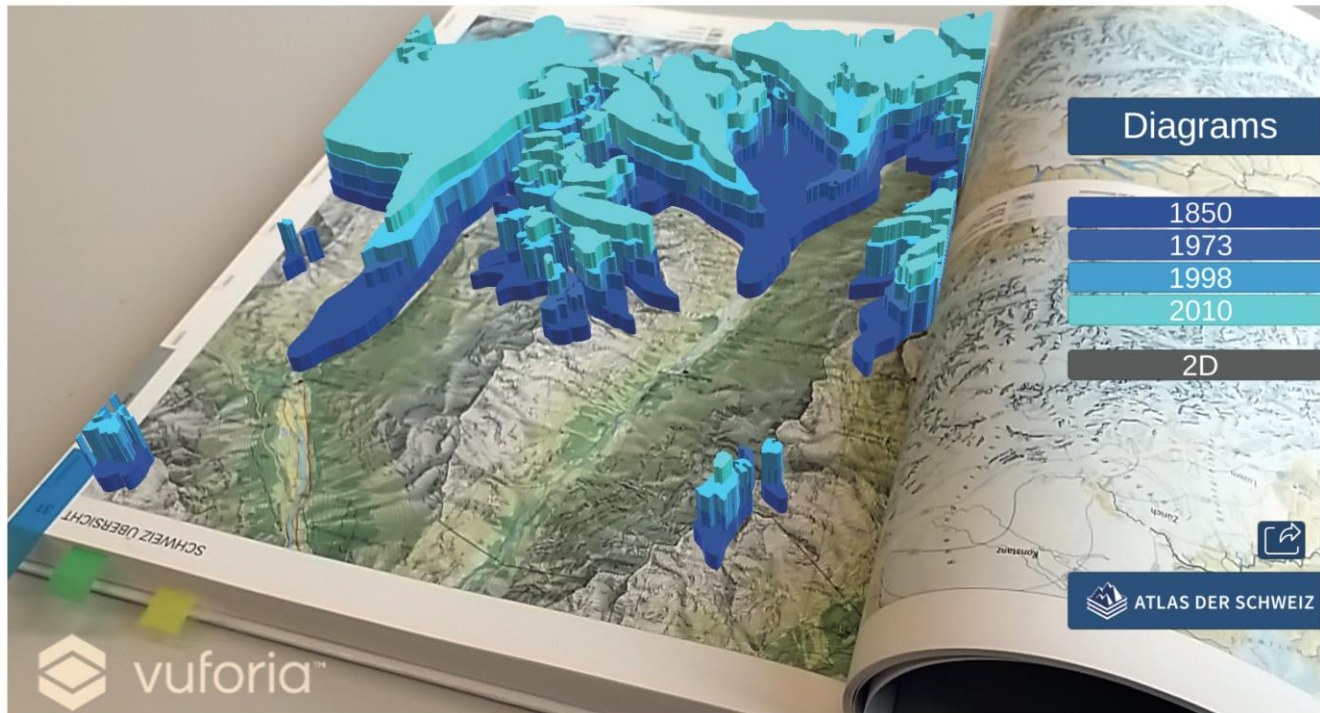


Augmented Reality for Printed School Atlases

Interdisciplinary Project Work



Institute of Cartography and Geoinformation, ETH Zurich
Geomatics MSc, Autumn Semester 2018, 11 January 2019

Authors: Cédric Dind (dindc@ethz.ch)
Stefan Schalcher (schalste@ethz.ch)

Examiner: Prof. Dr. Lorenz Hurni

Supervisors: Raimund Schnürer, Pascal Tschudi

External Supervisor: Mattia Ryffel

Preface and acknowledgements

Shortly after the announcements of the themes for the Interdisciplinary Project Work back in April 2018, we realized that this project was the most appealing one to us. Especially, we were very keen to work with Augmented Reality and examine its potential in the school environment. Additionally, creating an impressive application as an outcome of this project increased our motivation considerably.

We would like to thank our external supervisor Mattia Ryffel from Disney Research. His knowledge of Augmented Reality, especially of the game engine Unity, proved to be essential for the success of our project. His ideas and inputs during our meetings, as well as his pieces of advice whenever we faced a programming problem were crucial.

Our grateful thanks go also to our supervisors Raimund Schnürer and Pascal Tschudi. Their support, ideas and constructive criticism throughout the whole project were fundamental for the outcome of this project as well. Moreover, many thanks for their patience and for their review of our poster, presentation and the written reports.

Abstract

The purpose of this Interdisciplinary Project Work is to prototypically implement an Augmented Reality Android application, which visualizes georeferenced augmented 2D and 3D elements on maps from the Swiss World Atlas.

After a short literature review on existing Augmented Reality techniques and related works, the content of the planned application is determined: Three printed 2D maps from the Swiss World Atlas are chosen and an adequate connection to three digital 3D maps from the Atlas of Switzerland is established. Afterwards, the chosen datasets are transformed into georeferenced 2D or 3D objects on top of the maps using the cross-platform game engine Unity and the Augmented Reality software development kit Vuforia. During this implementation, the challenges of importing and displaying the datasets in Unity as well as adding functionalities and interactivity to the newly created objects are solved successfully.

The findings of this Interdisciplinary Project Work show that it is feasible to display augmented objects on top of maps and that such use of applications should be considered more often in schools in the future. A follow-up study may examine which didactic concepts are most suitable for the developed visualizations and functions.

Table of content

Preface and acknowledgements	ii
Abstract	iii
Table of content	iv
List of figures	v
List of tables	v
List of Abbreviations	vi
1 Introduction	1
1.1 Goals	2
1.2 Description of users	2
1.3 Methods and Steps	3
1.4 Time Management	4
2 Background	5
2.1 Augmented Reality	5
2.2 Review of related work and competing products	6
2.3 Feature detection/description/matching algorithms and rendering techniques	8
3 Implementation	11
3.1 Method	11
3.1.1 Map recognition & Tracking	11
3.1.2 Data import	13
3.1.3 Functionalities & Interactivity	15
3.2 Results	18
3.2.1 Glacier map of the Bernina region, Switzerland	18
3.2.2 Population density map of Europe	19
3.2.3 Tourism map of the world	20
4 Discussion	22
5 Summary	23
6 Outlook	24
Bibliography	25

List of figures

Figure 1: Time management	4
Figure 2: Explore the Livemap Switzerland with the accompanying app	6
Figure 3: The Disney Color and Play application	7
Figure 4: The Layar application	7
Figure 5: Flow chart of an AR application using markers	8
Figure 6: Pixels of the FAST algorithm	9
Figure 7: Vuforia target rating of the different maps	11
Figure 8: Target images of maps	12
Figure 9: Placement of virtual object on map covering two pages	12
Figure 10: Creation of a user defined coordinate reference system	13
Figure 11: The GUI overlay changes according to the detected map	15
Figure 12: Toggle object visibility on click on button	16
Figure 13: Animation on click on button	16
Figure 14: Feature query on click on object	17
Figure 15: Function for the pie chart billboards of the energy consumer groups	17
Figure 16: Interface of the glacier map	18
Figure 17: AR view of the glacier map	19
Figure 18: AR view of the population density map of Europe	20
Figure 19: AR view of the tourism map of the world	20

List of tables

Table 1: Milestones	4
Table 2: Overview of implemented visualizations	23

List of Abbreviations

ADS	Atlas der Schweiz (Atlas of Switzerland)
AR	Augmented Reality
GUI	Graphical User Interface
IDE	Integrated Development Environment
SWA	Swiss World Atlas
VR	Virtual Reality

1 Introduction

Nowadays, more and more students in classes of the secondary school levels in Switzerland are equipped with tablets. One of the reasons is the introduction of the new school subject “Media and Computer Science”. On the opposite side, the use of Augmented Reality (AR) is beginning to receive attention in classes. This Interdisciplinary Project Work shall investigate how these tablets can be used to display thematic information with AR techniques in combination with a printed school atlas.

With AR, it is possible to place digital content into the real world allowing the user to interact with this digital content in real time using such a device. AR techniques can therefore be used in combination with a printed school atlas to enrich such atlas maps by additional learning materials as well as interactive functionalities. Furthermore, using tablets may also increase the students’ motivation to explore geographic patterns and processes depicted in the maps. For all these reasons, the use of AR techniques in classes seems more than worth investigating.

The goal of this Interdisciplinary Project Work is to create an AR Android application for the Swiss World Atlas. A prototypical application should be implemented, which visualizes georeferenced augmented 2D and 3D elements on exemplary printed maps from the Swiss World Atlas. Besides this, the final implemented application should be enhanced by simple animations, interactive functionalities and a straightforward and user-friendly graphical user interface.

1.1 Goals

The overall goal of our Interdisciplinary Project Work is to ***create an AR Android Application with Unity and Vuforia for the Swiss World Atlas.***

In the end of our project, a well-functioning application should be implemented which visualizes a choice of georeferenced augmented 2D/3D elements on four suitable printed maps from the Swiss World Atlas. The application should further be well designed, easily comprehensible and intuitive.

To accomplish this, we divided our overall goal into the following sub goals:

- Choose 3 printed maps from the Swiss World Atlas suitable for our project
- Choose 3 themes from the Atlas of Switzerland suitable for our chosen maps
- Find suitable markers on the 3 chosen printed maps
- Processing/conversion of geodata
- Augmented 2D/3D elements are georeferenced
 - Implement 6 different visualization types
- Functional implementation of GUI for interactivity
 - Implement 4 different functions (not styled ones, just simple inputs like text boxes, dropdowns)
- Touch inputs (e.g. for feature query)
- Animations of 2D/3D elements (simple elements like trains or airplanes to make a map more alive)
- Short review of final application

1.2 Description of users

Our prototype of an AR application for Printed School Atlases should be used mainly in schools in Switzerland, where a printed version of the Swiss World Atlas (SWA) is being used. Mainly students in geography classes in secondary school and grammar school are the target group for our project. The age group will accordingly be between 12 and 20 years. With the help of our AR application and accompanying tablet, students should be motivated and encouraged to explore and learn geographic structures and processes depicted in maps.

Teachers can integrate our AR application during their geography lesson. It can be used to show existing thematic information in a different way or additional information on top of the printed maps. Therefore, our application can be used as a new and interesting teaching instrument or as an additional teaching assistance.

1.3 Methods and Steps

- **Project management:**
In a first step, we set up an online Kanban Board (Wekan) as our project management tool [10]. This helps to structure the work and make sure everything is tested and completed in time. It also works as a one-way communication tool to the supervisors
We used Git in form of GitLab, a web-based Git-repository manager, to be able to work on the implementation simultaneously, to track the changes we made, for easier prototyping and to backtrack errors and malfunctions. [15].
- **Literature research:**
To get the basic knowledge for this project literature research on Augmented Reality, 3D Cartography, Atlases and Application Development is done.
- **Market analysis:**
We analyze the market to learn from related projects and competing works.
- **Unity/Android Tutorials:**
Before starting with the implementation, we plan to do Unity and Android tutorials to get familiar with the programs and programming languages required to develop the application.
- **Rapid prototyping:**
To get some first examples of the application early on we use rapid prototyping as a strategy during the implementation. Seeing what works easily and where the limitations of the project are in an early stage of the implementation helps to refine the time management.
- **Testing:**
In the end, we use the feedback from the poster presentation at the Geomatikseminar to assess the application for robustness and usefulness.
- **Tools used to create the AR application:**
The data used in this project is completely provided by the SWA and the ADS. For the coordinate transformation of the data, the open source geographical information system QGIS is used [11]. The application is developed in Unity, a game engine that doubles as runtime system and integrated development environment (IDE) [12]. The programming IDE in this project is Microsoft Visual Studio [13] and the programming language C#. For the recognition and tracking, the AR SDK Vuforia is used. It is already included in Unity [14].

1.4 Time Management

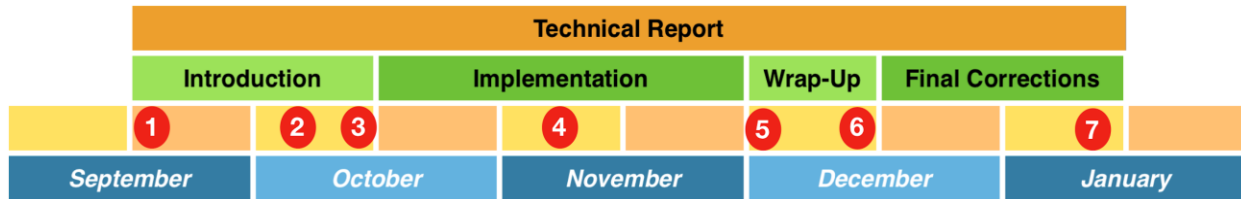


Figure 1: Time management

Table 1: Milestones

1	Tue, 18 September	Kick-off meeting with Disney Research
2	Su, 7 October	Milestone: Introductory part of report (incl. concept) + Choice of maps, themes, visualization types and functions
3	Wed, 10 October	Intermediate meeting with Raimund, Pascal and Professor Hurni: Discussion of concept
4	12 – 16 November	Second meeting with Disney Research: Discussion of implementation
5	Su, 2 December	Milestone: Implementation finished and documented
6	Thu, 13 December	Milestone: Poster and presentation at the ETH-Geomatikseminar
7	Fr, 11 January	Milestone: Handing in report and code

2 Background

In the following sections an overview of Augmented Reality and related work to our project is given.

2.1 Augmented Reality

Augmented Reality (AR) is “an enhanced version of reality created by the use of technology to overlay digital information on an image of something being viewed through a device [...]” (Merriam-Webster [23]). In other words, AR places digital content into the real world and allows the user to interact with this digital content in real time.

AR and Virtual Reality (VR) are trends towards integrating computer-generated content in our lives. While VR takes the user into a completely computer-generated world with no influences of the reality, AR augments the real world with virtual objects. Seeing the real world behind the virtual objects helps to not get motion sickness, which is possible in VR.

In the 1960s, Sutherland started the fields of VR and AR. He constructed the first VR system in 1968. The term “Augmented Reality” first appeared in 1992 in a project of Boeing. In the construction of aircrafts, they used a head-mounted device for guiding the assembly of wire bundles. Other use cases followed, for example a handheld AR device in 1993, a medical application in 1994 and a collaborative AR system for multiple users in 1996. [1]

AR systems got more and more common over the years and the general public was introduced to the concept of AR with the release of the mobile game Pokémon Go in 2016.

2.2 Review of related work and competing products

A short review of other work is given with a focus on the AR experience Livemap Switzerland (formerly Swiss Arena) and the AR applications Disney Coloring Book and Layar.

The *Livemap Switzerland* is located in the Swiss Museum of Transport (Verkehrshaus) in Lucerne. Consisting of a walk-on aerial photograph, it offers an unusual view of Switzerland. Every city, town and village can be explored as the high-resolution photography shows every detail of Switzerland. Additionally, an accompanying application brings this walk-on experience to life. With the help of AR, it is possible to project the latest weather, traffic information, pictures or places of interest onto the large-scale map.



Figure 2: Explore the Livemap Switzerland with the accompanying app [2]

Disney Research has developed an application called *Disney Color and Play*. This application is able to take a 2D colored drawing and map it to a 3D model. As it can be seen in figure 1, the colored character is animated in real time in 3D. The user simply points the tablet's camera at the coloring book, and the application will show how the texture of the 3D model changes as the user draws.



Figure 3: The Disney Color and Play application [3]

The mobile application *Layar* allows users to find various items based upon the built-in technologies in a regular smartphone (accelerometer, built-in camera, compass, GPS). These technologies are used together to identify the user's location and field of view. From that geographical position, various forms of data are laid over the camera view like inserting an additional layer.



Figure 4: The Layar application [4]

2.3 Feature detection/description/matching algorithms and rendering techniques

The basic sequence of steps in AR using markers are feature detection, feature description, feature matching and rendering. First, the marker is analyzed for features, which then have to be described. While running the AR application, it tries to match those features with features found in the receiving frame of the camera. When a match is found, the application computes the camera position and rotation from the perspective transformation of the marker. Finally, the virtual objects are rendered on the screen of the device accordingly (Figure 5).

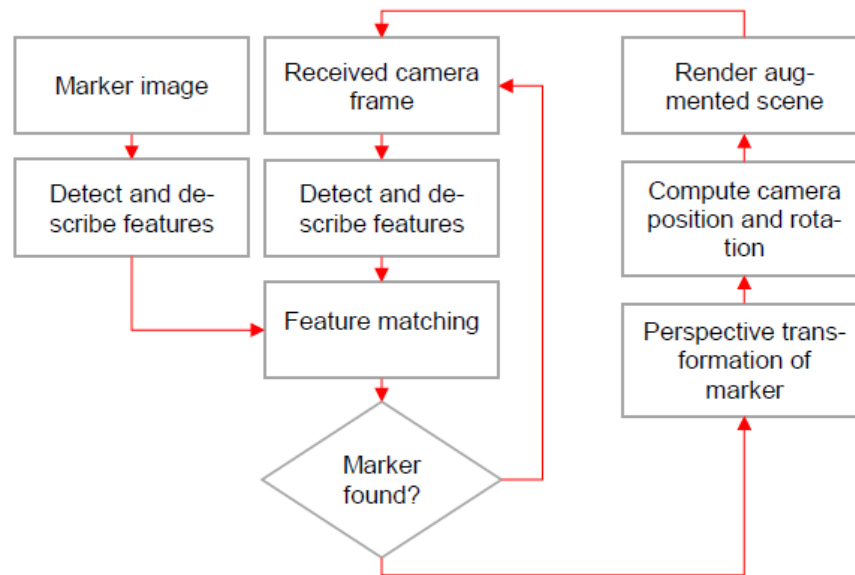


Figure 5: Flow chart of an AR application using markers

In this project, Vuforia is used for the AR workflow. With Vuforia not being open-source, we do not know what algorithms it uses for feature detection, description and matching. Therefore, we describe an exemplary selection of such algorithms in the following section.

There are many feature detection algorithms but for AR the algorithm should work in real time. An example of such an algorithm is FAST (Features from Accelerated Segment Test). Rosten and Drummond introduced it in 2006.

To decide whether a pixel P is a point of interest or not, the algorithm compares the brightness value of the pixel P to the values of 16 pixels that lie in a circle around P . If twelve continuous pixels on the circle are all brighter (or all darker) than the pixel P , the pixel P is labeled as a corner.

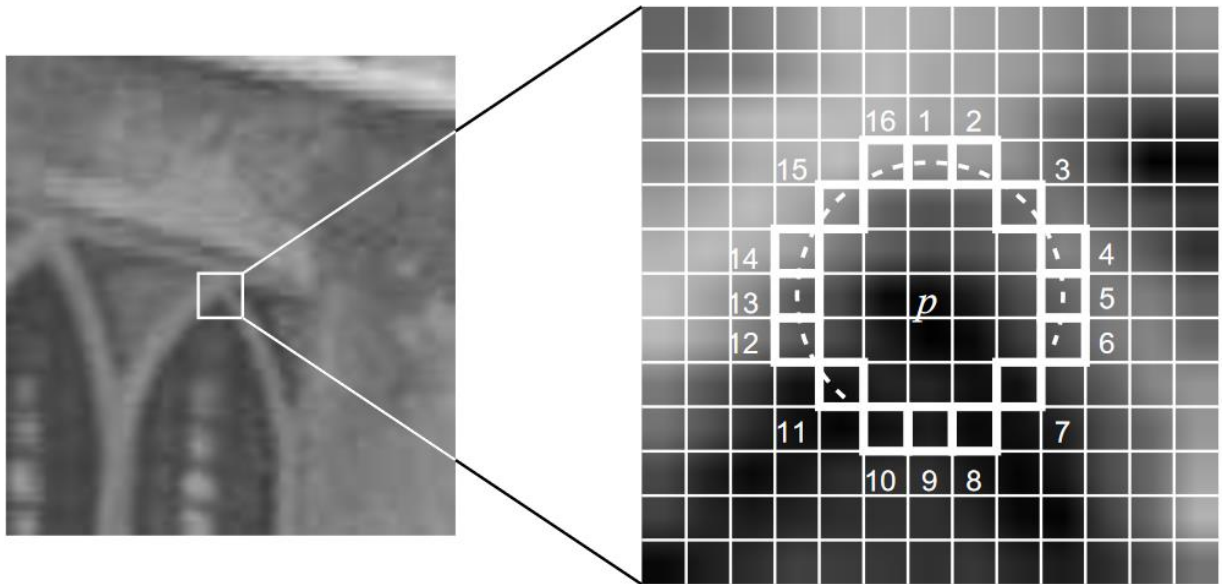


Figure 6: Pixels of the FAST algorithm [5]

To speed up the algorithm, it first checks if out of the pixels 1, 9, 5 and 13 at least three have a higher (or lower) brightness value than P . If that is not the case, P is rejected, else the algorithm checks for twelve pixels.

Because the FAST algorithm only detects features but does not describe them, it needs an additional feature descriptor, like HOG (Histogram of oriented gradients). Today there are also feature detectors which have an integrated descriptor, like SIFT (Scale-invariant feature transform) or SURF (Speeded up robust features). These descriptors describe the elementary characteristics of the surrounding image area of a feature (e.g. shape, texture or color). [6]

FLANN is a library for nearest neighbor searches. It contains a set of algorithms and calculates for a given dataset the optimal algorithm and parameters to “perform fast approximate nearest neighbor searches in high dimensional spaces” [7].

After the marker has been found by matching of the features of the marker and the features of the received camera frame, the position and rotation of the camera can be obtained from the perspective transformation of the marker in the camera frame.

Rendering is the way of transforming a 3D-scene into a 2D-image. There are different techniques to do this, for example ray tracing or rasterization.

Ray tracing uses single rays originating from a camera and determines their intersection with a geometric object to get a high-quality image as result. This process is rather slow and normally used for “photorealistic” rendering. Today, real-time ray tracing is possible but limited to high performance hardware, which means it will take some time until it can be used for AR applications on mobile devices. [8]

Rasterization on the other hand is faster but produces lower quality images. It is suitable for real-time application. Rasterization is usually realized via the rendering pipeline. This pipeline has three stages: In the application stage, the 3D scene is managed and the objects to be drawn are determined and sent to the computer graphics system. In the geometry stage, light intensities are computed. Through model and view transformation, projection, clipping and screen mapping the geometric primitives are transformed into screen coordinates. Finally, in the rasterization stage, the geometries are discretized via rasterization and the data is written into the final image [9].

One feature of AR applications is that they display visual content in real-time and the rendering has to adapt fast to changes made by interacting with the object. Therefore, rasterization is used for rendering the 3D objects in AR applications.

3 Implementation

3.1 Method

In this section, we talk about the methods that we used to create this application and how we used the tools stated in section 1.3.

3.1.1 Map recognition & Tracking

For the recognition and tracking of the map, the Augmented Reality software development kit (SDK) Vuforia is used [16]. It requires a registration in which the user obtains a license key, and it is not open source, but free to use. Unity supports Vuforia for a seamless integration. Vuforia matches extracted features of a marker image with those from camera frames from the device. This enables to compute the camera position and rotation and subsequently to render virtual objects on the screen of the device in real-time, which creates this Augmented Reality.

In this project, Vuforia is used for the recognition and tracking of maps. For that, the maps must be uploaded as images to the Vuforia Target Manager. The images must be smaller than 2.25 MB and 8- or 24-bit either PNG or JPG files. JPG files must be RGB or grayscale [17]. Vuforia analyzes the image and extracts features based on local contrast. The image targets get a rating from Vuforia for their traceability from one to five stars. A bad rating can be enhanced by increasing the local contrast of the image. The rating of the maps used in this project can be seen in Figure 7.




Target Name	Type	Rating
 068_1_europa	Single Image	★★★★★
 202_2_erde	Single Image	★★★★☆
 031_1_gletscher	Single Image	★☆☆☆☆

Figure 7: Vuforia target rating of the different maps

The map of Europe gets the best rating. As it can be seen in Figure 8, it is clearly the map with the highest contrast and most detected feature points. The map of Europe as well as the world map both are recognized and tracked well by the camera on the running application. The glacier map on the other hand has clearly less features. The image of this map has less contrast because the map is more detailed and less generalized. When running the application, it struggles

with detecting the glacier map, especially under bad lighting conditions. Holding the tablet parallel to the map and getting the whole map into the camera range helps with the recognition. Once recognized, the tracking of the glacier map works as well as with the other maps.

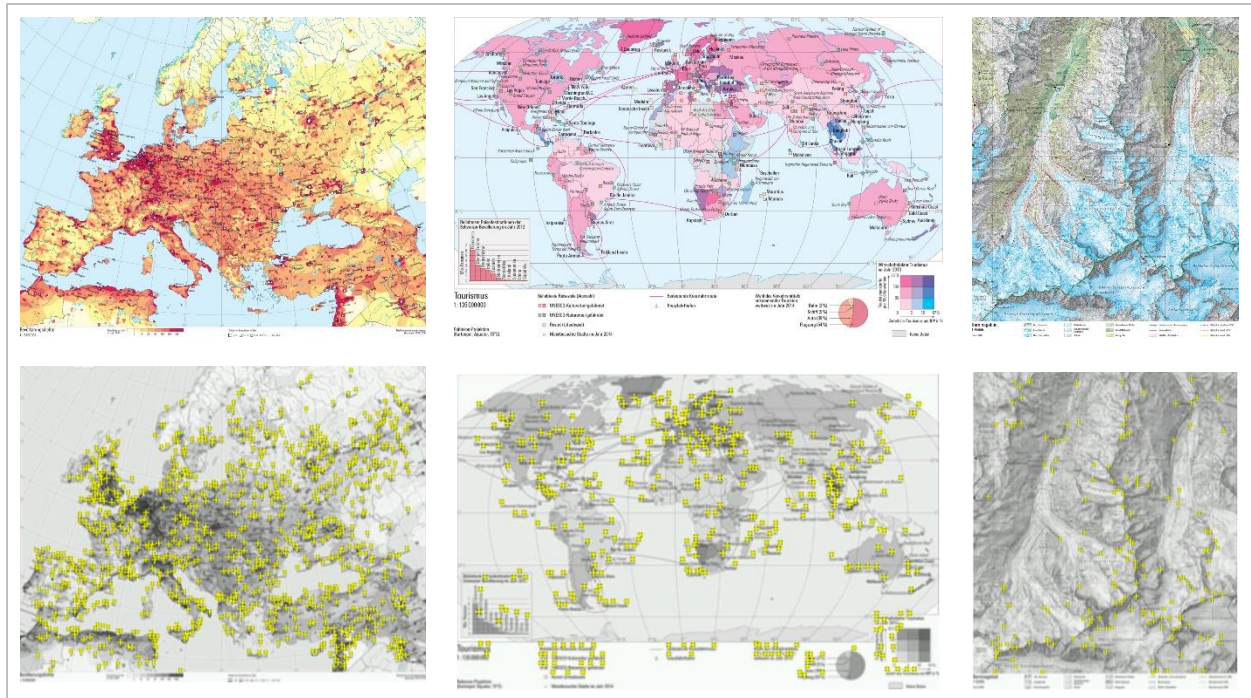


Figure 8: Target images of maps (left to right: Europe, World, Glacier; bottom: extracted features)

The map tracking and virtual object placement was tested on three devices. The oldest device (*Asus Transformer Pad TF300T* first release 2012) does not perform as smoothly as the two up to date devices (*Samsung Galaxy Tab S4* first release 2018 and *Sony Xperia XZ1 Compact* first release 2017) which do not have any problems displaying the virtual objects while tracking the map.

The placement of the objects happens by fitting a plane onto the recognized feature points because the input for Vuforia is a flat image. On a flat printed map, this works exactly and even on slightly bend pages of an atlas, due to the scale of the objects and the size of the device screen, nearly no deviation can be observed by eye. On strongly bend pages and maps that cover two pages of an atlas however, there are big deviations (Figure 9). Vuforia places the virtual object



Figure 9: Placement of virtual object (blue) on map (red) covering two pages [18]

based on the biggest recognized part of the map. This means on one page the positioning works perfectly while on the other there is an error in height and in the scale, because with the bending the width of the map gets shorter and the book is normally not opened in a 180° angle but a reflex angle.

3.1.2 Data import

The data was mainly available in form of GeoJSON files. GeoJSON is an open format based on JSON designed to represent geographical features with attributes. Those GeoJSON features are not directly importable in Unity. Two approaches were used to get the data onto the maps: In both approaches, first the GeoJSON files were imported into a QGIS project. There, the coordinates of the objects were transformed to fit the map projection. This was done by creating the reference systems of the maps as new user-defined coordinate reference systems (CRS) with the global projection specification parameters (Figure 10). The user coordinate systems were then applied to the features.

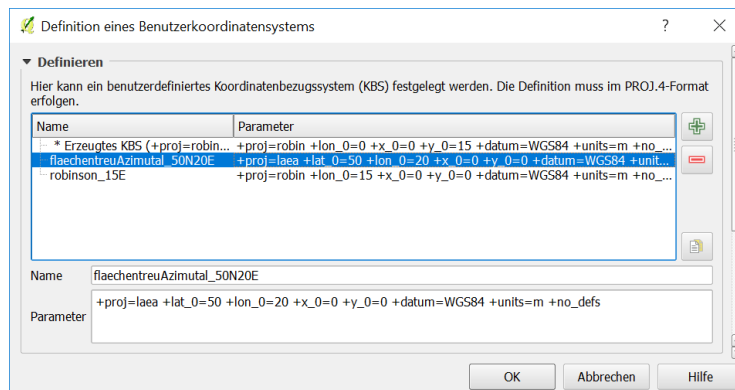


Figure 10: Creation of a user defined coordinate reference system

First approach: Import via Blender

Unity is not able to directly import data formats that contain geocoding. Blender is able to create files that are directly importable in Unity and with the BlenderGIS add-on it can also handle SHAPE files [19]. Therefore, the features were exported as SHAPE files from QGIS and imported in Blender. Unfortunately, this BlenderGIS add-on cannot completely handle the user created CRS. This means that the projection is correct but there is an unknown scaling error and an offset of the projection center. Blender files can then be imported in Unity, however the features have to be set manually to the correct position and correct scale on the map due to the errors mentioned before. Of course, this cannot be done precisely but by including the map frame in the features the frame can be set to fit the position and scale of the map in Unity rather well.

The data imported in this way was used in the application (glacier extents and tourism data) but the method was classified as imprecise, complex and using too many different programs.

Second approach: GeoJSON to C# object deserialization

This direct approach to import GeoJSON objects in Unity uses the GeoJSON .Net library [20] and the Newtonsoft Json.NET library [21]. In a C# script, the GeoJSON file is read into a string and deserialized with the Newtonsoft *JsonConvert* class into a GeoJSON.Net *FeatureCollection*. In the Unity Editor, objects that serve as prefabs are created. With the prefabs and the attributes of the features in the *FeatureCollection*, the virtual objects to display on the map are created. Applying a simple scale (size of the map in the atlas compared to the visible extent of the east coordinate in the map CRS) to the prefab places the objects correctly on the map.

3.1.3 Functionalities & Interactivity

For the sake of brevity of this report, all functionalities and interactivity are described in simplified schemas in this section. In Unity, all animations and functionalities are handled by scripts in C#.

The overall graphical user interface

The long-term goal for the graphical user interface (GUI) is to keep it as simple and clear as possible. For this reason, the GUI should take only a small amount of space on the display.

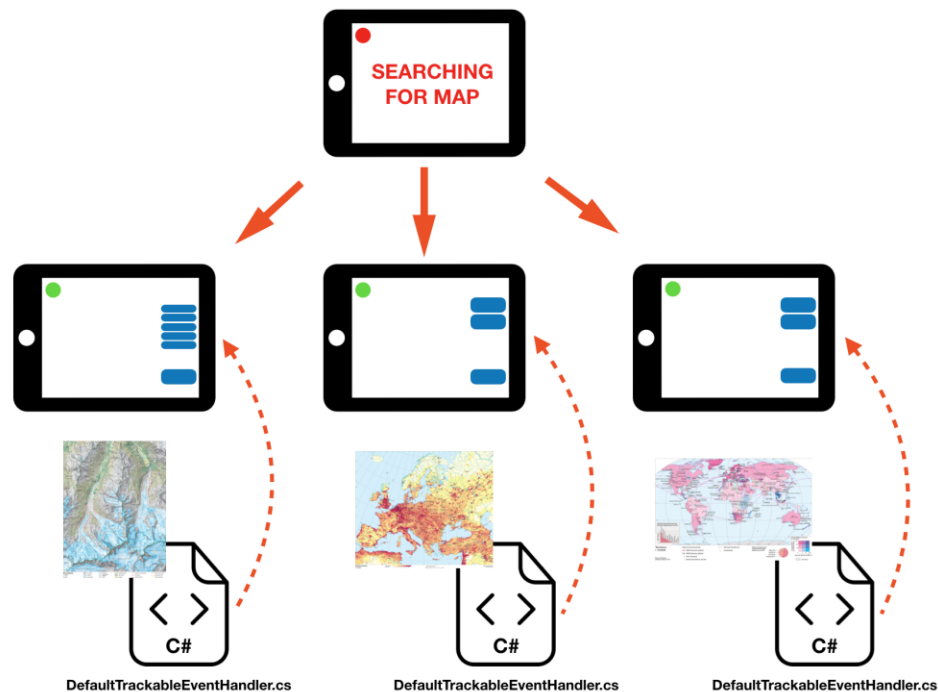


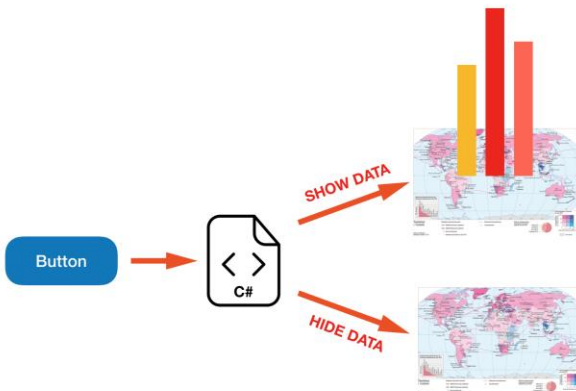
Figure 11: The GUI overlay changes according to the detected map

After an introductory page, the application switches into search mode, which is indicated by a small red status circle in the upper left corner of the display. As long as there is no map detected, the application stays in this mode and the message „SEARCHING FOR MAP“ is visible in the center of the display. Once a map has been detected, the application goes into map mode, which is now indicated, by a green status circle in the upper left corner.

Besides this, three different overlays have been implemented. To every image target in Unity, a C# script called „DefaultTrackableEventHandler.cs“ is linked and is executed once a map has been detected. In this script, a short code segment has been inserted which tells the application the correct overlay that has to be shown according to the detected map.

What all overlays have in common is on the one hand a save and share button where the user can take a screenshot of the current extent of the display and share it afterwards. On the other hand, a link button has also been implemented which takes the user to the Swiss World Atlas website. The rest of the GUI elements such as simple buttons for displaying and hiding data, time slider or a dropdown menu differ according to the currently detected map.

Buttons & Animations



All the buttons are linked to various C# script. So if a click on a button has been detected, the corresponding script including its functions is executed. Functions can cover for example setting augmented objects active or inactive for displaying them on the map, or starting an animation for such objects.

Figure 12: Toggle object visibility on click on button

Animations are predefined manipulations of the objects over a period. In Unity, this is achieved in the animator. In this animator, the user can manipulate (transform, scale...) a selected object in a predefined time frame to make such object appear, move or disappear on the fly. This animation is linked to the selected object and can afterwards be called and played through a C# script. For example, an animation for the legend has been implemented to make it appear by a button click.

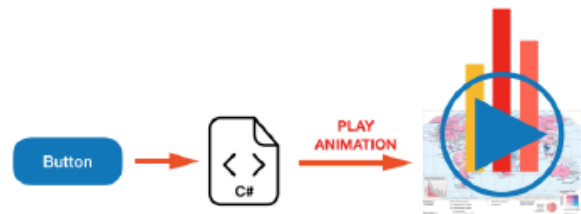


Figure 13: Animation on click on button

Feature Query

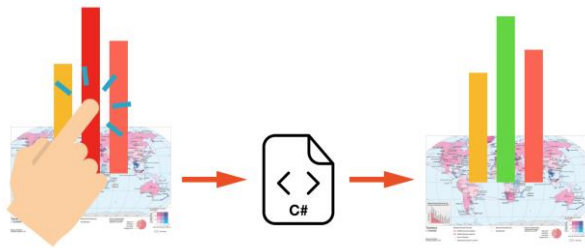


Figure 14: Feature query on click on object

The possibility to click on augmented objects and get information in real time is achieved with a C# script as well. In this case, the so-called RaycastHit function (<https://docs.unity3d.com/ScriptReference/RaycastHit.html>) is responsible for this. This function is called once per frame and checks in each frame, whether a touch input on the screen is interfering with an augmented object. If so, some code can be executed on the interfered object such as changing its color or showing its properties in a legend or in an info window.

```
void Update()
{
    if (Input.GetMouseButtonDown(0))
    {
        Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
        RaycastHit hit;
        if (Physics.Raycast(ray, out hit))
        {
            if (hit.transform.parent.transform.tag == "Diagramm")
            {
                string s = hit.transform.parent.transform.GetComponent<Text>().text;
                textObject.GetComponent<TextMesh>().text = s.Split('#')[0];
            }
            else
            {
                textObject.GetComponent<TextMesh>().text = "";
            }
        }
        else
        {
            textObject.GetComponent<TextMesh>().text = "";
        }
    }
}
```

Figure 15: The RaycastHit function for the pie chart billboards of the energy consumer groups

3.2 Results

The result of this project is a fully functioning Android application that enables the user to look at and interact with virtual components that are augmenting three maps of the SWA if the camera of the device is facing those maps. The graphical user interface of the application has been designed with simplicity and clarity in mind and it has an overall visual design scheme. The interactivity kept simple by limiting it to clicking (touch) and sliding.

3.2.1 Glacier map of the Bernina region, Switzerland

On the map from the SWA displaying the glacier extents in the Bernina region in Switzerland, two topics can be visualized in Augmented Reality.

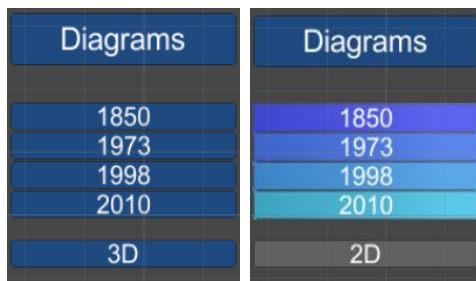


Figure 16: Interface of the glacier map (left: not toggled, right: toggled)

The interface of this map has a button to show diagrams, four buttons to show and hide different years of glacier extents, a button to switch between 2D and 3D and the rest of the standard interface.

On click, the buttons for the years change to the color of the polygon they represent. This way they act as layer switcher and as a legend for the data.

The other two buttons get grey when being toggled.

The glacier extents of different years are printed as lines in different colors on the map in the atlas. To make the differences clearer and to be able to show up-to-date data, the extents can be shown in AR with different colored polygons (Figure 17: left). The polygons of the different years (1850, 1973, 1998, and 2010) are on separate layers and each can be enabled and disabled. This way it is easier to compare pairs of the glacier extents. To show the difference more explicitly and to give the user a better experience, the polygons can be shown in 2D or as 3D extruded polygons. There is an extension in height animation when the 3D button is clicked. The polygons are imported with the first data import approach (Section 3.1.2.).

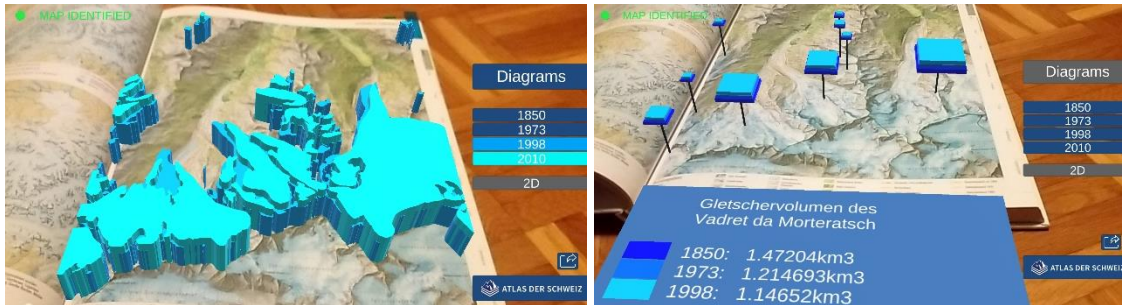


Figure 17: AR view of the glacier map (left: glacier extends; right: diagrams of glacier volumes)

The second visualization depicts 3D volume diagrams, so-called stacked cuboids as seen in Figure 17 on the right [22]. The diagrams show the volumes of glaciers in a quantitative manner with the different layers representing different years. For better comprehensibility, they are colored the same way as the polygons of the extents. With a click on the diagrams button, also a legend appears on the bottom of the page of the atlas. On click on one of the diagrams, the name of the glacier appears in the legend below and the values of the volume in km^3 . This way, the user can compare the glacier recession in absolute values.

3.2.2 Population density map of Europe

Two topics are visualized in Augmented Reality on the SWA map of the population density map of Europe.

The first visualization is the population density which is displayed as a 3D point cloud (Figure 18: left). In this cloud, the 3D points are positioned on top of each other and each point represents a population density of 100 inhabitants per 100 km^2 . For regions where the population density is lower, no points are generated. This decision has been made, because on the one hand the visualization gets too confusing and crowded as more points are being displayed. On the other hand, the tablet sometimes could not handle the large amount of displayed points very well.

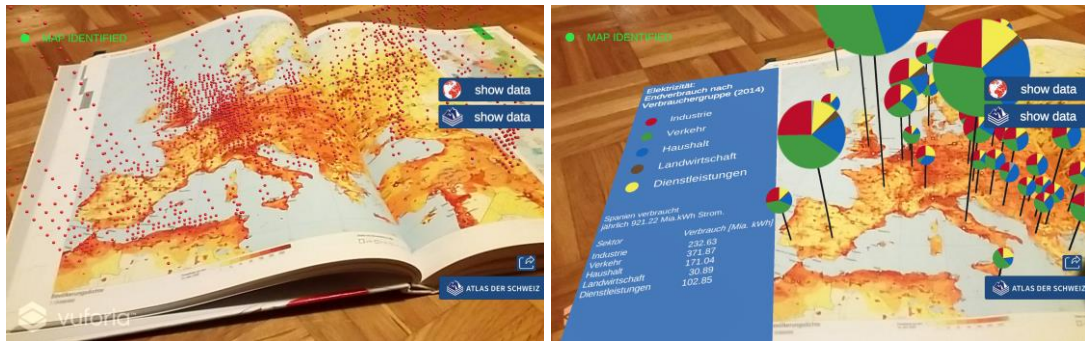


Figure 18: AR view of the population density map of Europe (left: 3D point cloud; right: pie charts)

The second visualization are pie chart billboards of the energy consumer groups per country in Europe (Figure 18: right). For a better understanding, an informative legend has been implemented in the left side map, where all the consumer groups are listed and the user can clearly distinguish between them. Furthermore, on click of one of these pie charts on the map, the name of the country appears in the legend with the exact numbers for each consumer group in that country.

3.2.3 Tourism map of the world

On the tourism map from the SWA, two different topics are visualized in Augmented Reality. Compared to the previous two maps, a different approach for displaying a legend has been chosen here. In contrast to display a legend on the side of the map, the legend in this map is displayed directly into the map.

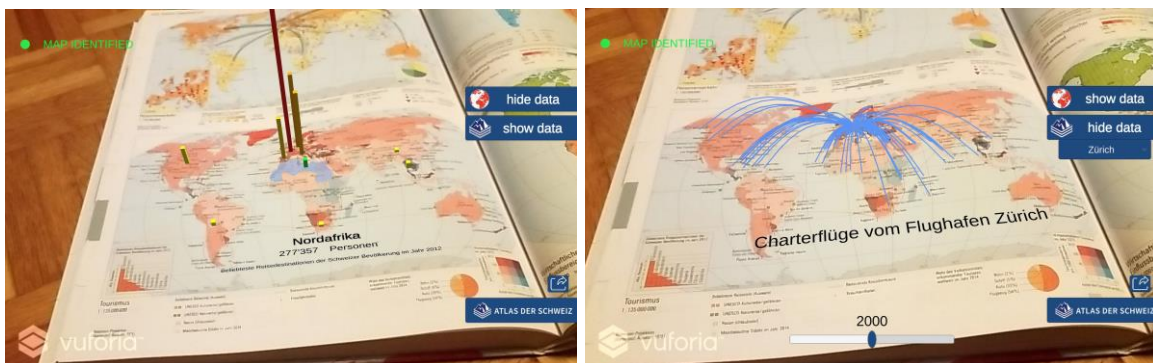


Figure 19: AR view of the tourism map of the world (left: 3D bars; right: 3D trajectories)

In the first visualization, 3D bar charts represent the most popular destinations for Swiss citizens (Figure 19: left). For that matter, the height of the bar is set according to the exact value of Swiss travelers. When clicking on one of the bars, the bar is highlighted in green. Additionally,

the territorial scope is highlighted and the number of Swiss travelers in that territory displayed in the legend.

The second visualization shows charter flights from different Swiss airports in form of 3D trajectories (Figure 19: right). Therefore, two new input interactions are implemented: a time slider and a dropdown menu. With the dropdown menu, the user can choose one airport out of the biggest Swiss airports. With the time slider, it is possible to select the charter flights from the year 1985 to 2016 and display them in different colors on the map.

4 Discussion

The implemented prototypical application shows that it is more than feasible to use Augmented Reality in classes. Especially, tablets in combination with a printed school atlas can be used very well to display additional thematic information. The implemented application furthermore shows that animations of augmented objects as well as interactions with these enrich the printed maps significantly.

During our project, many challenges were faced. We learned that projects with many automatically adjusted files like Unity projects need to have a good Git setup and predefined rules on how to handle files to be able to work independently on the same project. It is quite cumbersome to merge projects manually when Git has merging problems.

Time was needed for finding a way to import the GeoJSON data in the beginning of the project. The time we lost using Blender for the import would have better been invested into the finalization of the application. Fortunately, we found the right way for importing them using the GeoJSON .Net library.

Overall, we are satisfied with the quality of our maps. We managed to implement all desired augmented objects and their appearance and animations look nice and stable. Furthermore, all required functionalities and interactions are implemented and work stable as well. In contrast, we are not entirely satisfied with the result of the population density map of Europe. If we had a bit more time, we might have been able to find a better solution to handle that map which covers two pages of the atlas and causes big deviations and errors in height and scale of the augmented objects.

As Augmented Reality in today's classes is beginning to receive attention and no comparable products or techniques are on the market in the school environment, the development of such techniques has strongly to be considered. However, the use of AR must not replace existing teaching instruments, but especially serve as an additional instrument to support and motivate today's students in their learning environment.

5 Summary

Looking at our goals set in the beginning of the project, we can say that we succeeded in producing a working prototype of an AR application for the SWA. The three printed maps enhanced with two different types of virtual content for each map show the variety of possibilities that AR offers (Table 2).

Table 2: Overview of implemented visualizations

Swiss World Atlas	Augmented visualization 1	Augmented visualization 2
Switzerland, Overview Bernina area (p. 31)	Glacier extent of different years - Extruded polygons, Surfaces	Volume of selected glaciers of the measurement network - Stacked cuboids
Europe, Overview Population density (p. 68)	Population density - Point cloud	Electricity: end use by con- sumer group - Pie charts
World, Overview Tourism (p. 202)	Most popular holiday destinations of the Swiss - 3D bars	Charter flights: passengers on international flight routes - 3D trajectories

We designed a simple and nice to look at graphical user interface with a focus on simple interactivity through buttons, sliders, queries and animations. The interactive functionalities of the application are:

- Feature query
- Time slider
- Legend
- Save/share/print screenshot

With Vuforia, there is a stable and reliable way to recognize and track printed maps. On a flat map, the virtual content is placed precisely and even on slightly bend pages, the accuracy is sufficient. For strongly bend pages and pages covering two pages of an atlas, further research has to be done.

6 Outlook

Because this project is just a prototype of an AR application for the SWA, there are many opportunities to continue this work. It showed that it is feasible to produce such an application and the next step would be to test it in the actual environment that is in the schools where the SWA is used. A usability study with students from primary or secondary school can be conducted to discover if and how AR helps the students to learn from the SWA. Use cases can be tested to determine which new functionalities should be integrated into the project.

After a successful usability study, we can think about what the easiest way is, to integrate all maps of SWA into the application. There should be a way to make the import of GeoJSON data adaptable to the content of the files, a way to get a semi-automatic import function. Parameters like the coordinate reference system and the feature type can already be read from GeoJSON files and further parameters could be included in metadata files.

Different methods to track maps that extend over two pages of an atlas should be compared to figure out the best way to integrate two-page maps into the application.

The graphical user interface would have to be enhanced to include more information about the project. In addition, a help or a tutorial on how to use the application could be introduced. There should be more functionalities while still trying to keep it simple. It should use a more consistent layout that matches the design of the SWA. Furthermore, it should be a responsive design that can handle different device screen sizes.

Another idea is the gamification of the application to integrate it playfully in class. Working with maps, discovering specific details and solving questions and tasks would be rewarded with points. The teacher could set a goal to a certain topic that has to be achieved in class or as homework.

A follow-up study may examine in more detail which didactic concepts are most suitable for the developed visualizations and functions.

Bibliography

- [1] Schmalstieg, D.; Höllerer Tobias. (2016): Augmented reality: principles and practice. Addison-Wesley.
- [2] LIVEMAP on the App Store. <https://itunes.apple.com/sg/app/swis-sarena/id1112066355/?platform=ipad> (access 28.09.2018)
- [3] Live Texturing of Augmented Reality Characters from Colored Drawings - Disney Research. <https://www.disneyresearch.com/publication/live-texturing-of-augmented-reality-characters/> (access 28.09.2018)
- [4] Augmented reality company Layar gets \$1M boost, launches on Symbian | VentureBeat (2009). <https://venturebeat.com/2009/10/28/augmented-reality-company-layar-gets-1m-boost-launches-on-symbian/> (access 28.09.2018)
- [5] Rosten, E.; Drummond, T. (2006): Machine Learning for High-Speed Corner Detection. Computer Vision – ECCV 2006 Lecture Notes in Computer Science, 430–443.
- [6] Szeliski, R. (2011): Computer Vision: Algorithms and Applications. Springer.
- [7] FLANN - Fast Library for Approximate Nearest Neighbors (2013). <http://www.cs.ubc.ca/research/flann/> (access 26.09.2018).
- [8] NVIDIA RTX Ray Tracing | NVIDIA Developer (2018). <https://developer.nvidia.com/rtx/raytracing> (access 12.10.2018).
- [9] Heitzler M. (2017): Course “Cartography III” at ETH Zürich, Slides “Basics of 3D Graphics”.
- [10] Wekan — open-source Kanban (2018). <https://wekan.github.io/> (access 23.09.2018)
- [11] Welcome to the QGIS project! (2018). <https://www.qgis.org/en/site/> (access 23.09.2018)
- [12] Unity (2018). <https://unity3d.com/> (access 23.09.2018)
- [13] Visual Studio IDE, Code Editor, VSTS, & App Center - Visual Studio (2018). <https://visualstudio.microsoft.com/> (access 23.09.2018)
- [14] Vuforia | Augmented Reality for the Industrial Enterprise (2018). <https://vuforia.com/> (access 23.09.2018)

- [15] The first single application for the entire DevOps lifecycle - GitLab | GitLab (2018). <https://about.gitlab.com/> (access 23.09.2018)
- [16] Getting Started with Vuforia Engine in Unity (2018). <https://library.vuforia.com/articles/Training/getting-started-with-vuforia-in-unity.html> (access 23.09.2018)
- [17] Vuforia Target Manager (2018). <https://library.vuforia.com/articles/Training/Getting-Started-with-the-Vuforia-Target-Manager> (access 23.08.2018)
- [18] User "Freepik": Open book side view - Free education icons (2019). https://www.flaticon.com/free-icon/open-book-side-view_29634 (access 02.01.2019)
- [19] User "domlysz": Blender add-ons to make the bridge between Blender and geographic data (2018). <https://github.com/domlysz/BlenderGIS> (access 17.10.2018)
- [20] User "GeoJSON-Net": .Net library for GeoJSON types & corresponding Json.Net (de)serializers (2018). <https://github.com/GeoJSON-Net/GeoJSON.Net> (access 22.11.2018)
- [21] Json.NET – Newtonsoft (2018). <https://www.newtonsoft.com/json> (access 22.11.2018)
- [22] IKG ETH Zürich: Atlas der Schweiz - Volumen ausgewählter Messnetzgletscher (2018). <https://www.atlasderschweiz.ch/de/volume-of-selected-glaciers-of-the-measurement-network/> (access 01.12.2018)
- [23] Dictionary by Merriam-Webster (2018). <https://www.merriam-webster.com/> (access 28.09.2018)