

Exploring the potential of an Augmented Reality sandbox for geovisualization

Rebekka Bärthele ^{a,*}, Stefan Sauer ^a, Jan Wilkening ^a

^a University of Applied Sciences Würzburg-Schweinfurt – Rebekka Bärthele - rebekka.baerthele@study.thws.de, Stefan Sauer – stefan.sauer@thws.de, Jan Wilkening – jan.wilkening@thws.de

* Corresponding author

Abstract:

Geovisualization provides theories, methods, and tools for visualizing processes on the earth's surface. One of the key challenges of the field is to find interesting and engaging ways to display geographical data. In this paper, we focus on the Augmented Reality (AR) Sandbox. An AR Sandbox combines a box filled with sand with a 3D camera and a projector to explore the topography of the sand terrain in real time and allow water flow simulations. This direct interaction between user and topography makes the AR sandbox a fascinating, explorative tool in the field of geovisualization.

In this paper, we give an overview about how an AR Sandbox is built, what components are required, and how configuration, calibration or adding additional features work. We also discuss ways to improve the usability of the AR Sandbox or change its appearance to make it more suitable for specific use cases in geovisualization. Of particular importance is the DEM tool, which allows real-life data to be remodeled in the AR Sandbox to explore the terrain, create rainfall simulations, or export the current terrain as a DEM. Further advantages of using an AR sandbox for geovisualization are the hands-on methodology and gamification aspects, which motivate people to interact with topography and improve learning effects in the geosciences.

Keywords: augmented reality, sandbox, geovisualization

1. Introduction

The AR Sandbox was developed more than 10 years ago and still remains a fascinating and popular tool in the geosciences. As of last year, there were already over 800 AR Sandboxes built and registered worldwide, and probably many more that have not been registered yet.

The AR Sandbox uses a depth camera to continuously scan the surface of a box filled with sand, and transmits this data to a projector, which then projects a topographic map onto the sand. The map works in real-time, so that any change made in the sand-surface is recognized, and the projected map adapts within one second. It also includes a built-in hand recognition algorithm that allows the users to create a rain simulation by holding their hand over the sand surface. The AR Sandbox was developed at the University of California and is available on the developers' website as open source software (Kreylos, 2016).

As geovisualization is a relatively new field of study, new approaches to showcase its possibilities are very important to establish this subject in the scientific community and attract new students and stakeholders. Explorative approaches like the AR Sandbox, which provide a new and exciting way to interact with geodata, are therefore important tools for geovisualization.

The goal of our project was not only to build the AR Sandbox, but also to see which other available functions

are available to make the AR Sandbox even more valuable for geovisualization.

1.1 Development and applications

The AR Sandbox was developed by Oliver Kreylos, who was inspired by two prior projects with a similar approach. Both projects had their focus on entertainment and gaming aspects rather than on geographical schooling, and their development was eventually discontinued (Monobanda, 2011; SMARTmania.cz, 2011). Kreylos' version was the first one to be fully available for free and documented for anybody to use. He provides the full source code including instructions on how to build the AR Sandbox on his website. Furthermore, he started an online forum, which is active to this day, discussing additional functions and assisting with the installation process. There is also plenty of additional code written by other developers shared on this platform, which can add more functionality to the AR Sandbox.

The AR Sandbox is currently mostly used in schools or universities for teaching and advertising purposes. There is also an out-of-the-box solution available (Augmented Reality Sandbox - see AR-Sandbox.eu) and research has been done to explore its use for military purposes (Amburn et al., 2015).

1.2 Previous research

Since the AR Sandbox is a relatively new tool, there are only a few studies that have investigated the learning effects of the AR Sandbox.

One study (Soltis et al., 2020) measured the electrodermal activity of students using the AR Sandbox. The results suggested that the students' engagement during the use increased and stayed at a high level when the students were given a specific task to solve using the AR Sandbox. The students also stated that the AR Sandbox helped them understand topographic maps better.

In another study, almost all of the subjects positively evaluated the use of an AR Sandbox in the fields of geography and agriculture (Vaughan et al., 2017). Both studies had a relatively small number of subjects (between 30 and 70) and were, among other things, based on self-disclosure. Hence, these studies do not provide a clear indication about the AR Sandbox's effect on learning. However, they can give a first hint that students responded positively to the AR Sandbox in a learning environment.

2. Building an AR sandbox

The first step of building an AR Sandbox consists of assembling the necessary hardware components and making sure that they fulfil the requirements of the AR Sandbox. The process of installing, configuring, and calibrating the AR Sandbox is described on the AR Sandbox Website (Kreylos, 2016).

2.1 Components

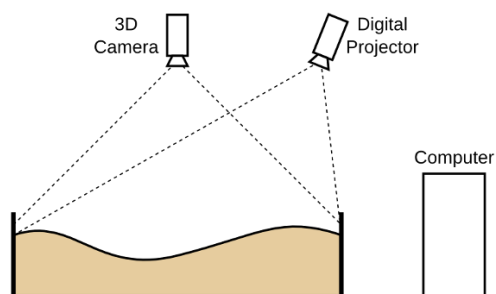


Figure 1: Schematic structure of the AR Sandbox. Image taken from the AR Sandbox Website (Kreylos, 2016)



Figure 2: Final setup of our AR Sandbox.

The main components of an AR Sandbox can be seen in Figure 1. For building an AR Sandbox, one needs a computer with a Linux Mint 19.3 Operating System, on which the sandbox-software can run. This computer should have a high-end graphics card to render the water simulation smoothly and in real time. For this setup, we used the NVIDIA GeForce GTX 1060 graphics card. In addition, a 3D-camera is needed to scan the sand surface. The AR Sandbox runs exclusively with the first version of the Kinect 3D-Camera by Microsoft. For projecting the generated map onto the sand, a short-distance projector, e.g., the BenQ MW632ST, is required. Furthermore, for the construction of the sandbox, wood is needed as well as white, reflective play sand. There are multiple versions of instructions available to construct the actual sandbox. For our project, we chose a simple wood setup (Beals, 2017). The setup was later updated to a construction that was built on a wagon to be moved easily, as to be seen in figure 2.

2.2 Software Installation

Multiple software components have to be installed to successfully start the AR Sandbox software. First, the developer recommends the Linux Mint distribution with the "Mate" desktop environment. Next, the recommended graphic driver is required since the default open-source driver by Linux is not powerful enough for the 3D graphics of the sandbox.

The sandbox software is based on another toolkit by Oliver Kreylos, called the Vrui-Toolkit, which was built to help developers with configuring 3D/VR environments. In order to communicate with the 3D camera, the Kinect 3D video package has to be installed to allow the computer to process the images from the Kinect camera. The last software component is the sandbox software itself. It contains a tool to calibrate the AR Sandbox and the software that calculates the projected image out of the Kinects 3D images.

2.3 Configuration and Calibration

When all the required hard- and software components are assembled and installed, the AR Sandbox must be manually configured. When the Kinect is positioned in a way that it can capture the entire sand surface, the base plane equation of the sand surface as well as the distance between camera and sand must be measured to later project the topographic lines with the correct orientation. Moreover, the corner points of the sand surface must be measured in the 3D image and passed on to the sandbox software to provide a bounding box for the water simulation. This passing on works with the so-called "BoxLayout" text file, that is intended for being modified by the user and is then read by the software (see Code block 1).

```
(0.0630499, -0.0443617, 0.997024), -101.2237
(-45.8689, -35.8371, -95.4177)
(48.9405, -33.1695, -97.5294)
(-46.3571, 33.916, -93.4914)
(48.2451, 34.6629, -95.9743)
```

Code block 1: BoxLayout file after configuration. Line 1: Base plane equation and distance between camera and sand (last value). Line 2-5: Corner points of sand surface as 3D-coordinates.

To calibrate the AR Sandbox and give the software common pass points in the projector and the 3D camera, the “Calibrate Projector” tool that comes with the sandbox software is used. The program projects intersecting lines onto the sand surface. By aligning the middle of a conventional compact disc (CD) with the crossing points of the lines as shown in Figure 3, the 3D camera detects the form and size of the CD.



Figure 3: Calibration process using a CD with the “Calibrate Projector” tool.

When the light turns green, indicating that the recognition was successful, the point can be confirmed by pushing a button. After approximately 8 confirmed points, a red cross appears and shows the accuracy of the current calibration. As soon as the distance between the crosses is below two millimetres anywhere in the sandbox, the calibration is sufficient.

All of the steps for this calibration process are described on the instructions page of the AR Sandbox website (Kreylos, 2016).

3. Functionality

After the calibration, the AR Sandbox is ready to be used. Its basic functionality contains a color-coded height map, which is projected onto the sand surface and updated every second. The basic map shows low-level terrain in green and blue, and higher-level terrain in red and white. It also draws topographic lines describing the sand structure with an interval of one centimeter. If the sand is moved around, the map and topographic lines change with a latency of one second to adapt to the new height information. The software also contains a hand-detection algorithm that

creates a flood of water when a hand is held above the sand surface (see Figure 4).

The water then reacts to the sand structure below the hand and creates masses of water according to the current terrain. The water masses also react to changes in the sand surface and thus allow the simulation of rivers, dam bursts, rain etc. Within the basic setup, there is no option to drain the water from the sandbox, so the water accumulates and only a reboot of the software allows for removing the water. The drain feature must be activated manually as described in section 4.

The 3D camera records images with a frame rate of 30 frames per second and a resolution of 600 x 480 pixels. Each pixel contains the height value of the corresponding point on the sand surface. The images are sent to the sandbox software, where they first get filtered to reduce gaps and noise in the data. With the data from the configuration, the software then creates a transformation matrix, with which the images are now matched onto the sand surface.

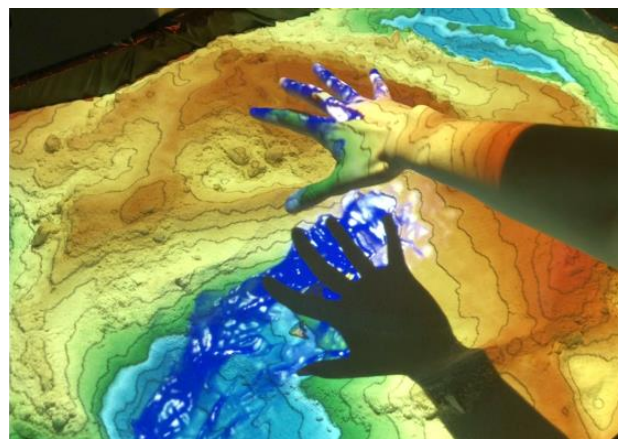


Figure 4: Creating water masses that react to the current terrain by holding your hand above the sand surface.

3.1 Post-setup adjustments

While the basic functions of the AR Sandbox are running at this point, there are certain settings that can only be adjusted once the software is running successfully. At the first start, the height map distribution might not be balanced correctly, and the ratio between low and high terrain is off. This indicates that the base level of the sand is set too high or too low. This can be adjusted in the “BoxLayout” file.

It is also possible that the measured corner points do not exactly match the sandboxes corners and there is a gap between the edge of the water simulation and the edge of the sandbox. This can also be fixed by adjusting the corner points in the BoxLayout-file.

3.2 Latency

When the sand terrain is being modified, the AR Sandbox updates the new image with a latency of one second. This is not due to the image processing or the frame rate of the camera. The latency allows the software to distinguish moving objects from the sand surface. It also stabilizes the topographic lines through a low-pass filter. Unstable lines

would not allow water masses to stay calm, since they would react to every noise in the height detection.

The recognition of moving objects is also crucial for the hand recognition algorithm. Without this recognition, hands would be classified as sand, and the topographic lines would be projected onto hands as well, which would also disrupt the water simulation.

4. Additional Functions

At this point, the AR Sandbox is functioning on a basic level. However, there are multiple options to further enhance the AR Sandbox. Some are already built into the software and just need to be activated, while others are available as separate plug-ins published by other developers. As another goal of this paper is to find specific functions that make the AR Sandbox even more practical and powerful for the field of geovisualization, the next chapter discusses and presents some available possibilities. These possibilities can be classified as follows: 1) improving usability, 2) customizing its appearance, and 3) combining it with digital elevation models.

4.1 Improving Usability

The AR Sandbox in its basic state can only be started by using the command line. It is beneficial to have the AR Sandbox start automatically when the computer is turned on, since one can then use the AR Sandbox without having to connect another monitor to the computer. The start command is therefore embedded into a desktop icon and added to the list of start-up applications to simplify starting the sandbox software. Now, it is possible for anybody to start the AR Sandbox, and no further knowledge about the command line or the software is needed.

There are also some settings that must be made every time the software is started, e.g., starting the software in full-screen mode or binding certain tools to keys to access them more easily. Some example settings can be seen in Code block 2. Kreylos recommends putting these settings into configuration files, so that they are automatically set when starting the AR Sandbox.

```
section Desktop
  inhibitScreenSaver true
section MouseAdapter
  mouseIdleTimeout 5.0
endsection
section Window
  windowFullscreen true
endsection
```

Code block 2: Excerpt from the configuration file, configuring mouse and screensaver settings.

Another feature that can be activated in the software is displaying the current sand terrain in an independent window on a screen. In this window, it is then possible to navigate freely, and the current terrain is displayed as a 3D model that updates automatically. This allows for viewing the current sandbox terrain from a different angle, for

example watching a dam break from the viewpoint of a village that is affected.

4.2 Customization

The default sandbox software comes with a standard colour scheme and render settings. By modifying the code, it is possible to adjust the visual appearance for any specific use case or preference. There are some command line tools for adjusting some settings for the appearance. For example, the command “-ucl” (use contour lines) combined with a number changes the interval of the contour lines when put after the starting command of the AR Sandbox. It is also possible to activate hill shading, change the water speed or rain strength through these commands.

The standard starting command calls the program and uses the commands „-uhm” (use height map) and -fpv (fix projector view), as shown in Code block 3.

```
cd ~/src/SARndbox-2.8
./bin/SARndbox -uhm -fpv -ucl 2
```

Code block 3: Command to start the AR Sandbox with its standard arguments plus the “use contour lines” command.

The color of the simulated water is specified in the “SurfaceAddWaterColor” shader script and can also be modified to give the water a different appearance (e.g., lava, ice, snow...). Instead of changing the script manually, we used a plugin that contains additional functions for the AR Sandbox, among them the function to toggle through multiple water appearances by pressing the assigned key (RiverWeyTrust, 2020/2022). This also allows changing the appearance while the AR Sandbox is running, since changing the script by hand would require rebooting the software each time.

The assignment of keys is achieved with an input graph, a text file containing multiple sections in which specific tools are called and binded to a key. There are multiple functions that are already built into the software, e.g., draining the sandbox from all water, creating rain over the entire surface, or creating rain locally where the cursor on the second monitor is positioned. These tools can be directly assigned by typing their names. External tools that are not already built into the software can be called by using their path and the tool class “ScriptExecutorTool” (see Code block 4).

```
section Tool15
  toolClass LocalWaterTool
  bindings ((Mouse, 5, 6))
endsection
section Tool16
  toolClass ScriptExecutorTool
  bindings ((Mouse, 9))
  executablePathName ../../weather2.sh
  arguments ()
endsection
```

Code block 4: Assigning the built-in LocalWaterTool (section 15) and the external tool “Weather2” (section 16).

The colors in the projected height map are defined in a cpt-file containing four columns: The first one describes the end of a height interval, while the other three contain an RGB value for that interval. The intervals and colors can be changed by modifying this file. The name of the file must remain “HeightColorMap” to be correctly referenced in the input graph. Therefore, it can only be modified between sessions, and not while the AR Sandbox is running. Different examples for an alternated visual appearance of the AR Sandbox can be seen in Figure 5.

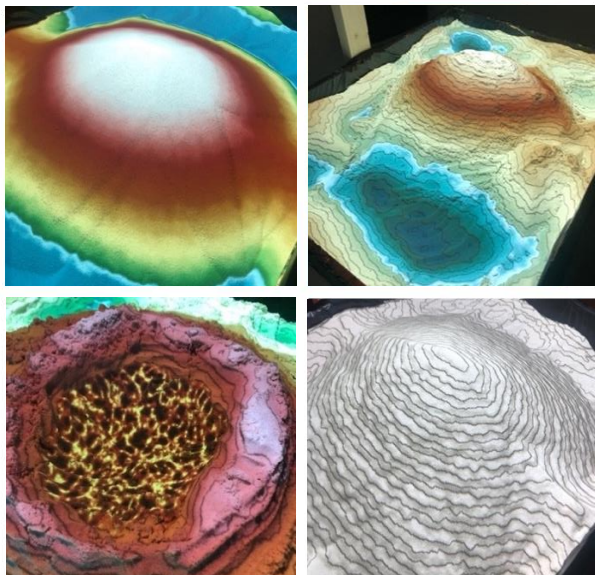


Figure 5: Examples for different appearance settings for the AR Sandbox.

4.3 Combination with digital elevation models

Digital elevation models (DEM) are used regularly in geovisualization for terrain analysis or creating 3D models, as they represent the terrain of an area. In itself, the AR Sandbox does only depict fictional data that is created by the users modifying the sand surface. By including DEMs into the AR Sandbox, we expand its functionality by allowing it to analyze actual terrain data. This makes the AR Sandbox especially valuable for geovisualization, since this field mostly deals with depicting existing terrains in different ways.

To import a DEM into the AR Sandbox, it must be in a specific file format that is not a common DEM file type: It must be a grid file, starting with two integers describing the overall number of points in x and y direction, followed by four integers describing the corners of the DEM. Below these entries, the individual points with their height value are listed. The DEM should not exceed the size of 640 x 480 entries, since this is the resolution that the 3D camera can capture. If the DEMs size exceeds this resolution, it is not possible to choose the area that is projected onto the sand.

Since this format is not very popular, a developer in the AR Sandbox community came up with a converter that creates a grid file out of TIFF images (Siki, 2017/2019). The converter is based on Python and GDAL (Rouault, Even et al., 2022). An alternative to this converter is to

analyze the DEM in GIS software and write the necessary information into a grid file manually.

The tool to display a DEM in the AR Sandbox is already built in but must be referenced in the input graph to be assigned to a key. The DEM view can then be toggled on and off using this key.

When the DEM tool is used, the AR Sandbox displays the difference between the current sand terrain and the imported DEM. Pixels with too much sand are displayed blue, pixels with too little sand are displayed red. Therefore, the user must shove sand from blue to red to recreate the imported DEM as seen in Figure 6. Pixels that are within a specified tolerance range are displayed in white. The extent of this range can be modified by using the “-dds” command with the desired number in the start command of the sandbox software.



Figure 6: From left to right: DEM tool before modifying the sand; DEM tool after modification; standard view after remodeling the imported DEM.

For this tool to work properly, the DEM’s ground level must be specified correctly in the “demVerticalShift” value in the input graph. Otherwise, the DEM tool shows too many red or too many blue values. Since DEMs are often too flat to represent the terrain accurately, a vertical scale is commonly used. This is also possible in the DEM tool using the parameter “demVerticalScale”. The usage of the DEM tool and its parameters as referenced in the input graph can be seen in Code block 5.

```
Section Tool12
toolClass DEMTool
demFileName ../rasterTiff.grid
demVerticalShift -3
demVerticalScale 4
bindings ((Mouse, 3))
endsection
```

Code block 5: Configured DEM tool in the input graph.

Furthermore, it is also possible to export the current sand terrain as a DEM raster (USGS file), as can be seen in Figure 7. This function is already built in and can be called using the Vru menu (opened on left mouse click) and assigning the “Save Bathymetry” tool to an unassigned key.

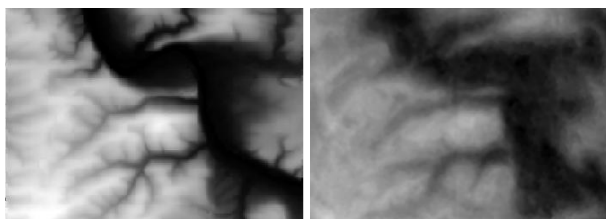


Figure 7: Left: DEM of Wuerzburg as tiff file to be imported. Right: Exported DEM after remodelling in the AR Sandbox.

5. Conclusion

In this paper, we described the steps and components needed to build an AR Sandbox. Furthermore, we explored how the AR Sandbox can be enhanced with additional functions to make it an even more valuable tool for geovisualization.

The AR Sandbox allows users to change the sand terrain and watch in real time how the topographic map changes. The hand detection algorithm allows creating rainfalls to explore how water flows in a certain terrain. There are also multiple ways to change the AR Sandbox's look and to improve its usability, to make it adaptable to any application scenario and easier to use for people without software-specific expertise.

For the field of geovisualization, the most interesting feature is importing DEMs into the software. The DEM tool bridges the gap between using the AR Sandbox as a fascinating tool, whose data is solely fictional, and using it for specific real-life applications and simulating existing terrains and water-flow scenarios.

There are also some studies suggesting the AR Sandbox has a positive effect on learning about topography and keeping students more engaged. Moreover, the AR Sandbox fascinates users and makes them excited about geographical processes and topography and allows a new form of contact with these disciplines. Since geovisualization is a relatively new subject in geosciences, such a captivating tool is very valuable to show the potential of geovisualization to future users.

Even if the official creator of the AR Sandbox, Oliver Kreylos, is currently not continuing development, there is a vast online community still actively expanding on the AR Sandbox. Hence, we hope and expect that the AR sandbox will gain publicity and become more commonly used in education.

6. References

- Amburn, C. R. et al., 2015. *The Augmented Reality Sandtable (ARES)*. DOI: 10.13140/RG.2.1.2685.0006
- Augmented Reality Sandbox—AR-Sandbox.eu*. Retrieved 1 March 2022, from <https://ar-sandbox.eu/>
- Beals, C., 2017. *Augmented Reality Sandbox will Blow Your Mind*. Retrieved 1 March 2022, from <https://www.bealsscience.com/post/2017/06/07/augmented-reality-sandbox-will-blow-your-mind>
- Kreylos, O., 2016. *Oliver Kreylos' Research and Development Homepage—Augmented Reality Sandbox*. Retrieved 20 April 2023, from <https://web.cs.ucdavis.edu/~okreylos/ResDev/SARndbox/>

- Monobanda (Director), 2011. *Project Mimicry—Ultimate sandbox game*. Retrieved 19 March 2022 from <https://www.youtube.com/watch?v=95QUvb5Sphw>
- RiverWeyTrust, 2022. *ARSandbox-Adds*. Retrieved 28 March 2022 from <https://github.com/RiverWeyTrust/ARSandbox-Adds>
- Rouault, E. et al., 2022. *GDAL (v3.4.2)*. DOI: 10.5281/ZENODO.5884351
- Siki, Z., 2019. *SARndbox_util* [Python]. Retrieved 23 March 2022 from https://github.com/zsiki/SARndbox_util
- SMARTmania.cz (Director), 2011. *Interactive Kinect sandbox*. Retrieved 1 March 2022 from <https://www.youtube.com/watch?v=8p7YVqyudiE>
- Soltis, N. A. et al., 2020. A novel approach to measuring student engagement while using an augmented reality sandbox. *Journal of Geography in Higher Education*, 44(4), pp. 512–531. DOI: 10.1080/03098265.2020.1771547
- Vaughan, K. L., Vaughan, R. E., and Seeley, J. M., 2017. Experiential Learning in Soil Science: Use of an Augmented Reality Sandbox. *Natural Sciences Education*, 46(1), 160031. DOI: 10.4195/nse2016.11.0031