
- Find database images relevant to each query, using retrieval
- Match the query images with SuperGlue
- Run the localization
- Visualize and debug

4.2.1. Adapted to our requirements

Four our work a few things stand out. The previously mentioned stability and robustness of the matching process holds for different weather or lightning conditions, as well as the change through the seasons as seen in the 4.2 figure below.

(a) Different view angles, portrait vs landscape.

(b) Day vs Night.

Figure 4.2.: Matching examples

But also different angles and positions, the rise of a construction site, or otherwise illuminated scenes achieved stable results as seen in Fig 4.3 and Fig 4.4. Also, the ability to run in real-time comes in handy, making the whole process enjoyable for the user.

As one can expect, these attributes always have to be balanced. So we adjusted certain parts when integrating the toolbox in our project.

[4]https://github.com/cvg/Hierarchical-Localization (visited on 01/17/2021)

Figure 4.3.: Construction side with a moved crane.

Figure 4.4.: Illuminated with stars over christmas time.

For initially **building the 3D model**, we use exhaustive matching, to simplify the process, which as a trade-off increases the runtime for building the model. This is something we find justifiable since there is no urgency to have the model build in the same hour. Exhaustively matching, just means we match all images against each other, instead of using some sort of image retrieval like NetVlad [6]] to match only similar images. This results in exponential run time and thereby ultimately setting a soft cap to the maximum number of images that are still feasible to use. Simultaneously the run time of the matching, for building the models as well as computing the pose, also depends on tuning the **hyper parameters**. Speed is gained by reducing the *number of keypoints*, adjusting the *nms_radius*, lowering the *keypoint_threshold*, decreasing the *max_keypoints* or the number of *sinkhorn iterations* [3]. All these changes are a double edged sword, as they all inevitable come with consequences. From lowering the accuracy, to less robustness, up to a complete breakdown. There is a complete research area on how to optimize hype parameters [17] using algorithms and search strategies [18], as well as tones of ways to measure the improvements, like precision and recall or benchmarks. "Tuning deep neural architectures to strike an optimal balance between accuracy and performance has been an area of active research for the last several years." [110]

Since this is a practical implementation and the parameters have already been tuned beforehand, we aim to understand what influence the parameters have and where we can cut corners, maybe making some scores a little less perfect but therefore gaining something else, like speed or simplicity. For example, we found that we could halve the number of keypoints, the number of points that are compared during matching, while not losing precision. Partly this is due to the overlay, that even with a perfect pose is not displayed a 100% fitting, as the content was maybe taken from another view angle and therefore is slightly distorted. Also, a slight displacement of 2-3cm still lets the AR content look real or isn't even seen.

The final factor, that we can influence to avoid bottlenecks, is *smart programming*. Simple measures like pre-loading the 3D SfM model to RAM, as well as more complicated solutions like using PyTorch JIT[5] can significantly increase the performance.

4.2.2. Handling images

We found that 150-200 images with our tuned parameters are a good setup, that will result in a well-formed 3D model, which matching being precise enough while also running in an acceptable time. The initial process of matching and building the model will then last between a day and an hour, depending on the hardware you are using and if you utilize a modern GPU or run it on the CPU. It can be noted that not requiring a graphics card is rather uncommon for applications that combine machine learning, computer vision and are supposed to run in real-time. Therefore having both options is a nice little advantage.

Another aspect that is worth mentioning is *image preprocessing*, as the toolbox has no tool for that. Maybe one of the most important tasks in computer vision, it is often dismissed as a boring job. During the whole process, a lot of information about the images has to be known and made available, like the camera intrinsics, or the image size and rotation. Also,

[5]JIT traces functions and tries to optimize them using just-in-time compilation. https://pytorch.org/docs/stable/generated/torch.jit.trace.html (visited on 12/28/2020)

Figure 4.5.: App Skatch created with figma.com (visited on 12/07/2020)

the images have to be in the right location, in the correct format. All this can and must be prepared in advance with caution and precision since it is very prone to errors, that are later hard to find.

4.3. Frontend

When we started working on the frontend, we first inspected the version from Tolstoi as well. We then decided to sketch our ideas for the application as seen in Fig 4.5, with the help of Figma a graphics editor and prototyping tool, to allow new ideas and not be narrowed down to the existing frame.

4.3.1. Basic features

We decided, that we would keep it simple as well. One initial button to start the localization process. The button would be enabled once a minimum map of the environment is created, which would usually be after a few seconds and helped to prevent later drifting of the content. Once the content is displayed, from a previously hidden drop-down menu the user could select which AR overlay should be displayed and another button would appear. This last button, the info button, when clicked, will display a textbox with information about the AR content. Clicking the button again would hide the textbox. Switching between content on the dropdown menu, when the Info is open, would also switch the displayed information. For both buttons, we selected pictograms, icons that convey their meanings through their pictures. We assumed an I for information and the classical GPS icon, for localization, are

meaningful enough.

Another point from the previous evaluation was the introduction of a photo feature, which could probably engage young people to use the app. We spend a lot of time discussing this feature, implemented it but in the end decided against it for various reasons.

4.3.2. Taking Selfies

As appealing as the idea might sound, taking images together with the augmented sight, goes hand in hand with a lot of complications. For this, to work we need to use depth information, like a depth map, as the human is supposed to be in the foreground.

Figure 4.6.: Pokemon "Selfies" with the Pokemon always in the foreground [106]

ArCore and ARKit are both working on occlusion features, but this is still more of a beta feature, as "People occlusion is supported on Apple A12 and later devices"[6], with the same going for ARCore, where it only works on a few devices. We still tried it, with medium success in most cases leading to the gate completely vanishing.

Besides, for the real seflie, we would need to switch to the front camera, another nontrivial task, as they are often inferior to the main camera and normally don't support AR applications. Still, this is a really interesting feature and something that will most likely be easily implemented in a year from now, with the rapid changes happening with AR technology.

For the sake of completeness, it is even now possible to achieve something similar to a selfie up to a certain point, as for example in Pokemon Go with the AR Snapshot. But what they are using is a little "cheating", as the Pokemon will always be in the foreground and you have to use the main camera. So a real selfie, with the Pokemon partly behind you is not possible at the moment. Sadly this is the precondition for us, as images with the Sendlinger gate in the foreground are pointless.

[6] ARKit Documentation: https://developer.apple.com/documentation/arkit/occluding_virtual_content_with_people (visited on 12/27/2020)

Figure 4.7.: Left images taken with the Iphone 12, the AR Object can be in the background, even so it gets clumpy sometimes. The right images, captured with the Pixel 3 don't support this function and the object moves to the foreground.

Finally with "View in 3D" a recent feature by Google it is even possible to have the AR object in the background. But it takes a significant amount of time in the beginning, up to 30 seconds, to calibrate the room, and also only the main camera can be used. Even so, the feature is impressive, it is neither too stable and can be clunky. Also, it depends once again on the device as seen in Fig 4.7. The full functionality was only available when using an iPhone 12, opposed to that on the Pixel 3 the object stayed in the front.

4.4. The Sendlinger Gate

The decision about a suitable sight was a hard one, since Munich has definitely no lack of interesting places. In the end it seemed to be a good idea to stay with a gate, and thus keep up the tradition of Tolstoi's work.

During medieval times four main gates occupied the entrances of Munich in all directions. The "Schwabinger Tor", the "Isar Tor", "the Neuhauser Tor", today called "Karlstor" and the "Sendlinger Tor". [71] To the left an image from the Sendlinger gate, as part of a city model from the year 1572 by Jakob Sandtner is depicted, what is most likely the earliest tradition of the gate. The Sendlinger gate maybe not as famous as some of his siblings in Munich, Germany, and the world, like the Victory Gate, the Brandenburg Gate tor, or the Arch of Constantine, but it is still full of history and has interesting stories to tell. The Sendlinger gate was first mentioned in a document 1319. It was part of the city

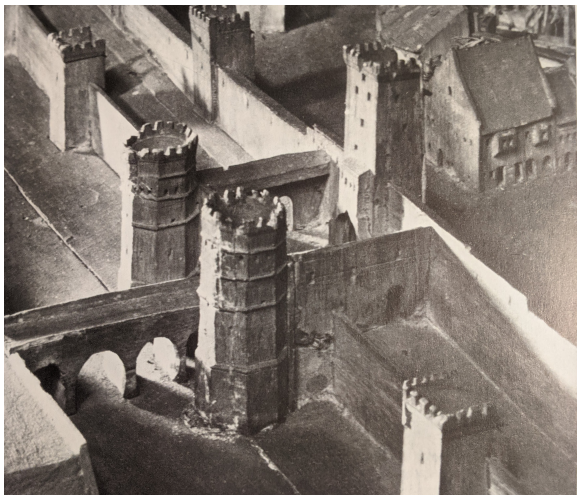


Figure 4.8.: 1572 Sendlinger Gate [71]

wall and has undergone many changes. We
decided to tell four different stories about the gate. We refer to the side facing towards the city as the front side, and the other as the back side.

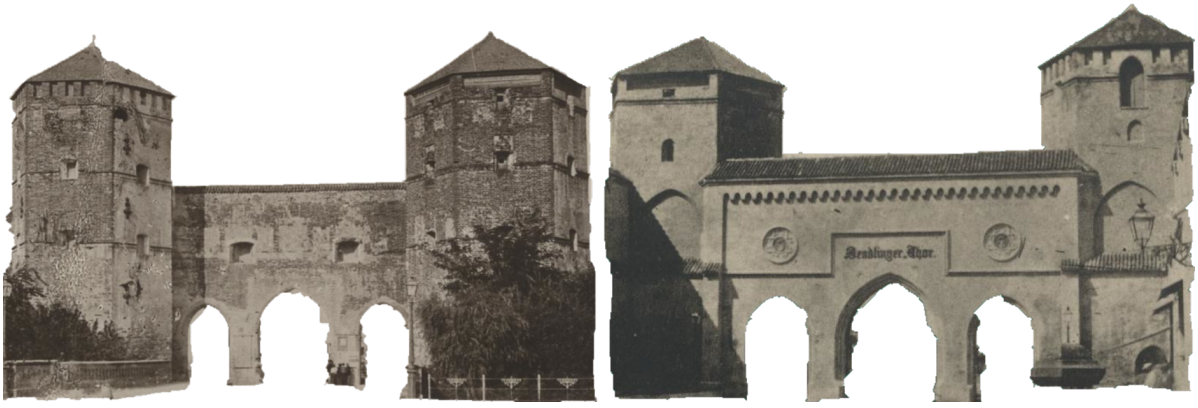


Figure 4.9.: Story 1: 1865 (left) the back and 1862 (right) redecorated front side of the Sendlinger gate

Story 1, Fig 4.9:
Chronological we start with the earliest. After the city gate had long lost all its defensive purposes, it had come down and was mainly something people complained about, since there not enough space for the traffic. Subsequently, in 1860 it was renovated and two smaller portals were added. Also, the old town side got redecorated according to medieval models, and the inscription: "Sendlinger Thor" was added, reminding everyone of the former purpose of the building.

Figure 4.10.: Story 2: 1910 (left) back side with 2 trams and 1906 (right) front side with only 1 tram line passing the Sendlinger gate

Story 2, Fig 4.10: Only a few years later in 1876, the horse streetcar on rails came to Munich, and already in 1892, an electric tramway was passing under the Sendlinger gate. The gate

was too narrow for two tracks, as seen from the city side image, which led to congestion and significant delays. So the calls for the demolition of the gate become louder and more urgent. This was a huge debate and historic preservation already played a role. "Luckily" the government declined all requests for demolition and insisted on a remodeling. 1906 the renewed conversion started, which resulted in the shape it kept, apart from small changes, till today.

Figure 4.11.: Story 3: A 2017(right) planned light installation and the destruction 1949 (right) after WW2

Stories 3 and 4 Fig4.11: The last stories tell from art and destruction. As most of Munich during the second world war, also the gate was hit by bombs and rebuilding the city afterwards was a slow process which went on for several years. Our final image tells an idea how the Sendlinger gate could have looked like. The image is the concept for a light installation that was planned in 2017, but never took place [43].

The stories are all based on the book "der Sendlinger-Tor-Platz in München" [71]. The images are from the "Stadtarchiv München", as well as the book. One of the main issues in creating the content is finding appropriate and interesting material. Here local archives and history books are the best places to start. The last story also shows some potential, that with this technology, you can not only retell stories but also bring a vision to life.

5. The Pipeline

In this chapter everything is put together. As explained in the introduction, the main goals of this project are to supply users with a tool, to be easy to use, easily extendable, and with as low a maintenance as possible. When talking about the complete process, we have to distinguish between two application levels. First, there is user 1, e.g. the curator of a museum or a cultural heritage agency, preparing a new sightseeing tour through his or her town. Here, the project has to be set up, a new model is created, and content prepared. We also call this step "the offline pipeline". This is the first and more sophisticated use, requiring a certain skill. It needs to be carried out in advance for each individual object, each monument or cultural heritage site.

The second user 2, is typically a tourist, the real-time user immediately interacting with the site. The tourist would be using the app on a portable device, giving background information on e.g. former states, functions and stories of the place and showing the AR content.

Both levels of usage are closely tied together since the app on the mobile device is constantly communicating with the model as set up by user 1 and running on a server.

We want to demonstrate the pipeline usage step by step, thereby explaining the core functionalities and show the implementation of our objectives. First, we will demonstrate the pipeline. We create the model for a new sight. Then we start the server with the new model added. Afterwards, on the application side, we import the model, add some content, and deploy the app. In addition, we created a video[1] explaining all the above steps in a visual fashion, which we highly recommend watching supplementary to the chapter.

Second, we explain the usage of the app. For a better visual demonstration we also created a video[2]. The interaction between both parts is covered in the section 5.2 as part of the offline pipeline.

The codebase for the complete project is accessible via GitLab[3].

[1]Link to tutorial video: https://youtu.be/gFo4LCvVha8 (visited on 02/08/2021)
[2]Demo video of the app ENSE: https://youtu.be/N2el-QiziO4 (visited on 02/08/2021)
[3]Link to our Repository: https://gitlab.com/KehnelP/ense (visited 02/10/2021)

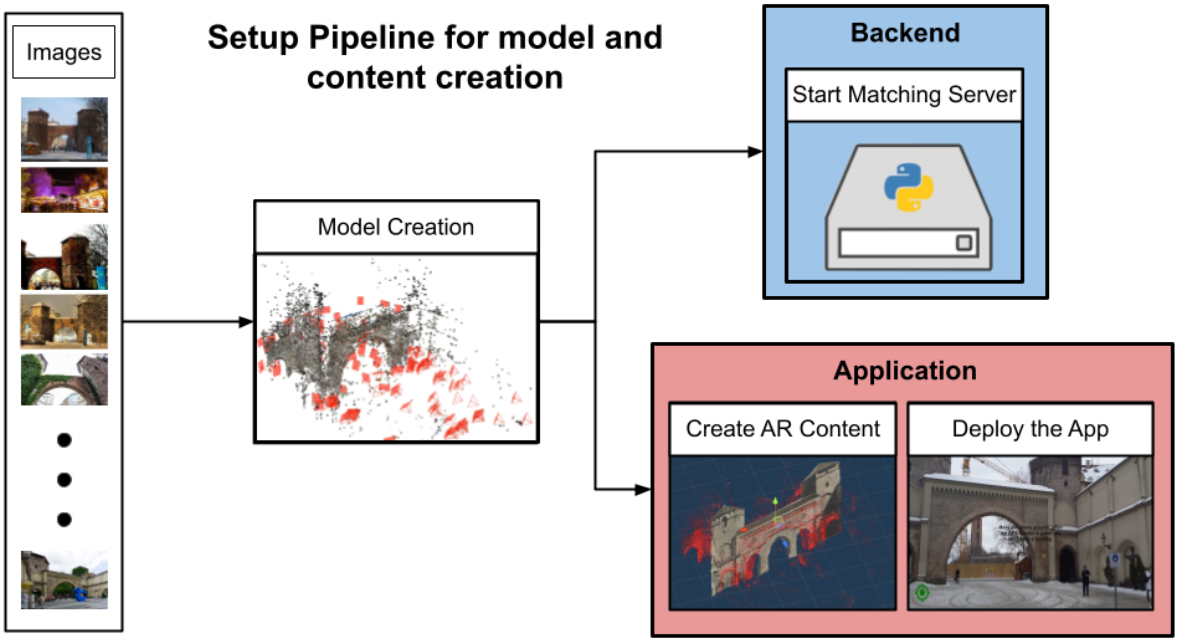


Figure 5.1.: With SfM, utilizing SuperPoint, SuperGlue and Colmap, a sparse 3D model is build from images. Next a HTTP server is started, using visual localization to match new query images against the created 3D model. Parallel AR content is created, fitted to the model and added to the application, which is then redeployed.

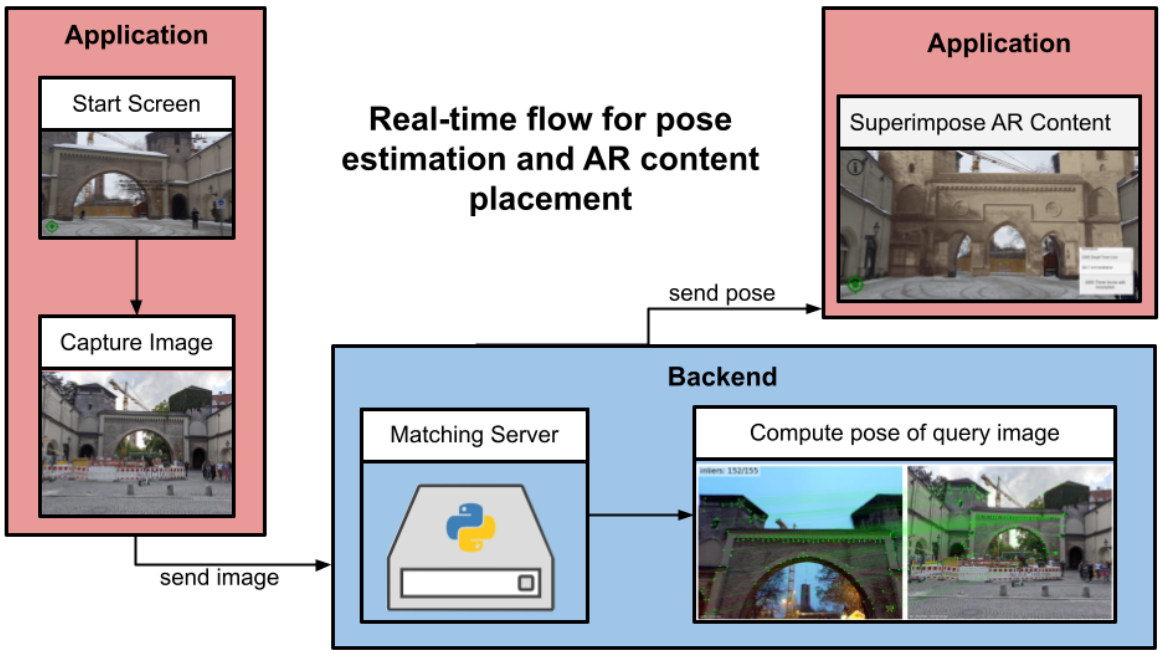


Figure 5.2.: On site, when using the app the localization process is started after pressing the GPS button. An image of the current view is captured and send to the server with a HTTP request. The server matches the image against the model and sends the computed pose back to the application. The AR content now is positioned in the real world relative to the pose, precisely overlaid over the building.

5.1. Creating the Model

To create a new model we run the script *create_model*. For the script to run successfully, we need to parse the mandatory argument *project_name*. This should point to the project directory in the datasets folder and is required to contain a subdirectory called images, where the images for the reconstruction are located (e.g *datasets/example_project/images*). There are no constraints for the format, size, or rotation of the images, however, for the reconstruction process, it is helpful to use pictures that were taken at different times and from different angles.

```
python3 create_model.py --project_name=example_project
```

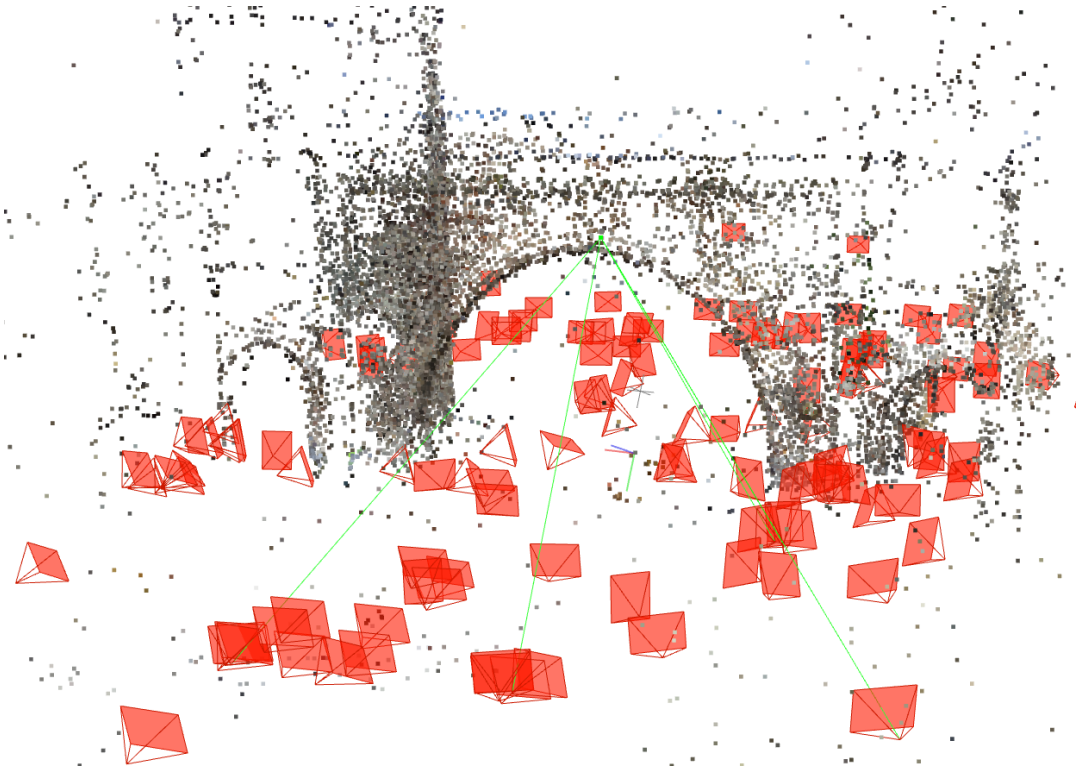


Figure 5.3.: Reconstructed sparse model of the Sendlinger Gate viewed in Colmap. The black dots are feature points, and the red planes are cameras. The selected feature point is connected to the camera images, where he got identified, via green lines.

If everything is setup correct, following steps are executed [111, 112]:

- Extract SuperPoint local features for all database images.
- Exhaustively match all image pairs with SuperGlue.
- Reconstruct a 3D SfM model.

- Triangulate the model with COLMAP.
- Store the output and create the directory structure for later localization.

This procedure is based on the *hierarchical localization toolbox*, the winner of the indoor/outdoor localization challenge at CVPR 2020[4] for more information check out the section 4.2 Hierarchical Localization Toolbox. Since we are running the matching exhaustively and are not using image retrieval to define the matching pairs like DIR [46] or NetVlad [6], the runtime increases exponential with every image added. For a stable result we recommend between 100-200 images, which will usually run in a few hours on a CPU, and in under an hour on a computer with a modern GPU. Although using more images results in a denser model, in our experience these models didn't perform better in the later localization process. In Fig 5.3 the result of a successful reconstruction is displayed.

Also, a set of standard configurations exists for the extraction and matching process. On default, our configuration *superpoint_real_time*, is selected for both. A variety of options exists and also own configurations can be added. Amongst other things, the model, preprocessing settings, and hyperparameters are defined there.

5.2. Running the server

Again a single script called *start_server* has to be executed. This will start an HTTPS server, that will listen on a specified port. When a post request from the AR App, containing a captured image of the sight and GPS coordinates reaches the server the localization process is started. First, via GPS the closest sight is selected. During the model creation, a GPS position for the new sight is created, by reading the EXIF data from the images and taking the average. This can also be adjusted manually. After a short preprocessing of the image, where the camera intrinsics and rotation is extracted, the pose of the camera, that captured the image will be reconstructed. This works similar to the model creation, by again extracting and matching features and then localizing the camera position.

```
python3 start_server.py
```

In the next step, the computed pose will be sent back to the App as the answer to the post request. Now on the frontend side, a few steps must be taken[5]:

- In colmap "the reconstructed pose of an image is specified as the projection from world to the camera coordinate system of an image using a quaternion (QW, QX, QY, QZ) and a translation vector (TX, TY, TZ)." [100]. In Unity, the quaternion is defined as $QX, QY, QZ, QW)$.

[4] `https://sites.google.com/view/vislocslamcvpr2020/homevisited12/6/2020)`

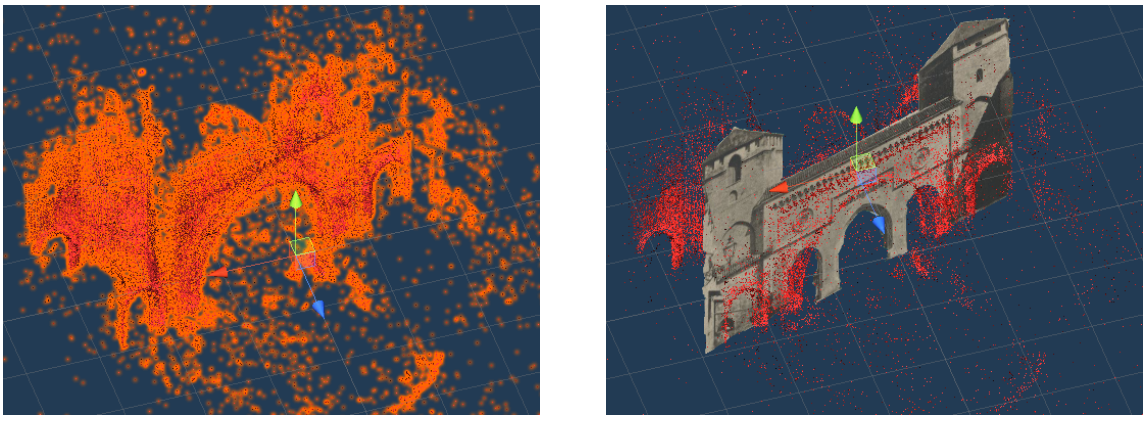
[5] To understand the described steps it can be helpful to go step by step through the equivalent lines of code: 1. *ColmapResultConstructor*, 2. *ConvertCoordinatesCOLMAPToUnity*, 3. *CreatePrefab* in AppManager.cs https://gitlab.com/KehnelP/ense/ (visited 02/08/2021)

- Colmap uses a right hand versus the left-hand coordinate system from Unity [134]. To convert between these two coordinate systems multiple steps are necessary. First, the negative of the position is multiplied with the inverse of the rotation, afterwards the y value of the position is negated. Secondly, the rotation is inverted, and then the x and z values of the quaternion are negated. This conversion can also be seen in other works like the COLIBRI VR Toolkit [33].
- As stated above the received answer contains a transition from the model origin to the camera pose. What we actually need is the reverse of that, meaning how to get from the camera to the model center point. Once we have that, this position has to be put relative to the camera position in Unity. Even more precisely, the position at time t, where t is the time when the image was captured.

Once we computed that final position, we can now spawn the selected prefab at this position.

5.3. Authoring content

In order for the application to show content for a model, we have to create content. Using Gimp or some other photo editing software, we then edit the images we want to overlay. They can be historical, or artistic, but also a contemporary image, where for example the whole building is just colored yellow works well. To avoid distortion, we recommend using images where the position of the camera relative to the building is central. During the project it has proven to be advantageous, to cut out the objects. Also when exporting the image you should use *.PNG*, since some other formats like *.JPG* don't support transparency.



(a) the model imported in unity

(b) content added to the model

Figure 5.4.: Creating content in Unity

Once the image is finished, the 3D Model can be imported as *.ply* file in the Unity project and turned into a new *prefab*. Now the desired content can be added as a child to the prefab.

Normally the detected features are easily seen in the image and with the help of them, the image can be resized and rotate to match the rotation and position in the real world. Finally a *text component* can be added to each image. If in the app a user presses the info button, the content of this entry is then displayed.

5.4. Deploying the App

Finally, the model has to be scaled, before it can be added to the model list. The reason for the scaling is known as *scale ambiguity* [55, p.61] [77]. The problem being that during the reconstruction process, at least without external knowledge, the model is only defined up to an unknown scale factor, which cannot be determined from images alone. To find the accurate scale, we have to know the size of our model in the real world or at least of some part of it, like a door. In unity a cube of the size *Vector(1,1,1)*, will match a cube of one cubic meter in the real world, so we can then use an object, defined with the correct dimensions, as a measurement tool.

When this is done, the last step is to drag and drop the object to the model list and the whole app is ready to be deployed to a device. A common mistake is renaming the model, though it is important that the model name matches the project name from Step 1, 5.1, Creating the Model.

5.5. Using the App

Now we are in part two, using the application. In front of the sight, we move the phone around so an initial map of the environment is created. After a few seconds, the locate button is enabled and we can press it to start the localization process. As mentioned above, the app takes a picture, sends it to the server and waits for the response, containing the current pose. As soon as the process is done, the various content we created earlier can be viewed and the additional information accessed. All steps can be retraced in Fig. 5.5 Furthermore, a demo video exists, which shows the complete process as well[6].

[6]Demo Video: https://youtu.be/N2el-QiziO4 (visited on 02/08/2021)

Figure 5.5.: Application Process. (Top Left) App started and a hint is displayed, (Top Right) after pressing the green button the localization process starts, (Middle Left) AR content is succesfully overlaid, (Middle Right) from the menu in the right corner different stories can be selected, (Bottom Left) A different story is selected and the user moved, (Bottom right) another story is selected and after the info button, top left corner, is pressed a text the infotext is displayed.

6. Evaluation

The goal of the evaluation is to answer our research questions and gather insights about the usability of the application, find out if the user liked the use of AR for sightseeing and how this approach performs compared to a classical approach of a tourist guide, showing images on paper.

To recap our research questions:

> *RQ: Is using this app superior to a classical sightseeing experience in front of a sight, regarding users learning and overall experience?*

To answer the question, a study was planned to be conducted with 30 participants, where each run should last for not more than 10 minutes. Due to special circumstances, see section 6.3, we could not conduct the study as planned.

6.1. Survey

For the evaluation of the application, we choose to use the System Usability Scale (**SUS**) survey [13], more precise the adapted version for non-native speakers, which modified item 8 due to the word "cumbersome" which apparently "English speakers failed to understand" [38].

We choose this approach for two reasons:

1. When we looked at comparisons "[t]he majority of the most used standardized usability questionnaires (e.g. SUMI, SUS, QUIS, CSUQ, etc.) covered general quality issues" [8]. The study further implies that, as expected, almost all of the most used questionnaires can be used and fulfill their job. It's rather personal preferences and use cases, where some differences can be found. For us, the number of questions was an important factor, since we wanted to engage random people on the street, who are already hard to convince to participate and have a short attention span, as they don't want to freeze outside in winter temperatures answering an endless amount of questions. So for example SUMI [69], with over 50 questions is not applicable, fur us. Another good candidate would have been the USE Questionnaire [81], but as it was not mentioned in the comparisons, we decided against it.

2. The decisive factor was, that the SUS survey was already used with success in the previous work, leading to good comparability.

Complementary to the standard SUS questions, we added five additional questions. We considered asking more precise questions about the stories, but as we didn't want the study to take longer than 10 minutes, 15 questions were the maximum we could justify.

The additional questions are:

- Compared to a City guide showing you the picture of the adjustments, how would you rate using this application. Here the answers go from "Way worse" to "Way better".
- Do you have the feeling, that you've learned something about the sight.
- Would you download this application from the AppStore.
- Could you imagine that this application is used for teaching.
- Which of the four stories did you like the most? Here the five answers are None, First, Second, Third, Fourth

If not written otherwise, they all follow the standard answer option from the survey, going from "Strongly Disagree" to "Strongly Agree". The surveys were created with Google Forms and are both included in the addenda.

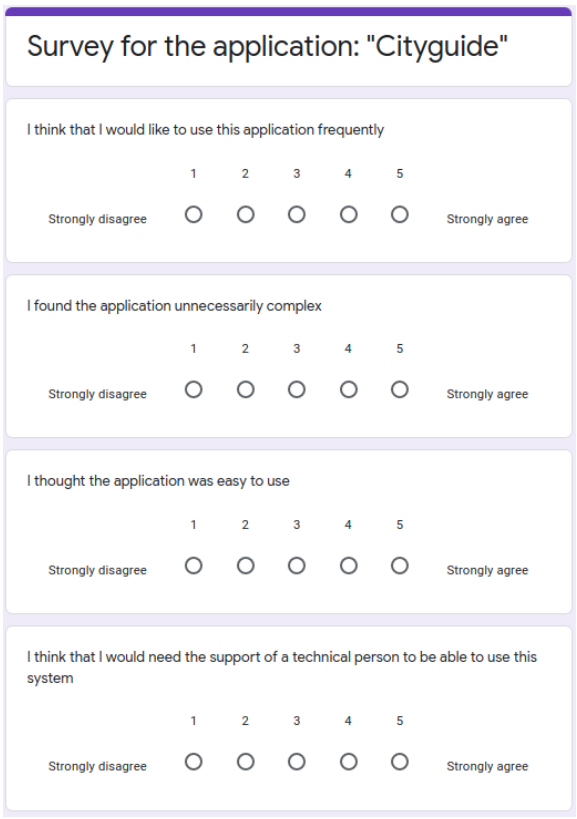


Figure 6.1.: Excerpt from the SUS Survey

6.2. Study Procedure

The study is conducted by a single person and carried out on the city side of the Sendlinger Gate. This posistion is chosen, due to construction on the subway station, severely limiting the available space on the other side.

The procedure is defined as followed:

1. Let the user fill out the pre-study Questionnaire.
2. Give a brief introduction to the topic and the project while explaining the concept of AR and overlaying images oversights, for the purpose of sightseeing.
3. Hand the device over to the participant and give the instruction to: "Open the app City Guide and try using it with the Sendlinger Gate"
4. Let the user fill out the modified SUS survey.
5. Ask for open questions and suggestions, they should be written done by the conductor.

During the whole process, the user is allowed to ask questions. All of them should be written down since they are valuable hints for ambiguity. During "Step 2", no further information should be given while the user is testing the application. But if questions occur regarding the usage of the app, they can be answered directly, other questions should be answered afterwards. Furthermore, the user should be pointed to the feature, where he can change the displayed timeline, in the unlikely case anyone will oversee it. The notice should only be given, when the participant wants to finish the testing. This has to be noted. At last, the conductor should observe, if the user starts walking around automatically or keeps standing in one place. Both questionnaires are filled out directly on the mobile phone. The instructor will open both surveys[1] in advance in separate browser tabs.

A sheet with all the instructions and a space to write down the notes is given to the conductor. In total, the process is planned to last approximately 10 minutes. One minute for step 1 the first survey. Then two minutes for step 2, the introduction. Again two minutes for testing in step 3. The second survey should last around three minutes and the final step, the open questions, and suggestions, is calculated with two minutes again.

6.3. Special Circumstances

In 2020 COVID-19 has been declared a pandemic by the World Health Organisation (WHO)[2]. During the year 2020 this lead to several lockdowns and curfews, which also had an impact on our evaluation, since it was not only harder to engage people on the street, but actually discouraged. Under these conditions, it was impossible to carry out the evaluation as planned. The only justifiable option we had, was to ask friends and coworkers to test the application for us and give some written or oral feedback. This resulted in an informal evaluation where the result is purely feedback based and includes the natural positive bias [118] friends and coworkers bring compared to randomly chosen participants.

6.4. Results and Feedback

In total 5 people (friends) used the application and gave feedback. Due to the informal nature of the feedback, as well as the low number of participants, no statistical analysis was performed.

Overall the feedback was positive, we did a shortlist of the most common reply's we got:

+ The diversity of the stories.
+ Actually useful application for AR and not just a "gimmick".
+ This would be amazing for a city guide, as it's way better to visualize

[1] Link to SUS Survey: https://forms.gle/k2fcNQAYnm4EvmfL9 (visited on 01/15/2021) and to the Pre-Study: https://forms.gle/4RRxWSdWdZfg8G7G (visited on 01/15/2021)7

[2] https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19—11-march-2020

+ Most inquiries and curiosity was about the light installation, with its colorful mentioned several times.
- The design got some critic as people criticized a lack in consistency and "you an see that the team, had more a computer science than a UX-Design background"
- Matching not perfect, drifting when moving the phone too fast.
- More features would be cool.

Surprisingly often people asked what will happen next if the app will be in the AppStore or any work with the city is planned.

Due to the low number of participants and the informal setup, no real conclusions can be drawn. We think the users were comfortable when using the app. The overall positive feedback also signals that this could definitely have potential.

7. Future Work

At this point in time a functional prototype exists, that fulfills all the requirements defined in the beginning and is capable of showcasing the potential with a basic use case. Still, there is a lot of work left, before the tool can be used in a day-to-day environment. If the project is continued, the most important issues, that need to be addressed immediately are:

- As the work so far can be seen as a general proof of concept, now a decision regarding the purpose and direction for the project has to be made and a clear use case defined. Is the next goal to use this for school classes and enhance the teaching experience or do we aim to target local city guides, to improve their tours, or is the target group just an ordinary tourist. A completely different direction could be developing this tool only for other researchers, maybe even without any computer science background, as the backbone for their projects. Depending on this decision, further development steps should be defined.
- Even more important is to fix the main weakness of the project, the total lack of any significant evaluation. Even so, the circumstances partly excuse this condition, any arguments and decisions made now, would be mainly based on pure speculation. If, for example, it turns out that tourists prefer to visit a city without using their phone, this would obviously have big consequences for the audience. So instead of developing the next step or adding more features, executing a study should have the highest priority.

  If the above idea of targeting other researchers is pursued, a different study would be advisable. Here the focus should be figuring out if the setup is already easy enough to use, what the typical sources for errors are and which parts need to be reworked. A research question could go in the direction of: "Can a user without a computer science background, only with the documentation and demo video get a result in a certain time."

Yet, there are issues distributed over the whole project, that certainly can be improved or need further work.

7.1. Backend

The backend is currently in a healthy state and most changes would be cosmetic or end in over-engineering. Minor changes, that should be considered when moving forward with the project, are:

- One part is to add 2-3 more sights, to see if this has any impact on the overall performance or stability. This also makes the application an overall better prototype.
- The backend server currently needs around 3 seconds to localize a pose. As explained, this can be improved by the pure use of better hardware, as well as further optimizing and tuning the hyperparameters. However, what would be more important, is to enable handling parallel requests, by using threading or other solutions. Otherwise, this will be a clear bottleneck when trying to scale the project.
- In a year from now, with the rapid development happening in AR and the mobile world, the backend should be reevaluated and maybe the computation can be moved to the phone itself and only content has to be downloaded from a server.

7.2. Frontend

The frontend, a little less optimized than the backend, has mainly issues regarding the design. Or as feedback said: " You can see that there was no UX designer in your team."

- First a more consistent style between the UI elements would be nice. Also currently text-based information, like the hint, is not always easy to read. So increasing the readability of text-based information, would also be desirable.
- "To zoom or not to zoom" - Scaling lets the user increase or decrease the size of an object and in mobile applications is most often performed with a pinch gesture.[59]. We decided at the beginning of the project, that zooming does not fit the concept of AR, as the content is designed to fit the real world. This decision was maybe questionable, and if many people expect and inquire about this feature, it might have to be reevaluated.
- Integrating an improved procedure to add content seems desirable, as this is currently the least optimized step in the pipeline. Maybe there exists a third-party tool, or an existing toolkit, as Colibri VR[1], can be converted to get this job done.
- Finally, there are some details to be noted about the models. Currently, there is no simple solution for finding the correct world-scale off the model. As this is a typical problem with 3D models, different solution approaches exist, but they would need to be evaluated and integrated. In the long run, also the size of the app is something to keep in mind, where 3D models and content are factors of influence. As of now, the application has a size of under 100MB and a single model with content requires under 10MB. So this problem is something to keep in mind but does not pose an immediate threat.

[1] An Open-Source Toolkit to Render Real-World Scenes in Virtual Reality: https://caor-mines-paristech.github.io/colibri-vr/ (visited on 01/09/2021)

8. Conclusion

Overall our tool performed very well and we are happy with the results. Our motivation was the lack of any existing foss solution for overlaying AR content on outdoor buildings. With our solution, we were able to remedy this problem.

We stuck to the guidelines, which we imposed on ourselves in the beginning: *Simplicity, Extensibility, Reusability* and it paid of. Two prime examples that can be directly attributable to them are: First, in the final stage of our development, we had to change a specific feature of the localization pipeline. This was easily implemented without the need to rewrite major parts of our code, as we could simply exchange the responsible function and only change at one position where required. Second, selecting the Hierarchical Localization Toolbox, for our visual localization process, was a direct consequence of *simplicity & extensibility*, analyzing existing solutions and choosing the best fit instead of building everything yourself. The project has now over 1000 stars on GitHub, which can be seen as a seal of quality in the computer science world, and worked excellently in the scope of our work.

Furthermore, we, unfortunately, could not conduct our study as planned. But we have done our best to set everything up and prepare the study, so once the circumstances go back to normal the study can be conducted without any further preparation. As a consequence, we were unable to answer our research question, based on data. From our informal evaluation, we conclude, that the project has potential as people enjoyed using it and valued the experience.

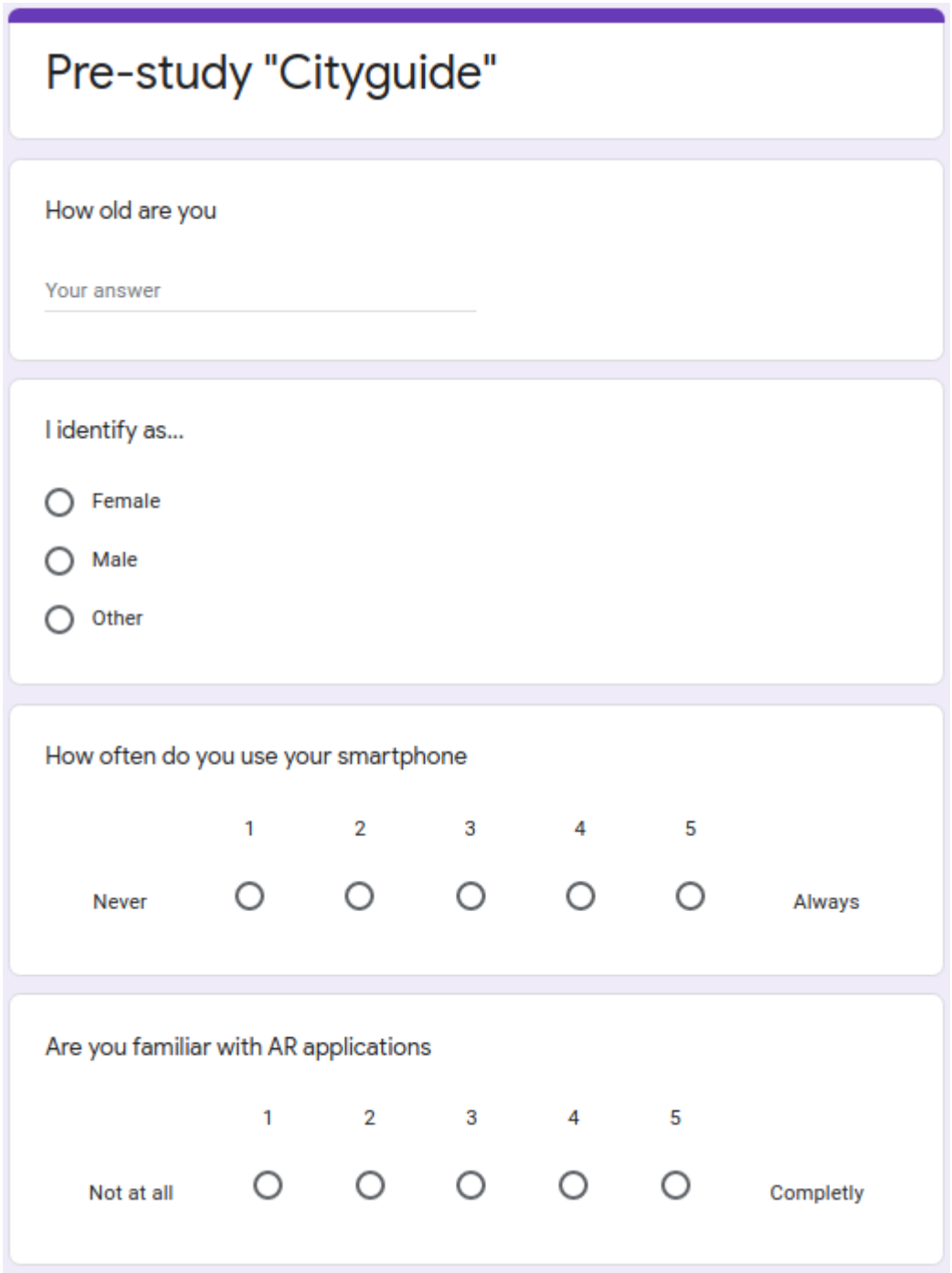
To sum up the most essential results: we build a good prototype and hope this work, or parts of it, will be used by other researchers and help them with their works. We hope that providing our tutorial and documentation results in a good user experience for them.

The overall results suggest a promising and enriching approach for cultural heritage projects. We therefore plan publication in ISMAR[1] or iLRN[2] once a proper evaluation will have been added.

[1]https://ismar21.org/ (visited on 01/010/2021)
[2]https://immersivelrn.org/ilrn2021/(visited on 01/10/2021)

A. General Addenda

A.1. Prestudy & SUS Survey



Pre-study "Cityguide"

How old are you

Your answer

I identify as...

- Female
- Male
- Other

How often do you use your smartphone

	1	2	3	4	5	
Never						Always

Are you familiar with AR applications

	1	2	3	4	5	
Not at all						Completly

Survey for the application: "Cityguide"

I think that I would like to use this application frequently

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

I found the application unnecessarily complex

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

I thought the application was easy to use

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

I think that I would need the support of a technical person to be able to use this system

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

I found the various functions in this application were well integrated

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

I thought there was too much inconsistency in this application

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

I would imagine that most people would learn to use this application very quickly

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

I found the system very cumbersome/awkward to use

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

I felt very confident using the system

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

I needed to learn a lot of things before I could get going with this application

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

Compared to a City guide showing you the picture of the adjustments, how would you rate using this application

	1	2	3	4	5	
Way worse	○	○	○	○	○	Way better

Do you have the feeling, that you've learned something about the sight

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

Would you consider downloading this application form the appstore

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

Could you imagine that this application is used for teaching

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

Which of the four stories did you like the most?

- [ ] First Story
- [ ] Second Story
- [ ] Third Story
- [ ] Fourth Story
- [ ] None

List of Figures

Glossary

AR Augmented Reality.

CH Cultural Heritage.

CV Computer Vision.

ENSE Enhance Sightseeing.

FOSS Free and Open Source.

GPS Global Positioning System.

HHD Hand-Held Display.

HMD Head-Mounted Display.

SfM structure from motion.

SIFT Scale Invariant Feature Transform.

SLAM Simultaneous Localization and Mapping.

SUS System Usability Scale.

Bibliography

[1] C. van Aart, B. Wielinga, and W. R. van Hage. "Mobile Cultural Heritage Guide: Location-Aware Semantic Search". en. In: *Knowledge Engineering and Management by the Masses*. Ed. by P. Cimiano and H. S. Pinto. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2010, pp. 257–271. ISBN: 978-3-642-16438-5. DOI: 10.1007/978-3-642-16438-5_18.

[2] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. "Building Rome in a day". In: *2009 IEEE 12th International Conference on Computer Vision*. ISSN: 2380-7504. Sept. 2009, pp. 72–79. DOI: 10.1109/ICCV.2009.5459148.

[3] J. Altschuler, J. Niles-Weed, and P. Rigollet. "Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration". In: *Advances in neural information processing systems*. 2017, pp. 1964–1974.

[4] J. Amakawa and J. Westin. "New Philadelphia: using augmented reality to interpret slavery and reconstruction era historical sites". In: *International Journal of Heritage Studies* 24 (Nov. 2017), pp. 1–17. DOI: 10.1080/13527258.2017.1378909.

[5] *AR: Experimenting with Vuforia Object Recognition on Android*. en. URL: https://sigma.software/about/media/ar-experimenting-vuforia-object-recognition-android (visited on 09/09/2020).

[6] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. "NetVLAD: CNN architecture for weakly supervised place recognition". In: *arXiv:1511.07247 [cs]* (May 2016). arXiv: 1511.07247. URL: http://arxiv.org/abs/1511.07247 (visited on 09/17/2020).

[7] R. Arandjelović and A. Zisserman. "DisLocation: Scalable descriptor distinctiveness for location recognition". In: *Asian Conference on Computer Vision*. Springer, 2014, pp. 188–204.

[8] A. Assila, H. Ezzedine, et al. "Standardized usability questionnaires: Features and quality focus". In: *Electronic Journal of Computer Science and Information Technology: eJCIST* 6.1 (2016). Publisher: Electronic Journal of Computer Science and Information Technology (eJCSIT).

[9] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. "Recent advances in augmented reality". In: *IEEE Computer Graphics and Applications* 21.6 (Nov. 2001). Conference Name: IEEE Computer Graphics and Applications, pp. 34–47. ISSN: 1558-1756. DOI: 10.1109/38.963459.

[10] R. Azuma. "Tracking requirements for augmented reality". In: *Communications of the ACM* 36.7 (1993). Publisher: ACM New York, NY, USA, pp. 50–51.

[11] R. T. Azuma. "A Survey of Augmented Reality". In: *Presence: Teleoperators and Virtual Environments* 6.4 (Aug. 1997). Publisher: MIT Press, pp. 355–385. DOI: 10.1162/pres.1997.6.4.355. URL: https://doi.org/10.1162/pres.1997.6.4.355 (visited on 08/26/2020).

[12] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk. "Learning local feature descriptors with triplets and shallow convolutional neural networks." In: *Bmvc*. Vol. 1. Issue: 2. 2016, p. 3.

[13] A. Bangor, P. T. Kortum, and J. T. Miller. "An empirical evaluation of the system usability scale". In: *Intl. Journal of Human–Computer Interaction* 24.6 (2008). Publisher: Taylor & Francis, pp. 574–594.

[14] H. Bay, T. Tuytelaars, and L. Van Gool. "Surf: Speeded up robust features". In: *European conference on computer vision*. Springer, 2006, pp. 404–417.

[15] M. K. Bekele, R. Pierdicca, E. Frontoni, E. S. Malinverni, and J. Gain. "A Survey of Augmented, Virtual, and Mixed Reality for Cultural Heritage". In: *Journal on Computing and Cultural Heritage* 11.2 (Mar. 2018), 7:1–7:36. ISSN: 1556-4673. DOI: 10.1145/3145534. URL: http://doi.org/10.1145/3145534 (visited on 08/26/2020).

[16] *Benchmarking Long-term Visual Localization*. URL: https://www.visuallocalization.net/workshop/eccv/2020/ (visited on 02/01/2021).

[17] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. "Algorithms for hyper-parameter optimization". In: *Advances in neural information processing systems* 24 (2011), pp. 2546–2554.

[18] J. Bergstra and Y. Bengio. "Random search for hyper-parameter optimization". In: *The Journal of Machine Learning Research* 13.1 (2012). Publisher: JMLR. org, pp. 281–305.

[19] S. Bianco, G. Ciocca, and D. Marelli. "Evaluating the performance of structure from motion pipelines". In: *Journal of Imaging* 4.8 (2018). Publisher: Multidisciplinary Digital Publishing Institute, p. 98.

[20] S. Blanco-Pons, B. Carrión-Ruiz, M. Duong, J. Chartrand, S. Fai, and J. L. Lerma. "Augmented Reality Markerless Multi-Image Outdoor Tracking System for the Historical Buildings on Parliament Hill". In: (2019). DOI: 10.3390/SU11164268.

[21] J. Carmigniani and B. Furht. "Augmented reality: an overview". In: *Handbook of augmented reality* (2011). Publisher: Springer, pp. 3–46.

[22] J. Challenor and M. Ma. "A Review of Augmented Reality Applications for History Education and Heritage Visualisation". en. In: *Multimodal Technologies and Interaction* 3.2 (June 2019). Number: 2 Publisher: Multidisciplinary Digital Publishing Institute, p. 39. DOI: 10.3390/mti3020039. URL: https://www.mdpi.com/2414-4088/3/2/39 (visited on 12/14/2020).

[23] Y.-L. Chang, H.-T. Hou, C.-Y. Pan, Y.-T. Sung, and K.-E. Chang. "Apply an augmented reality in a mobile guidance to increase sense of place for heritage places". In: *Journal of Educational Technology & Society* 18.2 (2015). Publisher: JSTOR, pp. 166–178.

[24] K.-H. Cheng and C.-C. Tsai. "Affordances of augmented reality in science learning: Suggestions for future research". In: *Journal of science education and technology* 22.4 (2013). Publisher: Springer, pp. 449–462.

[25] K.-H. Cheng and C.-C. Tsai. "The interaction of child–parent shared reading with an augmented reality (AR) picture book and parents' conceptions of AR learning". In: *British Journal of Educational Technology* 47.1 (2016). Publisher: Wiley Online Library, pp. 203–222.

[26] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio. "Attention-based models for speech recognition". In: *Advances in neural information processing systems* 28 (2015), pp. 577–585.

[27] C. B. Choy, J. Gwak, S. Savarese, and M. Chandraker. "Universal Correspondence Network". In: *arXiv:1606.03558 [cs]* (Oct. 2016). arXiv: 1606.03558. URL: http://arxiv.org/abs/1606.03558 (visited on 12/15/2020).

[28] P. Daponte, L. De Vito, F. Picariello, and M. Riccio. "State of the art and future developments of the Augmented Reality for measurement applications". en. In: *Measurement* 57 (Nov. 2014), pp. 53–70. ISSN: 0263-2241. DOI: 10.1016/j.measurement.2014.07.009. URL: http://www.sciencedirect.com/science/article/pii/S0263224114003054 (visited on 09/08/2020).

[29] D. DeTone, T. Malisiewicz, and A. Rabinovich. "SuperPoint: Self-Supervised Interest Point Detection and Description". en. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Salt Lake City, UT, USA: IEEE, June 2018, pp. 337–33712. ISBN: 978-1-5386-6100-0. DOI: 10.1109/CVPRW.2018.00060. URL: https://ieeexplore.ieee.org/document/8575521/ (visited on 08/17/2020).

[30] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[31] H. K. Dhonju, W. Xiao, V. Sarhosis, J. P. Mills, S. Wilkinson, Z. Wang, L. Thapa, and U. S. Panday. "FEASIBILITY STUDY OF LOW-COST IMAGE-BASED HERITAGE DOCUMENTATIONIN NEPAL". en. In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII-2/W3 (Feb. 2017), pp. 237–242. ISSN: 2194-9034. DOI: 10.5194/isprs-archives-XLII-2-W3-237-2017. URL: https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-2-W3/237/2017/ (visited on 01/18/2021).

[32] M. C. tom Dieck and T. H. Jung. "Value of augmented reality at cultural heritage sites: A stakeholder approach". In: *Journal of Destination Marketing & Management* 6.2 (2017). Publisher: Elsevier, pp. 110–117.

[33] G. D. de Dinechin and A. Paljic. "From Real to Virtual: An Image-Based Rendering Toolkit to Help Bring the World Around Us Into Virtual Reality". In: *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 2020, pp. 348–353.

[34] M. Ding, Z. Wang, J. Sun, J. Shi, and P. Luo. "CamNet: Coarse-to-Fine Retrieval for Camera Re-Localization". en. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South): IEEE, Oct. 2019, pp. 2871–2880. ISBN: 978-1-72814-803-8. DOI: 10.1109/ICCV.2019.00296. URL: https://ieeexplore.ieee.org/document/9008579/ (visited on 12/15/2020).

[35] M. Dunleavy, C. Dede, and R. Mitchell. "Affordances and limitations of immersive participatory augmented reality simulations for teaching and learning". In: *Journal of science Education and Technology* 18.1 (2009). Publisher: Springer, pp. 7–22.

[36] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler. "D2-Net: A Trainable CNN for Joint Detection and Description of Local Features". In: *arXiv:1905.03561 [cs]* (May 2019). arXiv: 1905.03561. URL: http://arxiv.org/abs/1905.03561 (visited on 01/18/2021).

[37] R. T. Fielding and G. Kaiser. "The Apache HTTP server project". In: *IEEE Internet Computing* 1.4 (1997). Publisher: IEEE, pp. 88–90.

[38] K. Finstad. "The system usability scale and non-native English speakers". In: *Journal of usability studies* 1.4 (2006). Publisher: Usability Professionals' Association Bloomingdale, IL, pp. 185–188.

[39] P. Fraga-Lamas, T. M. Fernandez-Carames, O. Blanco-Novoa, and M. A. Vilar-Montesinos. "A review on industrial augmented reality systems for the industry 4.0 shipyard". In: *Ieee Access* 6 (2018). Publisher: IEEE, pp. 13358–13375.

[40] J. S. Friesenhahn, J. A. Crews, and T. D. Schleiss. *Drift correction for industrial augmented reality applications*. Google Patents, Feb. 2020.

[41] H. Fukada, T. Funaki, M. Kodama, N. Miyashita, and S. Ohtsu. "Proposal of tourist information system using image processing-based augmented reality". In: *Proceedings of the 2011th Special Interest Group on Information Systems (SIG-IS) of Information Processing Society of Japan, Tokyo, Japan* (2011), pp. 14–15.

[42] D. Gálvez-López and J. D. Tardos. "Bags of binary words for fast place recognition in image sequences". In: *IEEE Transactions on Robotics* 28.5 (2012). Publisher: IEEE, pp. 1188–1197.

[43] A. Germany. *Das Sendlinger Tor wird bunt: Farbenfrohes für Nachtschwärmer*. de. Section: Das Sendlinger Tor wird bunt: Farbenfrohes für Nachtschwärmer. Aug. 2017. URL: https://www.abendzeitung-muenchen.de/muenchen/stadtviertel/das-sendlinger-tor-wird-bunt-farbenfrohes-fuer-nachtschwaermer-art-541798 (visited on 12/12/2020).

[44] J. Glover. *Unity 2018 augmented reality projects: build four immersive and fun AR applications using ARKit, ARCore, and Vuforia*. Packt Publishing Ltd, 2018.

[45] J. Glover and J. Linowes. *Complete Virtual Reality and Augmented Reality Development with Unity: Leverage the power of Unity and become a pro at creating mixed reality applications*. Packt Publishing Ltd, 2019.

[46] A. Gordo, J. Almazan, J. Revaud, and D. Larlus. "End-to-end Learning of Deep Visual Representations for Image Retrieval". In: *arXiv:1610.07940 [cs]* (May 2017). arXiv: 1610.07940. URL: http://arxiv.org/abs/1610.07940 (visited on 12/14/2020).

[47] F. Guimaraes, M. Figueiredo, and J. Rodrigues. "Augmented Reality and Storytelling in heritage application in public gardens: Caloust Gulbenkian Foundation Garden". en. In: *2015 Digital Heritage*. Granada: IEEE, Sept. 2015, pp. 317–320. ISBN: 978-1-5090-0254-2. DOI: 10.1109/DigitalHeritage.2015.7413891. URL: http://ieeexplore.ieee.org/document/7413891/ (visited on 12/14/2020).

[48] D. Gunning. "Explainable artificial intelligence (xai)". In: *Defense Advanced Research Projects Agency (DARPA), nd Web* 2.2 (2017).

[49] M. Haahr. *Creating Location-Based Augmented-Reality Games for Cultural Heritage*. Pages: 318. Nov. 2017. ISBN: 978-3-319-70110-3. DOI: 10.1007/978-3-319-70111-0_29.

[50] D.-I. Han, T. Jung, and A. Gibson. "Dublin AR: Implementing Augmented Reality (AR) in Tourism". In: *Information and Communication Technologies in Tourism 2014*. Journal Abbreviation: Information and Communication Technologies in Tourism 2014. Jan. 2014, pp. 511–523. ISBN: 978-3-319-03972-5. DOI: 10.1007/978-3-319-03973-2_37.

[51] A. J. Hanson. "Visualizing quaternions". In: *ACM SIGGRAPH 2005 Courses*. 2005, 1–es.

[52] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[53] O. Heaviside. *Electromagnetic theory*. Vol. 237. American Mathematical Soc., 2003.

[54] F. Herpich, R. L. M. Guarese, and L. M. R. Tarouco. "A Comparative Analysis of Augmented Reality Frameworks Aimed at the Development of Educational Applications". en. In: *Creative Education* 08.09 (July 2017). Number: 09 Publisher: Scientific Research Publishing, p. 1433. DOI: 10.4236/ce.2017.89101. URL: http://www.scirp.org/journal/PaperInformation.aspx?PaperID=77994&#abstract (visited on 09/08/2020).

[55] A. Heyden and M. Pollefeys. "MULTIPLE VIEW GEOMETRY". en. In: . *Projective Geometry* (), p. 63.

[56] T. Höllerer, S. Feiner, T. Terauchi, G. Rashid, and D. Hallaway. "Exploring MARS: developing indoor and outdoor user interfaces to a mobile augmented reality system". en. In: *Computers & Graphics* 23.6 (Dec. 1999), pp. 779–785. ISSN: 0097-8493. DOI: 10.1016/S0097-8493(99)00103-X. URL: http://www.sciencedirect.com/science/article/pii/S009784939900103X (visited on 08/31/2020).

[57] W. Huang, M. Sun, and S. Li. "A 3D GIS-based interactive registration mechanism for outdoor augmented reality system". en. In: *Expert Systems with Applications* 55 (Aug. 2016), pp. 48–58. ISSN: 09574174. DOI: 10.1016/j.eswa.2016.01.037. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417416000609 (visited on 08/17/2020).

[58] M. Humenberger, Y. Cabon, N. Guerin, J. Morat, J. Revaud, P. Rerole, N. Pion, C. d. Souza, V. Leroy, and G. Csurka. *Robust Image Retrieval-based Visual Localization using Kapture*. _eprint: 2007.13867. 2020.

[59] *Introduction - Augmented Reality Design Guidelines*. Section: Introduction - Augmented Reality Design Guidelines. URL: https://designguidelines.withgoogle.com/ar-design/augmented-reality-design-guidelines/introduction.html (visited on 09/10/2020).

[60] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. "From Structure-from-Motion Point Clouds to Fast Location Recognition". en. In: (), p. 8.

[61] A. Javornik, E. Kostopoulou, Y. Rogers, A. Schieck, P. Koutsolampros, A. Moutinho, and S. Julier. "An experimental study on the role of augmented reality content type in an outdoor site exploration". In: *Behav. Inf. Technol.* (2019). DOI: 10.1080/0144929X.2018.1505950.

[62] R. Johnson and J. Vlissides. "Design patterns". In: *Elements of Reusable Object-Oriented Software Addison-Wesley, Reading* (1995).

[63] M. E. Joorabchi, A. Mesbah, and P. Kruchten. "Real Challenges in Mobile App Development". In: *2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement*. ISSN: 1949-3789. Oct. 2013, pp. 15–24. DOI: 10.1109/ESEM.2013.9.

[64] H. Jun, B. Ko, Y. Kim, I. Kim, and J. Kim. "Combination of multiple global descriptors for image retrieval". In: *arXiv preprint arXiv:1903.10663* (2019).

[65] T. H. Jung and M. C. tom Dieck. "Augmented reality, virtual reality and 3D printing for the co-creation of value for the visitor experience at cultural heritage places". In: *Journal of Place Management and Development* (2017). Publisher: Emerald Publishing Limited.

[66] T. Jung, N. Chung, and M. C. Leue. "The determinants of recommendations to use augmented reality technologies: The case of a Korean theme park". en. In: *Tourism Management* 49 (Aug. 2015), pp. 75–86. ISSN: 02615177. DOI: 10.1016/j.tourman.2015.02.013. URL: https://linkinghub.elsevier.com/retrieve/pii/S0261517715000576 (visited on 12/13/2020).

[67] J. Keil, M. Zollner, M. Becker, F. Wientapper, T. Engelke, and H. Wuest. "The House of Olbrich—An augmented reality tour through architectural history". In: *2011 IEEE International Symposium on Mixed and Augmented Reality-Arts, Media, and Humanities*. IEEE, 2011, pp. 15–18.

[68] A. Kendall, M. Grimes, and R. Cipolla. "Posenet: A convolutional network for real-time 6-dof camera relocalization". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2938–2946.

[69] J. Kirakowski and M. Corbett. "SUMI: The software usability measurement inventory". In: *British journal of educational technology* 24.3 (1993). Publisher: Wiley Online Library, pp. 210–212.

[70] G. Klein and D. Murray. "Parallel Tracking and Mapping for Small AR Workspaces". en. In: *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. Nara, Japan: IEEE, Nov. 2007, pp. 1–10. ISBN: 978-1-4244-1749-0. DOI: 10.1109/ISMAR.2007.4538852. URL: http://ieeexplore.ieee.org/document/4538852/ (visited on 01/13/2021).

[71] H. Lehmbruch. *Der Sendlinger-Tor-PLatz in München: Eine Chronik in Bildern*. Kreissparkasse München, 1988.

[72] D. Li and W. G. J. Halfond. "An investigation into energy-saving programming practices for Android smartphone app development". en. In: *Proceedings of the 3rd International Workshop on Green and Sustainable Software - GREENS 2014*. Hyderabad, India: ACM Press, 2014, pp. 46–53. ISBN: 978-1-4503-2844-9. DOI: 10.1145/2593743.2593750. URL: http://dl.acm.org/citation.cfm?doid=2593743.2593750 (visited on 09/08/2020).

[73] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. "Worldwide pose estimation using 3d point clouds". In: *European conference on computer vision*. Springer, 2012, pp. 15–29.

[74] Y. Li, N. Snavely, and D. P. Huttenlocher. "Location recognition using prioritized feature matching". In: *European conference on computer vision*. Springer, 2010, pp. 791–804.

[75] D. C. Lindberg and D. C. Lindberg. *Theories of Vision from al-Kindi to Kepler*. University of Chicago Press, 1981.

[76] C. Linegar, W. Churchill, and P. Newman. "Made to measure: Bespoke landmarks for 24-hour, all-weather localisation with a camera". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 787–794.

[77] M. Lourakis and X. Zabulis. "Accurate Scale Factor Estimation in 3D Reconstruction". en. In: *Computer Analysis of Images and Patterns*. Ed. by D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, R. Wilson, E. Hancock, A. Bors, and W. Smith. Vol. 8047. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 498–506. ISBN: 978-3-642-40260-9. DOI: 10.1007/978-3-642-40261-6_60. URL: http://link.springer.com/10.1007/978-3-642-40261-6_60 (visited on 12/13/2020).

[78] D. Lowe. "Object recognition from local scale-invariant features". en. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Kerkyra, Greece: IEEE, 1999, 1150–1157 vol.2. ISBN: 978-0-7695-0164-2. DOI: 10.1109/ICCV.1999.790410. URL: http://ieeexplore.ieee.org/document/790410/ (visited on 12/16/2020).

[79] D. G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". en. In: *International Journal of Computer Vision* 60.2 (Nov. 2004), pp. 91–110. ISSN: 1573-1405. DOI: 10.1023/B:VISI.0000029664.99615.94. URL: https://doi.org/10.1023/B:VISI.0000029664.99615.94 (visited on 08/18/2020).

[80] B. D. Lucas, T. Kanade, et al. "An iterative image registration technique with an application to stereo vision". In: (1981). Publisher: Vancouver, British Columbia.

[81] A. M. Lund. "Measuring usability with the use questionnaire12". In: *Usability interface* 8.2 (2001), pp. 3–6.

[82] S. Lynen, T. Sattler, M. Bosse, J. A. Hesch, M. Pollefeys, and R. Siegwart. "Get out of my lab: Large-scale, real-time visual-inertial localization." In: *Robotics: Science and Systems*. Vol. 1. 2015.

[83] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry. *An invitation to 3-d vision: from images to geometric models*. Vol. 26. Springer Science & Business Media, 2012.

[84] R. C. Martin. *Agile software development: principles, patterns, and practices*. Prentice Hall, 2002.

[85] R. C. Martin. *Clean architecture: a craftsman's guide to software structure and design*. Prentice Hall, 2018.

[86] R. C. Martin. *Clean code : A handbook of agile software craftsmanship*. en. ISBN: 9780132350884 Publisher: Prentice-Hall. 2009. URL: https://cds.cern.ch/record/1281586 (visited on 12/11/2020).

[87] T. Masood and J. Egger. "Augmented reality in support of Industry 4.0—Implementation challenges and success factors". In: *Robotics and Computer-Integrated Manufacturing* 58 (2019). Publisher: Elsevier, pp. 181–195.

[88] *MauAR App*. de-DE. URL: https://mauar.berlin/en/ (visited on 01/22/2021).

[89] B. Meyer. *Object-oriented software construction*. Vol. 2. Prentice hall Englewood Cliffs, 1997.

[90] *Microsoft HoloLens | Mixed Reality Technology for Business*. en-us. URL: https://www.microsoft.com/en-us/hololens (visited on 01/26/2021).

[91] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt. "Scalable 6-DOF Localization on Mobile Devices". en. In: *Computer Vision – ECCV 2014*. Ed. by D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars. Vol. 8690. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2014, pp. 268–283. ISBN: 978-3-319-10604-5. DOI: 10.1007/978-3-319-10605-2_18. URL: http://link.springer.com/10.1007/978-3-319-10605-2_18 (visited on 08/17/2020).

[92] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt. "Scalable 6-dof localization on mobile devices". In: *European conference on computer vision*. Springer, 2014, pp. 268–283.

[93] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino. "Augmented reality: a class of displays on the reality-virtuality continuum". In: *Telemanipulator and Telepresence Technologies*. Vol. 2351. International Society for Optics and Photonics, Dec. 1995, pp. 282–292. DOI: 10.1117/12.197321. URL: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/2351/0000/Augmented-reality--a-class-of-displays-on-the-reality/10.1117/12.197321.short (visited on 09/08/2020).

[94] R. Mur-Artal and J. D. Tardos. "ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras". In: *IEEE Transactions on Robotics* 33.5 (Oct. 2017). arXiv: 1610.06475, pp. 1255–1262. ISSN: 1552-3098, 1941-0468. DOI: 10.1109/TRO.2017.2705103. URL: http://arxiv.org/abs/1610.06475 (visited on 08/17/2020).

[95] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. "Kinectfusion: Real-time dense surface mapping and tracking". In: *2011 10th IEEE international symposium on mixed and augmented reality*. IEEE, 2011, pp. 127–136.

[96] P. Nora. "Between memory and history: Les lieux de mémoire". In: *representations* 26 (1989). Publisher: University of California Press, pp. 7–24.

[97] *Now available: 3D Model Object Tracking*. en-US. July 2020. URL: https://www.wikitude.com/blog-sdk-9-2-now-available-3d-model-object-tracking/ (visited on 09/09/2020).

[98] P. Nowacki and M. Woda. "Capabilities of arcore and arkit platforms for ar/vr applications". In: *International Conference on Dependability and Complex Systems*. Springer, 2019, pp. 358–370.

[99] *OpenCV: Feature Detection and Description*. URL: https://docs.opencv.org/master/db/d27/tutorial_py_table_of_contents_feature2d.html (visited on 01/27/2021).

[100] *Output Format — COLMAP 3.7 documentation*. URL: https://colmap.github.io/format.html#images-txt (visited on 12/09/2020).

[101] C. Panou, L. Ragia, D. Dimelli, and K. Mania. "An architecture for mobile outdoors augmented reality for cultural heritage". In: *ISPRS International Journal of Geo-Information* 7.12 (2018). Publisher: Multidisciplinary Digital Publishing Institute, p. 463.

[102] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. "Coarse-to-Fine Volumetric Prediction for Single-Image 3D Human Pose". en. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI: IEEE, July 2017, pp. 1263–1272. ISBN: 978-1-5386-0457-1. DOI: 10.1109/CVPR.2017.139. URL: http://ieeexplore.ieee.org/document/8099622/ (visited on 12/15/2020).

[103] G. Pavlidis, A. Koutsoudis, F. Arnaoutoglou, V. Tsioukas, and C. Chamzas. "Methods for 3D digitization of cultural heritage". In: *Journal of cultural heritage* 8.1 (2007). Publisher: Elsevier, pp. 93–98.

[104] M. Pierrot-Deseilligny, L. De Luca, and F. Remondino. "Automated image-based procedures for accurate artifacts 3D modeling and orthoimage generation". In: *Geoinformatics FCE CTU* 6 (2011), pp. 291–299.

[105] D. A. Plecher, F. Herber, C. Eichhorn, A. Pongratz, G. Tanson, and G. Klinker. "HieroQuest - A Serious Game for Learning Egyptian Hieroglyphs". In: *Journal on Computing and Cultural Heritage* 13.4 (Dec. 2020), 30:1–30:20. ISSN: 1556-4673. DOI: 10.1145/3418038. URL: https://doi.org/10.1145/3418038 (visited on 12/14/2020).

[106] *Pokemon Go, 13 "Pokémon Go" Selfies That Wanna Be The Very Best*. en. URL: https://www.popbuzz.com/internet/social-media/pokemon-go-selfies-pictures-funny/ (visited on 02/01/2021).

[107] H. Rahaman and E. Champion. "To 3D or Not 3D: Choosing a Photogrammetry Workflow for Cultural Heritage Groups". In: *Heritage* 2.3 (2019). Publisher: Multidisciplinary Digital Publishing Institute, pp. 1835–1851.

[108] W. Reese. "Nginx: the high-performance web server and reverse proxy". In: *Linux Journal* 2008.173 (2008). Publisher: Belltown Media, p. 2.

[109] Y. K. Ro, A. Brem, and P. A. Rauschnabel. "Augmented reality smart glasses: Definition, concepts and impact on firm value creation". In: *Augmented reality and virtual reality*. Springer, 2018, pp. 169–181.

[110] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. "Mobilenetv2: Inverted residuals and linear bottlenecks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4510–4520.

[111] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk. "From Coarse to Fine: Robust Hierarchical Localization at Large Scale". In: *arXiv:1812.03506 [cs]* (Apr. 2019). arXiv: 1812.03506. URL: http://arxiv.org/abs/1812.03506 (visited on 08/17/2020).

[112] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. "SuperGlue: Learning Feature Matching with Graph Neural Networks". In: *arXiv:1911.11763 [cs]* (Mar. 2020). arXiv: 1911.11763. URL: http://arxiv.org/abs/1911.11763 (visited on 08/17/2020).

[113] R. Sasaki and K. Yamamoto. "A Sightseeing Support System Using Augmented Reality and Pictograms within Urban Tourist Areas in Japan". en. In: *ISPRS International Journal of Geo-Information* 8.9 (Sept. 2019). Number: 9 Publisher: Multidisciplinary Digital Publishing Institute, p. 381. DOI: 10.3390/ijgi8090381. URL: https://www.mdpi.com/2220-9964/8/9/381 (visited on 08/17/2020).

[114] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, F. Kahl, and T. Pajdla. "Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions". en. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT, USA: IEEE, June 2018, pp. 8601–8610. ISBN: 978-1-5386-6420-9. DOI: 10.1109/CVPR.2018.00897. URL: https://ieeexplore.ieee.org/document/8578995/ (visited on 11/30/2020).

[115] J. L. Schonberger and J.-M. Frahm. "Structure-from-Motion Revisited". en. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, June 2016, pp. 4104–4113. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.445. URL: http://ieeexplore.ieee.org/document/7780814/ (visited on 12/02/2020).

[116] Sergiu. *#bringArtToLife*. en-US. URL: https://artivive.com/ (visited on 01/25/2021).

[117] Y. Shavit and R. Ferens. "Introduction to Camera Pose Estimation with Deep Learning". In: *arXiv:1907.05272 [cs]* (July 2019). arXiv: 1907.05272. URL: http://arxiv.org/abs/1907.05272 (visited on 08/17/2020).

[118] A. Shaw, S. Choshen-Hillel, and E. M. Caruso. "Being biased against friends to appear unbiased". In: *Journal of Experimental Social Psychology* 78 (2018). Publisher: Elsevier, pp. 104–115.

[119] N. Snavely, S. M. Seitz, and R. Szeliski. "Photo Tourism: Exploring Photo Collections in 3D". en. In: (), p. 12.

[120] C. Stapleton and J. Davies. "Imagination: The third reality to the virtuality continuum". In: *2011 IEEE International Symposium on Mixed and Augmented Reality-Arts, Media, and Humanities*. IEEE, 2011, pp. 53–60.

[121] I. E. Sutherland. "A head-mounted three dimensional display". In: *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*. 1968, pp. 757–764.

[122] P. Tolstoi. "A Framework for location-based Augmented Reality Content on Mobile Devices". en. In: (), p. 67.

[123] *Tools, 5 Best Augmented Reality Development Tools for Developers*. en. Nov. 2019. URL: https://program-ace.com/blog/augmented-reality-sdk/ (visited on 09/08/2020).

[124] S. Ullman. "The interpretation of structure from motion". In: *Proceedings of the Royal Society of London. Series B. Biological Sciences* 203.1153 (1979). Publisher: The Royal Society London, pp. 405–426.

[125] D. Van Krevelen and R. Poelman. "A Survey of Augmented Reality Technologies, Applications and Limitations". en. In: *International Journal of Virtual Reality* 9.2 (Jan. 2010), pp. 1–20. ISSN: 1081-1451. DOI: 10.20870/IJVR.2010.9.2.2767. URL: https://ijvr.eu/article/view/2767 (visited on 09/08/2020).

[126] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. "Attention is all you need". In: *arXiv preprint arXiv:1706.03762* (2017).

[127] S. Vert and R. Vasiu. "Relevant Aspects for the Integration of Linked Data in Mobile Augmented Reality Applications for Tourism". en. In: *Information and Software Technologies*. Ed. by G. Dregvaite and R. Damasevicius. Communications in Computer and Information Science. Cham: Springer International Publishing, 2014, pp. 334–345. ISBN: 978-3-319-11958-8. DOI: 10.1007/978-3-319-11958-8_27.

[128] V. Vlahakis, M. Ioannidis, J. Karigiannis, M. Tsotros, M. Gounaris, D. Stricker, T. Gleue, P. Daehne, and L. Almeida. "Archeoguide: an augmented reality guide for archaeological sites". In: *IEEE Computer Graphics and Applications* 22.5 (2002). Publisher: IEEE, pp. 52–60.

[129] G.-D. Voinea, F. Girbacia, C. C. Postelnicu, and A. Marto. "Exploring Cultural Heritage Using Augmented Reality Through Google's Project Tango and ARCore". en. In: *VR Technologies in Cultural Heritage*. Ed. by M. Duguleană, M. Carrozzino, M. Gams, and I. Tanea. Communications in Computer and Information Science. Cham: Springer International Publishing, 2019, pp. 93–106. ISBN: 978-3-030-05819-7. DOI: 10.1007/978-3-030-05819-7_8.

[130] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. "PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes". In: *arXiv:1711.00199 [cs]* (May 2018). arXiv: 1711.00199. URL: http://arxiv.org/abs/1711.00199 (visited on 09/09/2020).

[131] Yan Ke and R. Sukthankar. "PCA-SIFT: a more distinctive representation for local image descriptors". In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* Vol. 2. ISSN: 1063-6919. June 2004, pp. II–II. DOI: 10.1109/CVPR.2004.1315206.

[132] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. "LIFT: Learned Invariant Feature Transform". en. In: *Computer Vision – ECCV 2016*. Ed. by B. Leibe, J. Matas, N. Sebe, and M. Welling. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 467–483. ISBN: 978-3-319-46466-4. DOI: 10.1007/978-3-319-46466-4_28.

[133] Z. Yovcheva, D. Buhalis, and C. Gatzidis. "Smartphone augmented reality applications for tourism". In: *E-review of tourism research (ertr)* 10.2 (2012), pp. 63–66.

[134] A. Zderadičková. "HoloCopy - kopirování v prostoru s Hololens". en. In: (June 2020). Accepted: 2020-06-19T22:51:31Z Publisher: České vysoké učení technické v Praze. Vypočetní a informační centrum. URL: https://dspace.cvut.cz/handle/10467/88253 (visited on 11/19/2020).

[135] L. Zhang and S. Rusinkiewicz. "Learning to detect features in texture images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 6325–6333.