Research Paper

GEOSPHERE

GEOSPHERE, v. 18, no. 2

https://doi.org/10.1130/GES02455.1

10 figures

CORRESPONDENCE:
florian.wellmann@cgre.rwth-aachen.de

CITATION: Wellmann, F., Virgo, S., Escallon, D., de la Varga, M., Jüstel, A., Wagner, F.M., Kowalski, J., Zhao, H., Fehling, R., and Chen, Q., 2022, Open AR-Sandbox: A haptic interface for geoscience education and outreach: Geosphere, v. 18, no. 2, p. 732–749, https://doi.org/10.1130/GES02455.1.

Science Editor: Andrea Hampel
Associate Editor: Clare E. Bond

Received 31 May 2021
Revision received 15 September 2021
Accepted 25 October 2021

Published online 8 February 2022

THE GEOLOGICAL SOCIETY OF AMERICA®

Open AR-Sandbox: A haptic interface for geoscience education and outreach

Florian Wellmann[1,6], Simon Virgo[1,2], Daniel Escallon[1,*], Miguel de la Varga[1,2], Alexander Jüstel[1,3], Florian M. Wagner[4], Julia Kowalski[5,6,†], Hu Zhao[5,6], Robin Fehling[1], and Qian Chen[1]

[1]Computational Geoscience and Reservoir Engineering, RWTH Aachen University, Wüllnerstrasse 2, 52062 Aachen, Germany
[2]Terranigma Solutions GmbH, Laurentiusstraße 59, 52072 Aachen, Germany
[3]Fraunhofer IEG, Fraunhofer Research Institution for Energy Infrastructures and Geothermal Systems, Am Hochschulcampus 1, 44801 Bochum, Germany
[4]Institute for Applied Geophysics and Geothermal Energy, RWTH Aachen University, Mathieustraße 30, 52074 Aachen, Germany
[5]Computational Geoscience, University of Göttingen, Goldschmidtstr. 1, 37077 Göttingen, Germany
[6]AICES Graduate School, RWTH Aachen, Schinkelstr. 2a, 52056 Aachen, Germany

ABSTRACT

Virtual reality concepts have been widely adapted to teach geoscientific content, most notably in virtual field trips—with increased developments due to recent travel restrictions and challenges of field access. On the spectrum between real and fully virtual environments are also combinations of digital and real content in mixed-reality environments. In this category, augmented-reality (AR) sandboxes have been used as a valuable tool for science outreach and teaching due to their intuitive and haptic interaction-enhancing operation. Most of the common AR-sandboxes are limited to the visualization of topography with contour lines and colors, as well as water simulations on the digital terrain surface. We show here how we can get beyond this limitation, through an open-source implementation of an AR-sandbox system with a versatile interface written in the free and cross-platform programming language Python. This implementation allows for creative and novel applications in geosciences education and outreach in general. With a link to a 3-D geomodelling system, we show how we can display geologic subsurface information such as the outcropping lithology, creating an interactive geological map for structural geology classes. The relations of subsurface structures, topography, and outcrop can be explored in a playful and comprehensible way. Additional examples include the visualizations of geophysical fields and the propagation of seismic waves, as well as simulations of Earth surface processes. We further extended the functionality with ArUco-marker detection to enable more precise and flexible interaction with the projected content. In combination, with these developments, we aim to make AR-sandbox systems, with the additional dimension of haptic interactions, accessible to a wider range of geoscientific applications for education and outreach.

INTRODUCTION

Spatial thinking in three dimensions is an essential skill for geoscientists and, consequently, an important part of geoscience education (e.g., Kastens and Ishikawa, 2006; Newcombe, 2012; Liben and Titus, 2012; Ormand et al., 2017). This skill is often taught in field studies, but the recent limitations and travel restrictions, as well as an increased awareness of limitations of inclusiveness on field trips (e.g., Giles et al., 2020) have led to a rapid development of alternative methods to educate spatial thinking. For example, virtual field trips incorporate a variety of these methods including access to digital data sets, through GoogleEarth (e.g., von Hagke et al., 2019; Whitmeyer and Dordevic, 2021), thus increasing the availability of technology to digitize outcrops and repositories of virtual outcrop models (e.g., Bemis et al., 2014; Bistacchi et al., 2015; De Paor, 2016; Cawood et al., 2017; Nesbit et al., 2020; Cawood and Bond, 2019), as well as specialized software (Buckley et al., 2019). Several studies have shown positive results using virtual outcrop models to support thinking in 3-D (Shinneman et al., 2020; Zhao et al., 2020; Bond and Cawood, 2021). Within this virtual environment, this model is considered static, but many possibilities for interaction exist—for example, for interpretation and measurements in outcrop models (Buckley et al., 2019).

As a result of these developments, virtual environments have already played an important role in geoscience education. However, they are only an end-member in a spectrum of virtualization: between real and fully virtual environments, a range of mixed environments can be considered (Milgram and Kishino, 1994). Within this range, augmented reality (AR) plays an increasingly important role with digital content draped over real features (Mathiesen et al., 2012; Carbonell Carrera and Bermejo Asensio, 2017a). One of these developments is the augmented-reality sandbox (AR-sandbox), in which, digital content is projected onto a sand surface that can be freely shaped. These systems offer a new opportunity for the learning environment: the possibility of a haptic interaction with the content.

Florian Wellmann https://orcid.org/0000-0003-2552-1876
*Present address: Exploration and Environmental Geophysics, ETH Zürich, Sonneggstrasse 5, 8092 Zürich, Switzerland
†Present address: Methods for Model-based Development in Computational Engineering, Faculty of Mechanical Engineering, RWTH Aachen University, 52062 Aachen, Germany

The AR-sandbox idea was mainly developed at University of California, Davis (UC Davis) (Jenkins et al., 2014; Reed et al., 2014; Reed et al., 2016), in a National Science Foundation (NSF)–funded project (https://arsandbox.ucdavis.edu), based on a Czech prototype system (smartmania.cz https://youtu.be/8p7YVqyudiE). This system features topography and contour maps, as well as surface water flow simulations and has been used in many educational institutes and museums worldwide (Woods et al., 2016; Kundu et al., 2017; see also the section on educational resources in the Appendix). A recent study has shown that AR-sandbox systems can be used in a variety of ways to improve learning experiences, already in the setup using topography and water flow (Hod and Twersky, 2020).

We present here Open AR-Sandbox, an implementation of the AR-sandbox system that enables an extension to a significantly wider range of additional content through an open-source software interface in the now commonly used programming language Python (https://www.python.org). We provide specific methods to access all low-level functionality in the AR sandbox but also high-level methods to generate control interfaces in the form of widgets in Jupyter notebooks (Kluyver et al., 2016) (https://jupyter.org). We also extended the haptic interaction capabilities through an implementation of ArUco markers for computer vision (Garrido-Jurado et al., 2014), which enables the placement of point and orientation data directly in the sandbox. Because Python has an extensive ecosystem of modules for geosciences and enables easy access to powerful machine learning and simulation tools, our adaptation opens up a vast set of possibilities to develop custom AR-sandbox applications for users with a reasonable knowledge of programming language.

In the discussion that follows, we will first outline the technical setup of typical AR-sandbox systems. We provide details on our Open AR-Sandbox software and its technical requirements, as well as details on the implementation of the ArUco markers. Then, we describe several native modules used in different applications for geoscientific education and outreach, specifically a link to 3-D geomodelling for structural geology and mapping classes, a module for the presentation of and interaction with geophysical fields, and a module to present dynamic content, such as landslide simulations. We also present examples for more abstract uses, such as the representation of derivatives and vector fields, as well as optimization algorithms. Finally, we show with a combination to a deep machine learning system how the link to the Python ecosystem provides access to functionalities. These functionalities enable creative and engaging use of the AR-sandbox system with a vast range of possible applications for education and outreach in particular for, but not limited to, the geoscientific community.

HARDWARE AND SYSTEM SETUP

Augmented-reality sandboxes consist of a box of sand that can be freely sculpted by hand. The topography of the sand is constantly scanned with a depth camera or sensor, and a computed image is projected back onto the sand surface, augmenting the sandbox with digital information. A typical setup is shown in Figure 1:

1. Sandbox

The basic setup is literally a box of sand. Several variations exist, in both the geometry and the material of the box, as well as the choice of sand. Boxes exist in all sizes, from small portable systems with a side length of 30–40 cm, to the typical teaching- and exhibition-size boxes with 1.0–1.5 m side length. Many commercial systems consist of boxes made of hard plastic but also wood. Transparent or semi-transparent side panels made from glass or plastic also add an interesting feature, providing an unobstructed view from the side.

For the sand itself, normal play sand can be used for most applications. However, sand mixed or coated with silicone oil (also called "kinetic sand" or "magic sand" in toy stores) can provide for better haptic interaction and does not lose cohesion over time. This type of sand can be purchased or simply mixed using normal sand and silicone oil as a cheaper and more adjustable option.

2. Sensor

The sand surface is continuously scanned with a motion-sensing device. Widely used are the Microsoft Kinect® sensors, which utilize a time-of-flight infrared detection method. The widely used method by UC Davis operates with a Kinect for Xbox 360 (Kinect 1) sensor. In Open AR-Sandbox, we also integrated drivers for the newer generation Kinect sensors, as well as Realsense sensors, such as the L515 Lidar.

3. Data Processing

The scanned signal is then processed in a computer. The bottleneck in the processing step is the availability of drivers for the respective scanners. The widely used software package by UC Davis (https://arsandbox.ucdavis.edu/instructions/hardware/) is available for Linux. Our implementation in Open AR-Sandbox operates on Linux, Windows, and MacOS.

The sensor driver provides the digital information on the sandbox elevation. This information is not yet usable in its raw form and must be processed to enable further use. In particular, the area of interest must be calibrated to the extent of the sandbox, and the height level must be adjusted. Finally, the processed elevation data can be used and combined with methods and functions to generate content based on these data. This step is described in detail in the section below. In essence, all processing methods generate an image for projection back onto the sand surface.

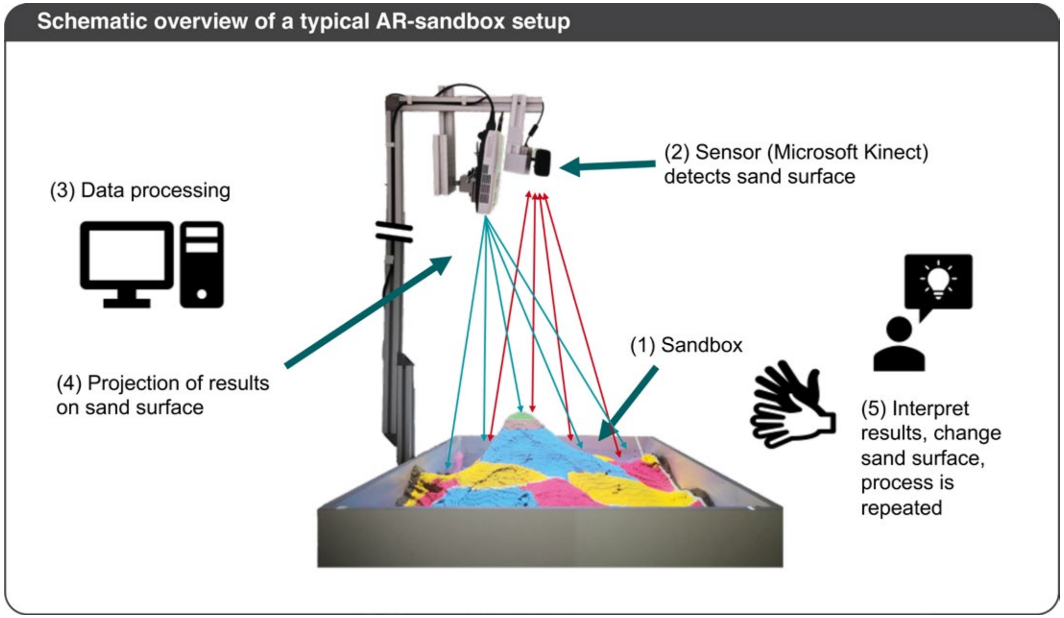


Figure 1. Setup of a typical AR-Sandbox system. For details on the different processing steps, please see the description in the text.

4. Projection

The projection itself can be performed with a standard image projector. Ideally, the field of view of the used projector should be similar to the sensor with the two instruments mounted at similar height. We tested several conventional projectors and found throw ratios between 0.8:1 and 1.2:1 to be well suited. Ultra-short throw and long-range projectors are not recommended.

5. Haptic Interaction

The essential user interaction happens directly in the sandbox itself: by first interpreting the visualized results within the 3-D scene, as well as the haptic interaction by shifting sand. The changed sand surface then triggers an update in the processing chain, resulting in an adjusted image. In this way, the user experiences direct feedback from the interaction, which is the core learning experience with the AR-Sandbox.

Additional modes of interaction have also been implemented: in the UC Davis software, a “hand wave” underneath the sensor triggers simulated precipitation at that position. Many systems also have a monitor to show additional content—or even a touch screen (as in our version) to enable precise interaction with the content. We extended the interaction possibilities further through the integration of ArUco-marker detection (see below).

Instructions on how to build these sandboxes are also available online (e.g., https://web.cs.ucdavis.edu/~okreylos/ResDev/SARndbox/Instructions.html). In addition to the standard setup, we equipped our system with an additional touch screen, fixed at the projector stand. This screen enables a simpler interaction with the software and allows for the presentation of additional content, such as cross sections, virtual wells, or even full 3-D representations of the projected material.

THE OPEN AR-SANDBOX SOFTWARE

The core of our development is the Python software package Open AR-Sandbox, a package to interact with the functionality in AR-sandboxes and provide the link to a wide range of Python packages for modeling and simulation. The package is open-source and has a modular structure, making it easy to develop new modules or add functionality. In this section, we describe the general software concept and provide details on currently implemented modules, and we show the application control and interaction in Jupyter notebooks. The software is divided into four main components: the sensor, markers, modules, and projector. In combination, these package components provide access to the complete interaction and update cycle described above. Additional technical details on the software, the package design, the installation, and the system calibration are provided in the Appendix. For up-to-date information, please also consult the project website (https://github.com/cgre-aachen/open_AR_Sandbox).

Jupyter Notebooks as Tutorials and Teaching Material

Jupyter notebooks are now widely used as front-ends for teaching material, tutorials, and scientific programming lab books (e.g., Perkel, 2018). These notebooks are based on a server-client structure, in which a programming kernel (in our case, Python; but kernels for other languages also exist) runs as a server on a local or remote machine, accessed by the client Jupyter notebook, which runs in any standard internet browser (Kluyver et al., 2016). Therefore, the clients do not require any additional software to be installed, apart from the Jupyter modules, which are now part of most standard Python distributions. The advantage of these notebooks is that they combine executable programming cells with blocks of text and images, as well as output in either printed form or diagrams and figures, which are also displayed directly in the notebook. In addition, so-called widgets, buttons or sliders to adjust parameter values, enable a high level of interactivity, even without adjusting the code itself. In combination, all of these aspects make Jupyter notebooks an excellent environment for interactive programming content.

We provide a set of Jupyter notebooks with Open AR-Sandbox, both as tutorials about the functionality and basic aspects such as calibration, and examples for the functionality, as well as full teaching material. Several of these notebooks are described below, but they are constantly extended and adapted. For a complete and up-to-date list of notebooks, please visit the online Open Ar-Sandbox CGRE github-repository (https://github.com/cgre-aachen/open_AR_Sandbox).

Interactive Control Using Panel Dashboards

Panel (https://panel.holoviz.org) is an open-source Python library that allows users to create interactive web applications and dashboards by connecting widgets to plots, images, tables, or text. For our Open AR-Sandbox system, Panel is coupled with the plotting library Matplotlib (https://matplotlib.org) to create animated visualizations projected onto the sand's surface. Then, by extending the capabilities of Panel with widgets, we can take full control of the system, to change and manipulate the image construction parameters more intuitively over a connected touch screen. This integration of widgets into the central plotting system allows for the generation of simple graphical user interfaces (GUIs) to provide access to the functionality for users without programming knowledge.

An example for such a Panel setup is shown in Figure 2 for the main plotting interaction module. This is the base module, which already contains methods to visualize topography with different colormaps and contour lines, as well as the option to superpose the effect of illumination through artificial hillshading.

Every module has its own custom-made widgets panel, allowing users to modify specific constants, change flag variables, execute functions, or display additional information that is not represented on the sand surface, such as text or several subplots. In addition, we provide a template module for a simple possibility to extend the capabilities with further custom methods.

ArUco Markers

We extended the functionality of the haptic interaction in the AR-Sandbox through the implementation of marker detections. For this purpose, we use ArUco markers (Fig. 3A), a type of binary matrix marker specifically designed for use in computer vision applications (Garrido-Jurado et al., 2014). The square binary pattern of an ArUco marker does not only encode information (e.g., a marker identification number), it also allows users to extract the position and even the 3-D orientation of a marker relative to the camera from a single image.

The implementation of markers allows for many additional possible interactions. The simplest example is the extraction of cross sections of the elevation profile along a marker line (Fig. 3B). Other possibilities are to use the markers as seed locations for optimization algorithms or process simulations, or to define the location of geophysical sources and sensors. Examples are provided in the following section.

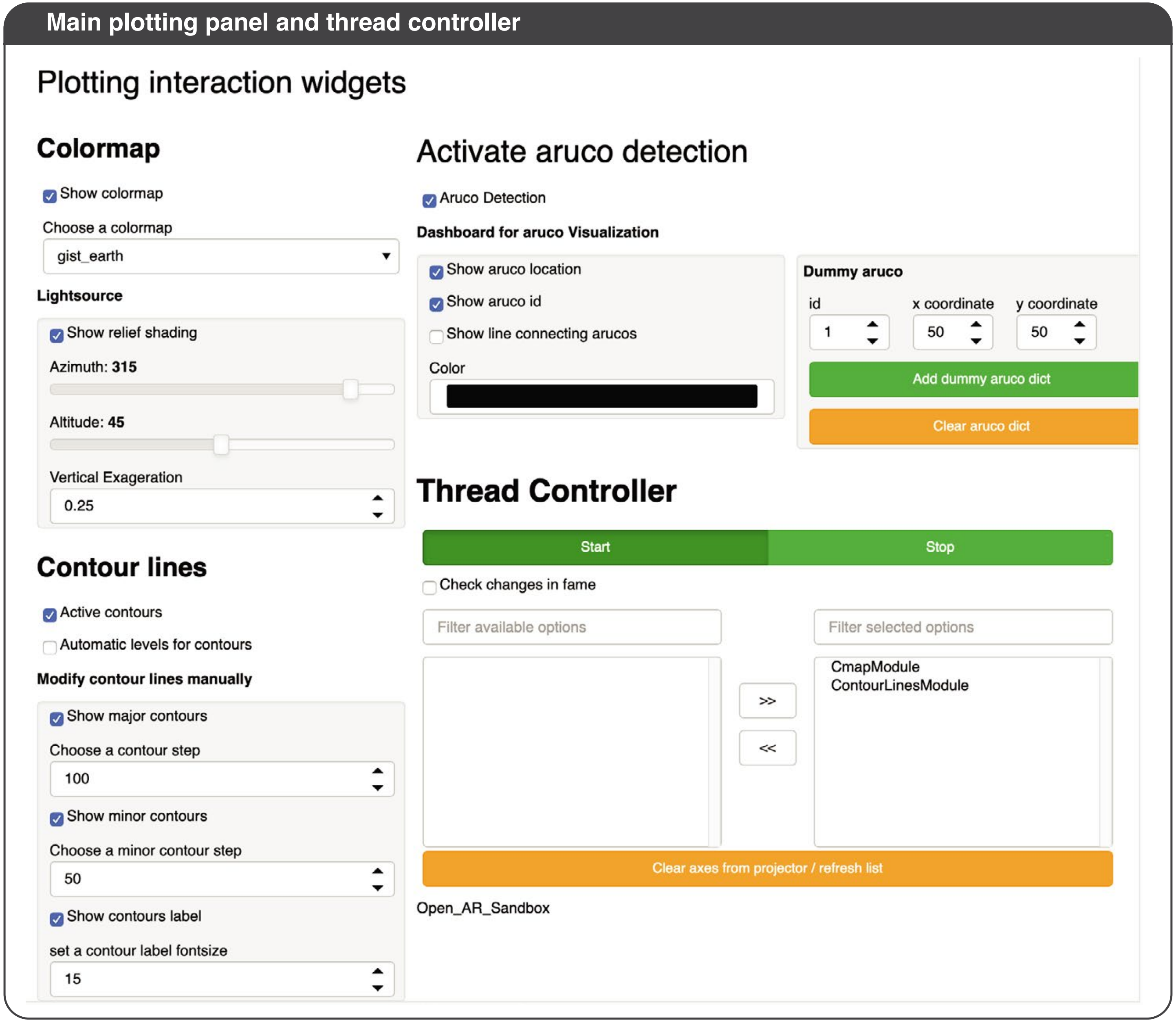
■ EXAMPLE APPLICATIONS

Visualizing Topography and Contour Lines

Visualizing and interacting with topography is a module that is also widely used in the AR-Sandbox software by UC Davis. The idea is straightforward: we use the scanned elevation field directly and project it back as a scalar field with an appropriate color scale. The elevation range can be adjusted to fit the example that is considered in each specific case. In addition to the elevation field itself, contour lines can be represented (Fig. 4A).

Even this relatively simple module enables an interesting interaction with the sandbox. It can be used to explain the meaning of contour and elevation lines in maps, and to train the mental reconstruction of 3-D shapes from contour lines. In addition, this setting is also used directly to teach geomorphological content (see link list in the Appendix for examples). We also implemented the option to calculate artificial hillshading and to add it to the projection, thus providing an additional impression of structure (Fig. 4B).

The AR-sandbox software by UC Davis enables a simulation of "artificial rain" with a hand-waving gesture. We did not implement this element at this stage, but we enable the definition of a sea level, to be visualized in the AR-sandbox (Fig. 4C). The sea level can be adjusted with a slider widget, to visualize the effect of sea-level changes.



Main plotting panel and thread controller

Plotting interaction widgets

Colormap

☑ Show colormap

Choose a colormap: gist_earth

Lightsource

☑ Show relief shading

Azimuth: **315**

Altitude: **45**

Vertical Exageration: 0.25

Contour lines

☑ Active contours

□ Automatic levels for contours

Modify contour lines manually

☑ Show major contours

Choose a contour step: 100

☑ Show minor contours

Choose a minor contour step: 50

☑ Show contours label

set a contour label fontsize: 15

Activate aruco detection

☑ Aruco Detection

Dashboard for aruco Visualization

☑ Show aruco location

☑ Show aruco id

□ Show line connecting arucos

Color: ■

Dummy aruco

id	x coordinate	y coordinate
1	50	50

Thread Controller

□ Check changes in fame

CmapModule
ContourLinesModule

Open_AR_Sandbox

Figure 2. Main plotting widgets to control the colormap and shading of the image, the contour lines of the topography, and the visualization of the ArUco markers.

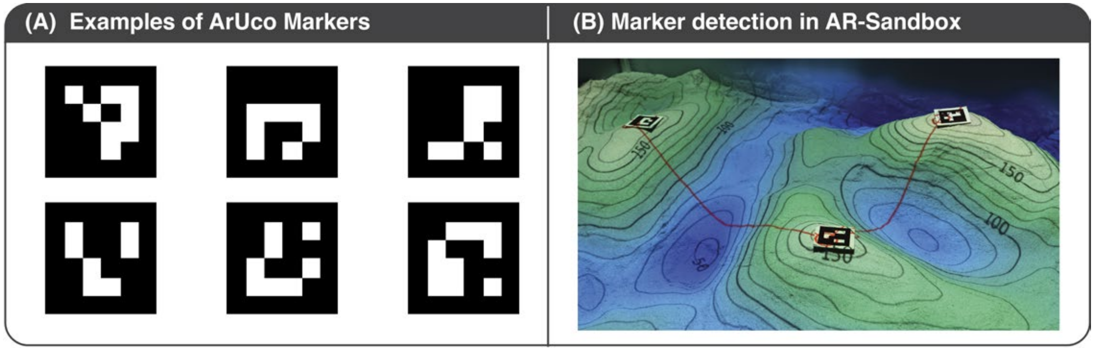

(A) Examples of ArUco Markers
(B) Marker detection in AR-Sandbox
150
100
150
100
50
150

Figure 3. (A) Example of an ArUco marker; (B) placement and detection in AR-Sandbox.

In addition to this standard setting to visualize topography, our system provides additional methods of interaction. Through the link to ArUco markers, it is possible to create cross sections through the elevation field, following the positions of the markers in the sandbox. Contour maps and profiles can also be extracted for further use. The ArUco markers can also be used to trigger different settings, for example, to change the color scheme in the sandbox or to switch contour lines on and off.

Saving and Loading Elevation Fields

Open AR-Sandbox has a module to load and save elevation fields. This module is useful to store a current elevation field for further use—for example, to generate reproducible examples for teaching and outreach. In order to re-create a previously saved elevation field in the sandbox, it can be loaded as a target field. In the next step, we provide a difference plot between the current sandbox field and the loaded (target) field. The target field can then be adjusted manually until the difference between the elevation field in the sandbox and the target field is minimized.

Structural Geology and Geomodelling

Integration of 3-D Geomodelling with GemPy

A main motivation to develop Open AR-Sandbox was to provide an intuitive and haptic access to geological mapping and modeling through integration of a full 3-D geomodelling package. The workflow to visualize geological 3-D models in the AR-sandbox is then straightforward. We start with an initialized 3-D geological model, adjusted for extent and elevation range in the AR-sandbox. Next, we calculate the geological map from the intersection between the 3-D model and the scanned sand elevation. The geological map is projected back

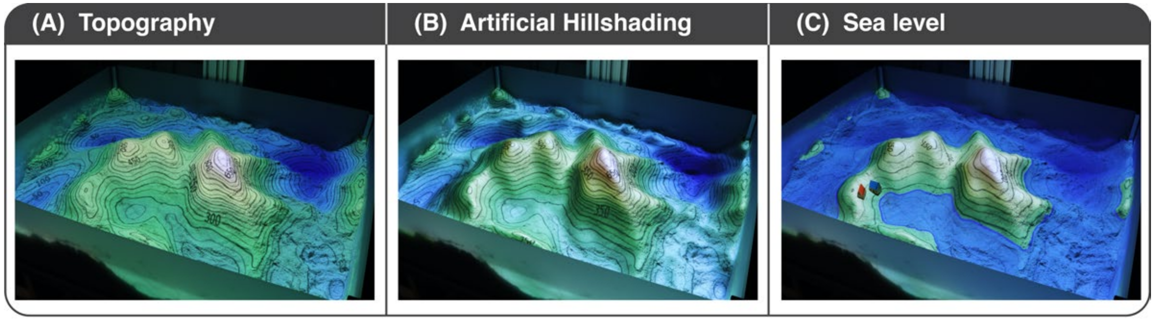


Figure 4. (A) Visualizing topography, (B) added artificial hillshading, and (C) a scene with model houses and defined sea level.

onto the sand surface. The key aspect here is that the manual change of the sand surface leads to an instantaneous update of the projected geological map. Because the calculation of the geological map is continuously updated, this leads to the impression that geological units and structures are extending into the sand volume.

The advantage of using a haptic interaction lies here in the possibility to quickly adjust the surface and then to directly observe the effect on the exposed geological interfaces (and, therefore, the changed geological map). With this approach, basic learning aims such as the determination of bed dips from the interaction with topography ("method of intersecting V's") or the effect of surface orientation on observed (apparent) dip (e.g., Bennison et al., 2013) can be quickly constructed and visualized in 3-D. Such a demonstration would also be possible using pen and paper (as in textbooks) or with a computer model on a screen. However, the interaction in the sandbox simplifies the adaptation of the elevation field through direct haptic interaction. Furthermore, important features can directly be pointed out in the 3-D scene of the projected field.

In practical use, it is instructive to start with simple geological settings, such as multiple horizontal layers where layer interfaces follow the elevation contour lines. The link between geological interface outcrop lines and topography is then directly obvious, and, with a suitable generated sand surface, it is easy to interpret the continuation of the interface below and above the sand surface. Simple quizzes can be integrated, for example, about the expected interface position at depth, which can then be verified by actually digging into the sand at this location.

We provide a set of typical geological models with increasing structural complexity with Open AR-Sandbox, as well as a variety of methods for visualization (e.g., highlighting the position of faults versus offset only; different color schemes). It is then instructive to see how students interact with the content in the sandbox, and one can discuss the typical problems encountered when interpreting 3-D geology from non-ideal outcrop conditions.

In order to enable the representation of complex geological settings such as thrust faults, dome structures, and isoclinal folds, we integrated a full 3-D geomodelling software package, GemPy (Varga et al., 2019), into Open AR-Sandbox.

The open-source geomodelling software GemPy is based on an implicit surface representation approach (see Wellmann and Caumon, 2018, for details on different 3-D modeling approaches). The interpolation of geological interfaces and orientations is performed on the basis of a co-kriging method (Lajaunie et al., 1997). The method allows for a joint interpolation of successive geological interfaces, for example resulting from continuous sedimentation processes, in one scalar field. Multiple events and unconformities can be implemented with a combination of multiple fields. In addition, faults and full fault networks can be modeled; these networks offset the surrounding geological interfaces (and other faults). For more details on GemPy and for examples and use cases, please see the project website (www.gempy.org).

From Maps to Models

The link between GemPy and Open AR-Sandbox enables the generation of geological scenarios for the widely used teaching materials from classes on geological maps (e.g., Bennison et al., 2013; Roberts, 2013). As an example, we describe here the transformation of one of these typical mapping exercises into the framework of the Open AR-Sandbox, with a dipping layer-stack from Bennison et al. (2013), in a typical workflow for model setup and interaction in class (Fig. 5A):

(1) **Digitize map and create model**: The first step is to first reproduce elevation and 3-D geological model from the teaching material. This step can be achieved through methods provided in the additional Python package GemGIS (https://github.com/cgre-aachen/gemgis); these methods enable the generation of input for GemPy from geographic data, for example from digitized geographic shapes in the GIS software QGIS or ArcGIS. With these methods, we obtain the digital elevation model for the teaching example, as well as the 3-D geological model in GemPy-format (Fig. 5B).

(2) **Reproduce elevation field**: The next step is to define the model lateral and vertical extent and to reconstruct the elevation field in the sandbox, using the difference plot representation described above (Fig. 5C).

(3) **Examine**: The geological model can then be visualized in the AR-sandbox for examination and interaction. In a classroom context, the first instructive step is to compare map and 3-D view (Fig. 5D). The possibility to interact with the geological map in 3-D quickly leads to a more visual interpretation of maps and their link to full 3-D models. As additional material, the full 3-D geological model can be visualized on a separate screen.

(4) **Interact and modify:** As described above, the geological model can then be investigated through a haptic interaction with the sand surface (Figs. 5E and 5F). In addition, ArUco markers can be used to define positions of cross sections, which are extracted from the GemPy model and can be displayed on the screen, also possibly for comparison with previously constructed cross sections by the students.

Additional aspects of interaction could be the extraction of "virtual wells" from the GemPy model at specific locations (also visualized on the screen), or the extension of the model extent to discuss aspects of extrapolation and continuation in space, to predict the expected position of an interface in an area where the model is not yet projected, and then to verify the prediction with an extension of the model projection. The accuracy of the prediction could even be quantified through the positioning of an ArUco marker.

Saving and Exporting Models to Generate New Teaching and Training Sets

In extension to the implementation of existing teaching materials, the Open AR-sandbox system can also be used to interactively generate new teaching material. For example, it is possible to adjust the elevation model for one of

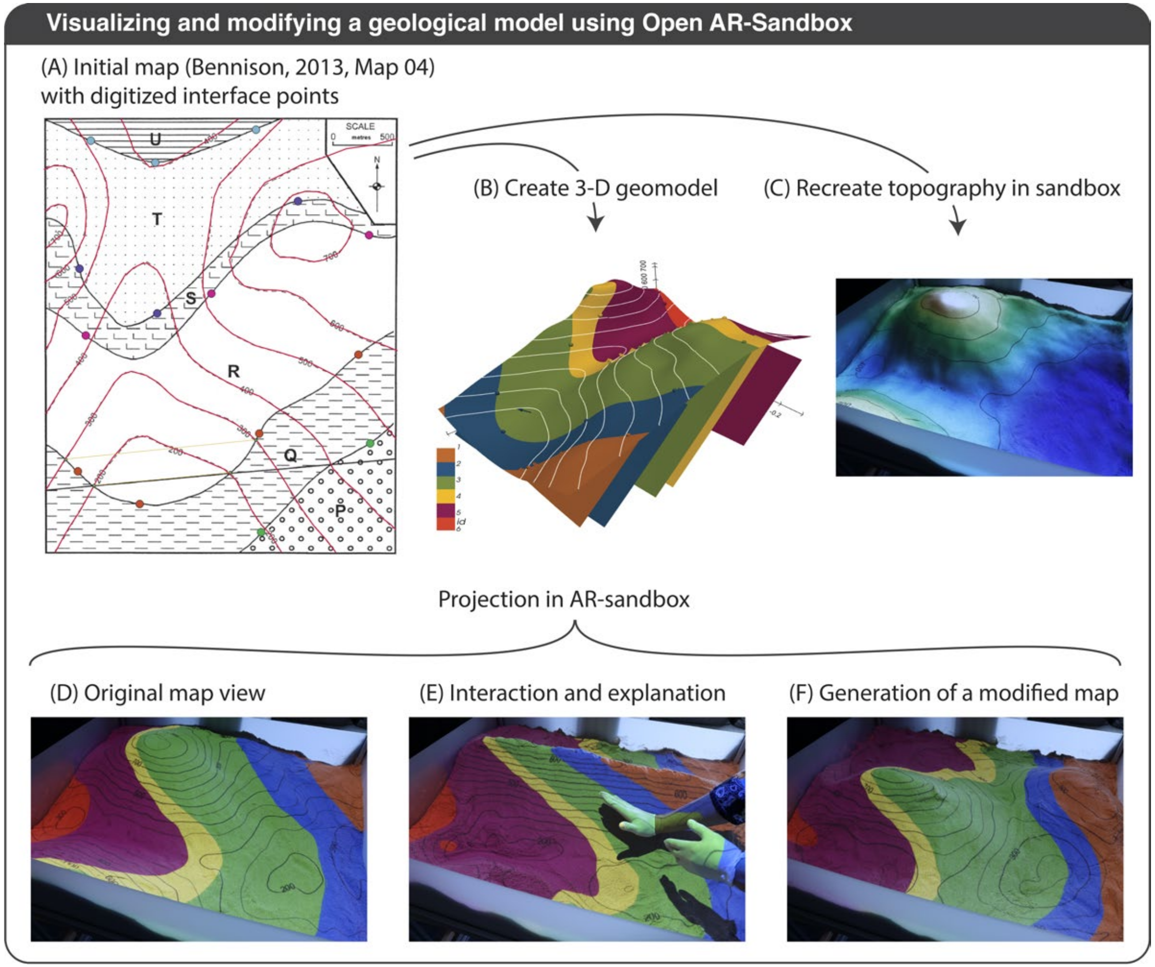

Visualizing and modifying a geological model using Open AR-Sandbox
(A) Initial map (Bennison, 2013, Map 04) with digitized interface points
SCALE
0
metres
500
N
U
T
S
R
Q
P
(B) Create 3-D geomodel
(C) Recreate topography in sandbox
Projection in AR-sandbox
(D) Original map view
(E) Interaction and explanation
(F) Generation of a modified map

Figure 5. Geological model representation: (A) digitization in GIS (example here from Bennison et al., 2013). (B) Reconstruction of topography in AR-Sandbox. (C) Generated 3-D model. (D) Geological map, calculated from 3-D geomodel and topography, projected in AR-Sandbox, view similar to original map. (E) and (F): modified topographies and updated geological map projections.

the typical teaching examples by simply generating an elevation in the sandbox and then extracting the newly generated geological map with elevation contour lines. This approach would lead to a different geological map but for the same 3-D geological model—for instructive use in class or as a basis for adjusted or even individualized examination materials.

The tight interaction with GemPy also enables the generation of new mapping exercises, based on an individual geological model through the haptic adjustment of the model topography, as an alternative to artificial elevation models generated with numerical methods. The advantage of the haptic interaction here is that the topography can be sculpted and adjusted by hand, and the updated geological map becomes visible, enabling the generation of maps where certain geological features are easier—or also more challenging—to see and interpret.

In addition to the haptic adjustment of the elevation field, the combination of ArUco markers with the geomodelling software GemPy also opens up the possibility for a haptic manipulation of the geological model itself. For example, it is feasible to adjust the position of interface points through ArUco markers in the model and therefore to change layer thicknesses or fault positions. Due to the detection of the angle, it is also possible to adjust the layer or fault dips with ArUco markers.

Real-World Geology in the AR-Sandbox

The integration of geomodelling with the AR-sandbox is not limited to teaching material, but can be used to re-generate real-world geological settings. In order to integrate real-world settings into the sandbox, we can follow the steps described above ("maps to models"), from 3-D geological model creation to the reconstruction of the elevation field. Caution must be taken that the lateral and vertical model extent can feasibly be represented in the sandbox (a matter of scaling between the required level of detail and sand grain size), and that the geometric elements, especially large gradients of cliffs and rock faces, can actually be reconstructed. The choice of sand here can play an essential role (see above). In a real-world setting, it would also be possible to project a landscape image, for example, from satellite photography, onto the sand surface, to add an additional level of realism.

Geophysics

In addition to the natural interpretation of the sandbox surface as an elevation field in the previous examples, we can also interpret it as an abstract field of values—opening up an entire additional range of possibilities to interact with visualized content. One possibility is to interpret the sand surface as a parameter field for a subsequent simulation of physical fields and processes. Results of the simulation can then be projected directly back onto the sand surface. In the following discussion, we will present some implemented methods, again made possible through the link to additional Python packages, enabling the simulation methods.

Geoelectrical Fields

For the simulation and visualization of geoelectrical fields, the sand elevation map is interpreted as a distribution of electrical resistivity, while ArUco markers can be used to position current source and sink. The widget shown in Figure 6 allows users to define the minimum and maximum values of electrical resistivity used for linear scaling.

The sandbox resolution and the resistivity field are subsequently used to simulate electrical current flow based on a finite-element approach leveraging upon the geophysical modeling and inversion library pyGIMLi (Rücker et al., 2017, https://www.pygimli.org). To facilitate a quick visual response after interaction with the sand, the finite-element mesh is coarsened by a user-specified factor compared to the resolution of the sandbox sensor, and the resistivity distribution is linearly interpolated onto this coarser mesh. The resulting potential field is then projected onto the sand overlain by streamlines depicting electrical current flow (Fig. 7A). Additional ArUco markers may be used to place potential measuring electrodes that allow calculation and visualization of the characteristic sensitivity patterns of commonly applied four-point measurements such as dipole-dipole or Wenner configurations. Given the broad applicability of the underlying Poisson equation, this module can also be used to teach the principles of Newtonian gravity, stationary hydraulic flow, or heat conduction. The existing interface to pyGIMLi can be extended readily to the use of its other physical forward operators, e.g., seismic and/or georadar raytracing or flow and transport simulations.

Seismic Wave Propagation

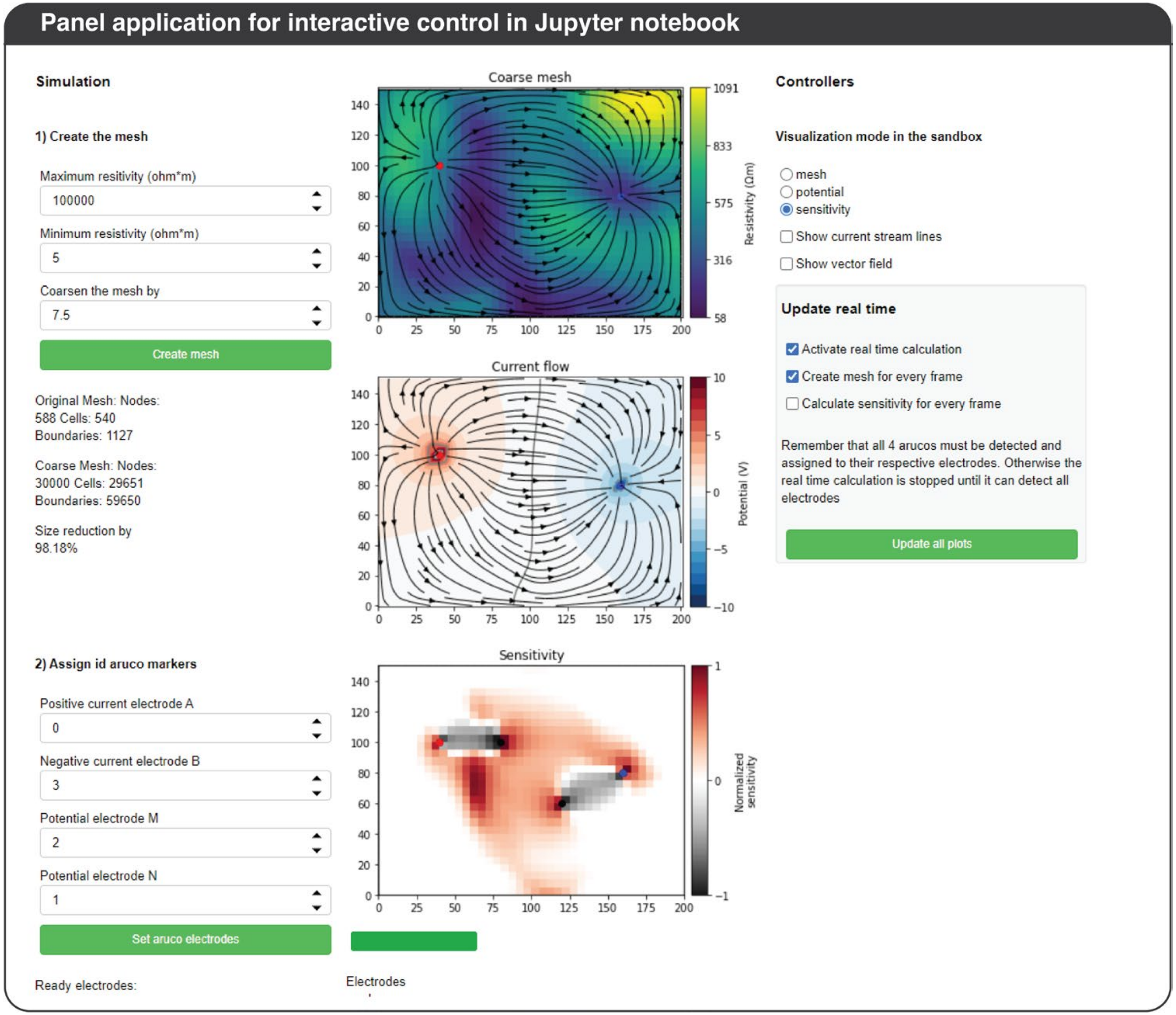
The elevation of sand can be interpreted as a seismic velocity distribution, which in turn allows simulation of the propagation of seismic waves through a heterogeneous 2-D medium. We solve the acoustic wave equation using a finite-difference approach implemented in the open-source Python package Devito (Louboutin et al., 2018, https://www.devitoproject.org). ArUco markers allow positioning of the seismic source, which could represent the epicenter of an earthquake for example. The resulting amplitudes over time are then projected as an animation back onto the sand (Fig. 7B).

Mathematical Methods and Fields

In addition to the abstract interpretation of the sandbox elevation as a parameter field for geophysical simulations, we can also use it to visualize mathematical concepts in a similar way.

Vector Fields and Derivatives

The tight integration of Open AR-Sandbox with other Python modules enables a direct calculation of derivatives on the basis of the elevation field. These fields can then be used to calculate and visualize several geometric aspects, such as gradients in different directions (Fig. 8A), or they can be used to visualize vector fields and streamline plots. Similar to the analysis of geophysical fields, these representations can then be used to discuss geometric elements on a purely abstract level. However, the link to geoscientific questions is also interesting, for example in a geomorphological context and questions of erosion and deposition. Higher-order derivatives can be used to discuss curvature or, in combination with diffusion models for landscape



Panel application for interactive control in Jupyter notebook

Simulation

1) Create the mesh

Maximum resitivity (ohm*m): 100000

Minimum resistivity (ohm*m): 5

Coarsen the mesh by: 7.5

Original Mesh: Nodes: 588 Cells: 540 Boundaries: 1127

Coarse Mesh: Nodes: 30000 Cells: 29651 Boundaries: 59650

Size reduction by 98.18%

2) Assign id aruco markers

Positive current electrode A: 0

Negative current electrode B: 3

Potential electrode M: 2

Potential electrode N: 1

Ready electrodes:

Controllers

Visualization mode in the sandbox

○ mesh
○ potential
● sensitivity

□ Show current stream lines

□ Show vector field

Update real time

☑ Activate real time calculation

☑ Create mesh for every frame

□ Calculate sensitivity for every frame

Remember that all 4 arucos must be detected and assigned to their respective electrodes. Otherwise the real time calculation is stopped until it can detect all electrodes

Figure 6. Panel application to control the geophysical field simulation, combining input fields, explanation, and output. Similar Panel widgets are also available for the other example applications in Open AR-Sandbox.

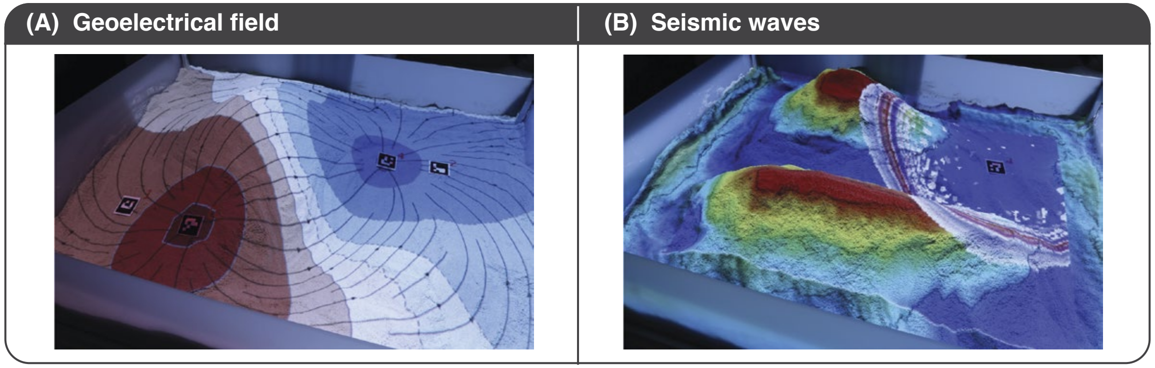


Figure 7. (A) Representation of geoelectrical field, ArUco markers control positions of electrodes; (B) Simulated seismic wave field with ArUco marker to denote trigger location.

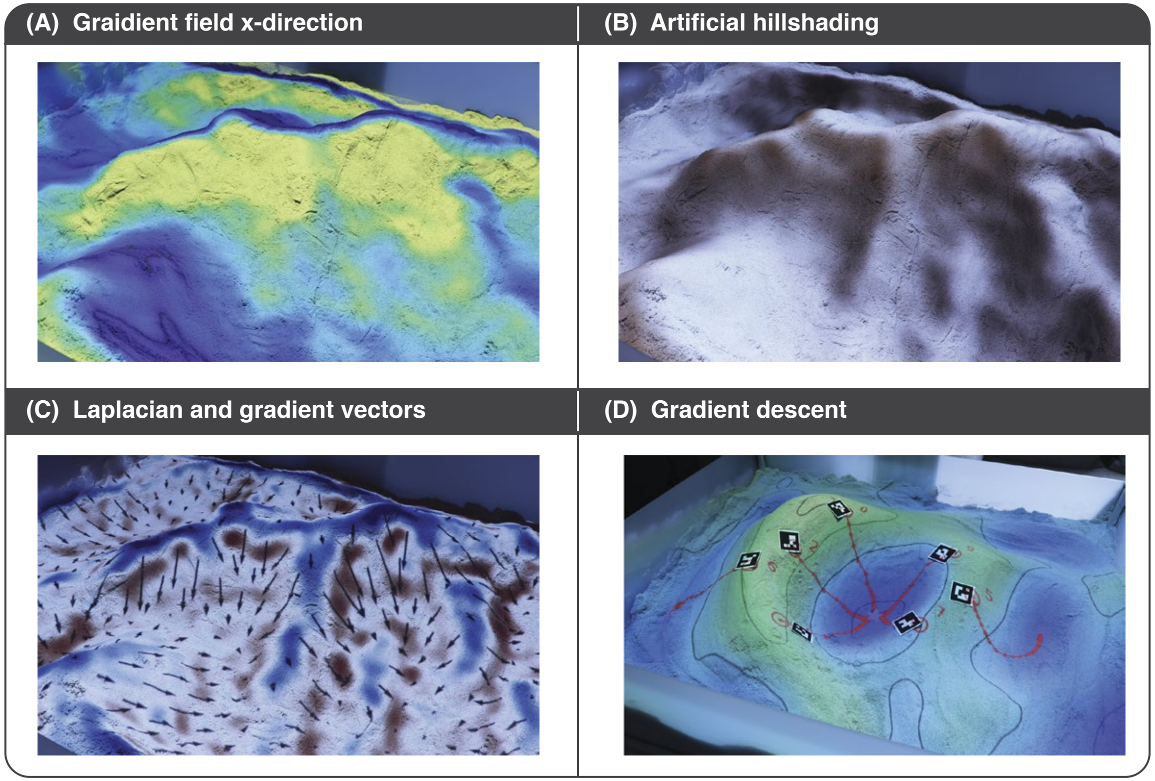

(A) Graidient field x-direction
(B) Artificial hillshading
(C) Laplacian and gradient vectors
(D) Gradient descent

Figure 8. Representation of fields and derivatives: (A) Gradient field in x-direction; (B) calculated artificial hillshading; (C) Laplacian and gradient vectors; (D) Visualization of gradient descent.

evolution (e.g., Chen et al., 2014), the Laplacian field can be calculated and visualized in the sandbox (Fig. 8C).

Optimization Algorithms

Another, more abstract, possibility is the interpretation of the sandbox axes as parameter axes, and the elevation as a cost function or a joint distribution in the context of optimization or calibration for two parameters. Optimization algorithms play an increasingly important role in geoscientific applications—they are ubiquitous in many geophysical investigations (e.g., Oldenburg and Li, 2005; Rücker et al., 2017) but also in many geological modeling studies (e.g., Wellmann and Caumon, 2018) and in all related machine learning applications (e.g., Dramsch, 2020). As an aide to teaching the potential and limitations of typically used algorithms and the required geometric aspects, two-dimensional parameter fields can be presented in the sandbox, also with the possibility to sculpt parameter fields where algorithms fail to detect the global minimum (for example, with multiple local minima) through haptic interaction. ArUco markers can ideally be used to define initial parameter values.

Simple optimization concepts such as gradient descent can be readily understood in this geometric context: locate the minimum from a given starting position. At this step, it is instructive to compare the effect of different starting positions and the difficulty to detect the global minimum in the presence of local minima (Fig. 8D). An additional simple, but mathematically more evolved, method to visualize is the concept of conjugate gradients.

In addition to the deterministic methods, we also implemented probabilistic sampling algorithms (e.g., Brooks et al., 2011) in Open AR-Sandbox. In the context of the sandbox, the determination of the posterior density in a probabilistic inversion corresponds to a reconstruction of the elevation field. The prime example is the Markov chain Monte Carlo algorithm (Metropolis et al., 1953; Brooks et al., 2011), which samples the parameter space in an iterative, stepwise manner. We integrated the classical Metropolis-Hastings algorithm (Metropolis et al., 1953), as well as an adaptive approach, where the step-size is continuously adjusted (Haario et al., 1999), and Hamiltonian Monte Carlo (HMC), a sampling method that utilizes derivative information and a link to Hamiltonian dynamics (e.g., Neal, 2011; Betancourt, 2017). Starting from an initial position defined with an ArUco marker, the step-wise sampling can be visualized in the AR-sandbox, together with the emerging density field, which can then be compared to the actual elevation model.

Landslide Simulation

Taking advantage of the possibility of loading and saving elevation fields, we can create or re-create real-world scenarios based upon which topography-related natural hazards can be assessed, for instance the risk due to rapid mass movements, such as landslides or snow avalanches. Existing snow avalanche and landslide simulation tools (e.g., Christen et al., 2010; Mergili et al., 2017) require a digital representation of the topography, knowledge about the mobilized material (position and magnitude), as well as information on the slope's surface properties. Because the actual simulation consists of a finite volume solution to an underlying mathematical process model, the simulation does not evaluate instantaneously and must be offloaded.

From the Open-AR-Sandbox, we first obtain the topography (digital elevation model [DEM]) and pass it over to the landslide simulation software. There, the topography is further processed into a computational grid. The mobilized material is provided to the landslide solver as a surface polygon, referred to as the release area. The topography can be haptically selected in the sandbox by means of ArUco markers. Our workflow supports both a single-release area, as well as the selection of several release areas for the same topography, which corresponds to several simultaneous or time-shifted slope failures. Finally, we must set model parameters, such as turbulent and dry friction values. In a real-world application, it is these parameters that critically influence the outcome of any model-based hazard analysis, and they must be chosen with care based on available field observations. Any calibrated material parameters cannot readily be scaled down to the sandbox setting, such that any parameter choice for a virtualized landslide in the sandbox has an arbitrary component. Yet, the sandbox setting does allow us to investigate relative changes, such as: what happens if the friction is consistently reduced throughout the complete flow path. While these aspects could also be investigated in a fully digital model, the possibility to adjust the topography through interaction in the sandbox enables a quick evaluation of the response to different topographic features. For example, slope gradient and aspect can be changed, additional valleys created, or barriers inserted. As in the examples above, we also see a main value in the possibility to discuss these aspects directly at the 3-D object, and to perform changes directly through the manual interaction.

The digital topography representation, release area information, and surface property encoding material parameters are exported to the solver, which will calculate the spatio-temporal evolution of the mass movement's height and velocity. The latter are post-processed into relevant information from a geohazards engineering point of view, e.g., deposition area and impact pressure of a landslide as a function of its initial volume. These results are projected back to the sandbox (Fig. 9). Interactive widgets can be utilized to explore the transient nature of the landslide or snow avalanche and even gives the impression that the sandbox is running a simulation in real time.

Landscape Generation Using Deep Learning

In the section on real-world settings, we presented the possibility to project aerial landscape images onto the sand surface. But as many settings described above are in synthetic examples, it would also be interesting to have the possibility to project artificial aerial landscape images. The generation of landscape images, often including elevation models, is a main topic in itself in the field

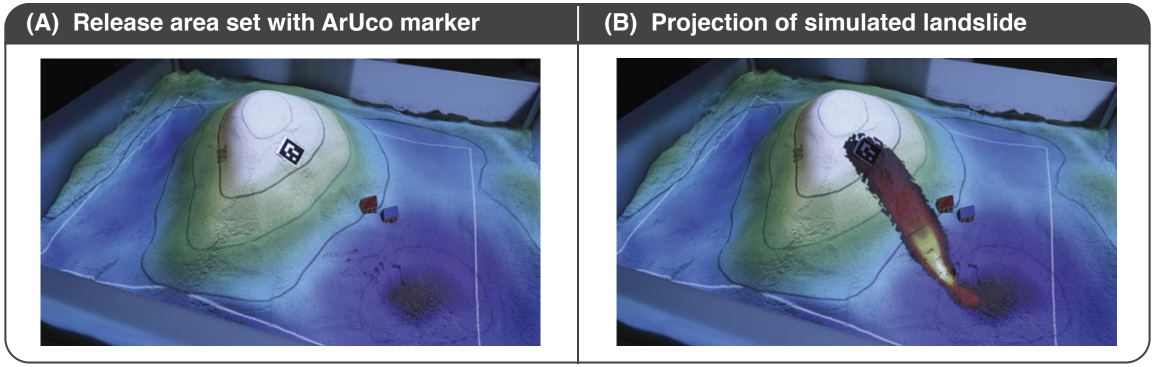

(A) Release area set with ArUco marker
(B) Projection of simulated landslide

Figure 9. Simulated landslide simulations in AR-Sandbox: (A) Definition of topography and release area with ArUco marker. (B) Projection of simulated mass flow thickness.

of computer graphics and many methods exist, for example, to generate landscapes as parts of computer games. Many of these methods use deterministic algorithms. In recent years, machine learning methods are becoming more widely applied in this field (e.g., Galin et al., 2019).

Although several of these methods could be implemented for landscape generation, also in the sandbox, we implemented another approach, adjusted to our specific setting: we trained a deep neural network for image translation (Isola et al., 2017) using the deep learning framework Tensorflow (Abadi et al., 2016) on a set of realistic digital elevation models and corresponding satellite images of the central Alps.

Example results are presented in Figure 10. After initial training, landscape predictions can be performed very quickly, in the order of seconds, allowing for a fast generation and projection of an image after adjustment of the elevation field in the sandbox.

This module could, for example, be used to show applications of deep learning on geoscientific data. In addition, the generated landscape itself provides interesting possibilities in combination with the other available modules. For example, it would be possible to combine a generated landscape image with an underlying geological model and to only view outcrops at specific locations, leading to a more realistic possibility to present and discuss the typical lack of perfect outcrop positions. Or to combine the generated satellite image with projected landslide simulations, to add more realistic elements to the scene. Also here, additional creative implementations are certainly possible.

Extending the Functionality of Open AR-Sandbox

The implemented methods provide already a wide range of visualization and interaction possibilities, but certainly there are more possibilities. We therefore implemented methods that enable an easy extension of the functionality of Open AR-Sandbox. This can be done in either of two ways: First, through the use of a dedicated prototyping module, which facilitates the creation of an external script (using your Jupyter notebook, for example) and the generation of an *update* function that receives the axes of the image, the raw frame, the location of the ArUco markers, and the extent (for scaling). This method will allow you to modify or create content and to directly visualize it in the Open AR-Sandbox. Another option is to create a separate class with methods, for example, based on the already existing modules, and to add this module to the thread. Both possibilities are explained in the online tutorial sections.

Teaching and Presentation Concepts

The AR-sandbox environment can be applied in a variety of teaching concepts. As mentioned in the introduction, many AR-sandboxes on the basis of the UC Davies system are already in active use to teach topography and geomorphology (Jenkins et al., 2014; Woods, et al., 2015; Reed et al., 2016; Woods et al., 2016). Applications range from the interactive explanation of concepts, over self-defined tasks in the interaction with the content, to complete student group projects, explaining concepts with prepared videos (see Appendix for examples). McNeal et al. (2020) performed a study on the impacts of learning with AR-sandboxes for topographic map assessment in undergraduate geography courses, and the authors found that using the sandbox increased confidence in interpreting topographic maps. On the other hand, Giorgis et al. (2017) did not find a significant gain in using the AR-sandboxes in an instructor-led exercise for purely topographic map reading. However, the authors also suggest that methods should be investigated to increase the efficacy of the AR-sandboxes, and they suggest a tighter integration into a class program at multiple instances. Hod and Twersky (2020) present a comprehensive overview of studies about the conventional topographic AR-sandbox systems in

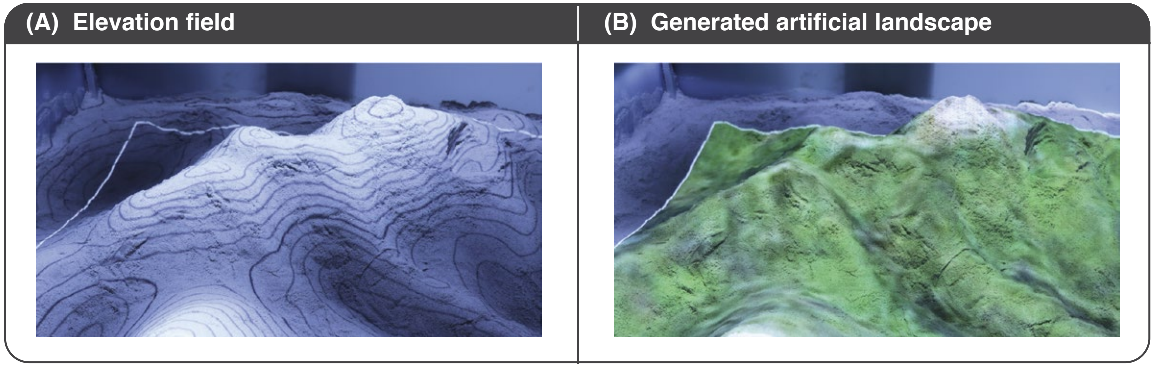


Figure 10. Artificial landscape generation: (A) Elevation field in the sandbox, which is forwarded to a Deep Neural Network to generate an artificial landscape image, which is then projected back onto the sand surface (B).

teaching geographic concepts. They conclude that improvements could be observed through interaction with the AR-sandbox. However, at times these improvements were only evident over a longer interaction time—and may thus have been missed in previous studies that only focused on single tasks. Specifically, they showed that the use of AR-sandboxes significantly increased engagement with the study object, a result also observed in a recent study by Soltis et al. (2020). The added functionality implemented in Open AR-Sandbox can be used in either self-paced or instructor-led classes. Based on the previous results on topography only, we expect an overall even more positive outcome, as the methods we integrated greatly increase the level of interaction. We have used the system in several classes, including introductory classes in computational geosciences and geological mapping and modeling classes, through the geomodelling module and the link to GemPy. Self-guided projects in this context can also include the generation of the geological model itself.

In addition, the general beneficial element of haptic interaction in education has been shown in previous studies (Minogue and Jones, 2006; Edwards et al., 2019). With the added complexity of elements in Open AR-Sandbox, it provides the basis for a wider use of haptic collaborative elements in education. We have used the system in a student project on mathematical optimization, which resulted in the implementation of the optimization methods presented above, and in a Bachelor thesis for the visualization of the generated landscape scenarios. The possibility to access the elevation field through the implementation in Python and Jupyter notebooks enabled even students with limited prior programming knowledge to use these projects, and it is motivating for students to see results in the AR-sandbox for further interaction.

Additional developments could also include the serious games in this system, for example a geological mapping exercise with limited outcrop conditions, which could be progressively exposed during the game, to train the development and testing of geological hypotheses. Such a lab-based mapping experience may also be used to teach geological thinking in a more inclusive environment (Carabajal and Atchison, 2020; Giles et al., 2020). We plan to extend our own teaching material in this direction, with the aim to develop additional teaching material in the form of Open Educational Resources (OERs), and we look forward to additional contributions from the community.

Programming Skills Development

Computational thinking and programming skills are increasingly considered important aspects for geosciences (e.g., Gunderson et al., 2020), but it is difficult to fit programming courses in already full geoscience curricula. As evaluated by Jacobs et al. (2016), a suitable approach can be an adaptation to blended learning concepts and links to geoscience applications. In their detailed assessment, Jacobs et al. (2016) discuss the use of Jupyter notebooks as an excellent environment for teaching programming. Beyond their use in applied programming courses (e.g., Cardoso et al., 2018; Reades, 2020), they have become a de-facto standard tool for open science (Kluyver et al., 2016; Randles et al., 2017; Perkel, 2018, Fangohr et al., 2021). These notebooks use a standard browser as front end for a programming kernel and combine programming code for various interpreted programming languages (Python, Julia, R, and others) with direct text-based and graphical output and additional blocks with descriptions and content in markdown format (Granger and Pérez, 2021). A range of educational tools for geosciences are already available in this format (e.g., Jacobs et al., 2016; Aiken et al., 2018, Beucher et al., 2019), and we consider the combination of programming in Jupyter notebooks with direct haptic interaction in an AR-sandbox as a possibility to create educational content for a diverse set of geoscience topics, with the additional benefit of providing hands-on programming experiences on multiple levels of coding:

(1) In fully developed control panels on a (touch) screen and through ArUco markers in the sandbox itself, requiring no to limited programming skills;

(2) Jupyter notebook code cells, which can be extended with additional code, for example, to adjust settings such as color scheme and elevation range;
(3) Through the development of completely new notebooks with additional content, based on the provided template.

With this flexibility, the system can be used to combine geoscientific content and programming on various skill levels, as an experimental system where results of the programming steps can directly be investigated in a haptic and inspirational way.

DISCUSSION

The presented examples show how a wide range of geoscientific concepts can be visualized in AR-sandbox systems, for a direct haptic interaction in augmented reality—as an additional method to teach geoscientific content, in addition to the rapid developments using Virtual Reality environments. AR-sandboxes are highly interactive and engaging tools, but they have so far been limited to elevation fields and flow simulations (Jenkins et al., 2014; Woods et al., 2015; S. Reed et al., 2016). Our implementation in the Open AR-Sandbox software shows that a wide range of additional geoscientific concepts can be visualized with these systems. The level of haptic interaction in this system is, in the current state, not possible in fully virtual environments. With the development of Open AR-Sandbox, we aim to facilitate the inclusion of haptic components into diverse geoscientific teaching activities, for example, in the context of experimental learning (Kolb, 1984; Kundu et al., 2017) or as an additional technology to teach 3-D map-reading skills (Carbonell Carrera et al., 2017b).

To open up the possibility of these creative use cases to a wider audience, we implemented an interface to the AR-sandbox systems with a specifically developed open-source package, Open AR-Sandbox, in the programming language Python. To facilitate contributions by the geoscientific community, we made the software available under a permissive LGPL license (see Appendix).

The direct integration with Python, in particular, enables an easy entry point for further developments due to the access to the growing ecosystem of geoscientific tools. A prime example of this possibility is the link to the 3-D geological modeling package GemPy. The combination of real-time modeling and updated visualization in the AR-Sandbox enables a degree of haptic interaction with geological maps and models, leading to an intuitive understanding of the 3-D content, beyond 3-D block models. At the same time, this interaction provides access to novel geological modeling concepts, which can be used beyond the visualization and interaction in the sandbox itself. This same interaction aspect is used in the presented examples of geophysical and geohazard simulations, as well as the deep learning integration. We see the Open AR-sandbox systems as an ideal entry point to these methods, in a playful teaching context, but with the potential for further use and development beyond the use in the AR-sandbox itself.

In order to increase the level of interactivity in the AR-sandbox environment even further, we implemented the detection of ArUco markers (Garrido-Jurado et al., 2014). Our examples already show the intuitive use of markers to generate cross sections and virtual drill holes, to place sources and sensors in geophysical simulations, and to use them as starting points for landslide simulations. We also mentioned the possibility to use markers to quickly swap between different content in the sandbox, for example, between different types of geological models in a classroom environment. In addition to these developments, we envision many more possible interactive elements on this basis, for example, to change the dip angle in geological layers and faults through an identification of ArUco marker orientation. Multiple markers could be used to adjust settings, for example, the vertical elevation scale or the color ramp of the used color scale. We are convinced that many more creative implementations are possible on this basis.

In the examples presented above, we show how new applications can be generated, on the basis of a set of interaction modules, in Jupyter notebooks. Because these notebooks are becoming more widely used in teaching and outreach, also in the field of geosciences, we presume that the methods implemented in Open AR-sandbox will not only be accessible to educators for generating new material, but that they can also be quickly used by students. This aspect is also an entry point to raise motivation about programming and digitization with students, an important aspect to prepare our students for the ongoing transformation in many fields of geosciences.

CONCLUSION

AR-sandboxes are a highly engaging haptic environment to interact with content in 3-D. An understanding of the 3-D nature of the Earth is essential to many societal challenges—from the sustainable use of natural resources, over infrastructure and building projects that involve the subsurface, to the occurrence and risk posed by geohazards. With the software presented here, Open AR-Sandbox, we aim to enable a new level of interaction with haptic content in addition to the fully virtual environments that are now increasingly available for geoscience education and to facilitate a wider use of these systems in education and outreach. The system furthermore supports computational skill development and provides access to the growing ecosystem of geoscientific tools implemented in Python. In this way, we also hope to contribute to the fascination around the use of digital methods, and programming specifically, to the field of geosciences.

ACKNOWLEDGMENTS

The development of Open AR-Sandbox has been supported by an Exploratory Teaching Space (ETS) grant and an Exploratory Research Space (ERS) grant "ExPARGeo" by RWTH Aachen University as part of the Excellence Strategy, funded by the Federal Ministry of Education and Research (BMBF) and the Ministry of Culture and Science of the German State of North Rhine-Westphalia (MKW) under the Excellence Strategy of the Federal Government and the Länder. Further funding came through a Digital Teaching Fellowship for Florian Wellmann by the state of North Rhine-Westphalia and the Stifterverband. We thank Hans de Bresser and an anonymous reviewer for their

careful review of the manuscript and the helpful suggestions to clarify the contribution and to highlight the potential benefits of haptic interactions for education and outreach with AR-sandboxes.

APPENDIX

License and Scientific Collaboration

Open AR-Sandbox is currently hosted on the CGRE github-repository: https://github.com/cgre-aachen/open_AR_Sandbox. However, in the interest of scientific collaboration in an open-source project, we strongly suggest to clone the repository to your system, to ensure that extensions and adaptations can later be integrated back into the main repository. If you intend to perform major adaptations, then we suggest that you create a fork of the repository, for full individual control. Please note that the open-source license LGPL requires you to make your modifications available under the same license conditions. Best practice in the community is to provide these modifications back to the main repository in the form of a pull request. You are, however, allowed to use the software in any derivative product, even in a commercial form, if you use it as it is (without modification). If you use the software in any type of scientific publication, then please also cite this manuscript.

For more information on collaboration in open-source projects and the use of git as a distributed version control system, please consult online documentation (e.g., https://guides.github.com/introduction/git-handbook/).

Open AR-Sandbox is already used in geological mapping classes at RWTH Aachen University and also at Queensland University of Technology (https://youtu.be/E9bQTNInEmY).

Existing Use-Cases and Commercial Systems (Selection)

(1) **UC Davies**: https://arsandbox.ucdavis.edu, learning materials: https://arsandbox.ucdavis.edu/wp-content/uploads/2016/11/Shaping-Watersheds-AR-Sandbox-Facilitation-Guide.pdf
(2) **USGS AR-Sandbox lesson**: https://prd-wret.s3.us-west-2.amazonaws.com/assets/palladium/production/atoms/files/AR%20Sandbox%20Tutorial_Updated%20121219%20%281%29.pdf
(3) **Playful**: mostly children's entertainment (Artistic, some geoscientific content, e.g., volcanoes): https://ar-sandbox.com
(4) **RWTH Aachen Virtual Geomorphology YouTube channel**: https://www.youtube.com/channel/UCK2bn1Gf5f-MSWWImZQ-DWA
(5) A**R-Sandbox implementation in Unity3D (MIT License)**: https://github.com/jloehr/AR-Sandbox

Package Design

All software packages work independently, but the combination of all four components allows users to have the complete cycle of acquiring the data (sand surface and marker positions), processing this information to generate an image and project this image back into the sandbox.

(1) **Sensor package**: Manages the connection with the hardware to obtain the raw information of the sand depth surface and the color image (or commonly referred to as the depth space and the color space accordingly). To date, we support the Kinect V1 sensor (in Linux) and the Kinect V2 sensor (in Windows and Linux) by using the Microsoft Kinect drivers with appropriate Python wrappers. Inclusion of more sensors can be developed in this space as long as a Python wrapper exists for the respective piece of hardware.
(2) **Markers package**: Uses the color image and ArUco markers to detect the pixel-precise position in the color image. This position then needs to be mapped to the equivalent position on the depth image because both images have different resolutions and lens distortions. The mapping is achieved using the CoordinateMapping class (https://docs.microsoft.com/en-us/previous-versions/windows/kinect/dn758445(v = ieb.10)). The outputs are the coordinates of the identified ArUco marker in the depth space.
(3) **Module package**: This component takes the information from the sensor and markers to generate an image to project onto the sand. Modules contain the logic for the different applications and are therefore of central relevance to transfer the depth information to geoscientific content.
(4) **Projector package**: This package finally generates a matplotlib (https://matplotlib.org/) figure and axes embedded in a Panel/Bokeh server (https://panel.holoviz.org) to enable a high flexibility and control over the projected images (see examples in main document). Over these axes, the modules will plot and change the figure accordingly so an image can be plotted and displayed back to the sand surface by the projector.

Calibration

For every new setup using Open AR-Sandbox, the sensor and projector must be calibrated first. The calibration of the projector depends on the resolution of the hardware and the field of view of the projector in relation to the distance to the sandbox. The projector displays a panel dashboard with the main frame for the main plotting content to be visualized on the sand surface itself (Panel plot), as well as optional areas for a legend, a profile view or an interactive control area. As a first step, we need to be sure that at least the entire sandbox including the four corner poles are covered by the illuminated area and that the projected image is square (keystone correction). Then, using Panel widgets as sliders, the plotting panel extent can be adjusted to match the borders of the sandbox by changing the vertical and horizontal origin on the upper left corner and then shifting the width and height of the main frame.

To calibrate the sensor software, we use the same method as before by using sliders to adjust the horizontal and vertical extent of the depth image, as well as the depth calibration. The horizontal and vertical calibration is used to adjust to the extent of the sandbox frame, while the depth calibration is used to indicate the maximum and minimum value of the sandbox's vertical distance away from the sensor in millimeters. At the end, we obtain a defined volume of interest, the bounding box that fits the physical dimensions of the sandbox, and the region where the sandbox will acquire and project back the information. All of these steps are integrated in a Jupyter notebook as an interactive guide to simplify the calibration process.

Pre-Processing of the Elevation Field

We perform several pre-processing steps before using the elevation field, to reduce the effect of noise from the sensors. A first step is the rearrangement between the sensor image and the elevation in the sandbox, to obtain values that are positive upwards in the subsequent processing steps (i.e., that the minimum vertical values actually correspond to the bottom of the sandbox). In order to obtain a smoother field, both a filtering (e.g., Gaussian filtering) in each time step, as well as an averaging over multiple time steps is implemented and can be configured in the sensor class. The averaging over time steps leads to some latency in the update process, but for a reasonable number of frames, this effect is barely visible.

It is also possible to "freeze" the current sandbox elevation field, through an interruption of the updating process. Such a setting may be suitable for a more careful investigation of the projected results, to avoid the detection of the user's hands, which are also detected as a change in the elevation field.

REFERENCES CITED

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., and Zheng, X.,2016, Tensorflow: A system for large-scale machine learning: Proceedings, 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), Savannah, Georgia, 2–4 November, p. 265–283.

Aiken, J.M., Aiken, C., and Cotton, F., 2018, A Python library for teaching computation to seismology students: Seismological Research Letters, v. 89, no. 3, p. 1165–1171, https://doi.org/10.1785/0220170246.

Bemis, S.P., Micklethwaite, S., Turner, D., James, M.R., Akciz, S., Thiele, S.T., and Bangash, H.A., 2014, Ground-based and UAV-Based photogrammetry: A multi-scale, high-resolution mapping tool for structural geology and paleoseismology: Journal of Structural Geology, v. 69, p. 163–178,, https://doi.org/10.1016/j.jsg.2014.10.007.

Bennison, G.M., Olver, P.A., and Moseley, K.A., 2013, An Introduction to Geological Structures and Maps: London, Routledge, 179 p.

Betancourt, M., 2017, A conceptual introduction to Hamiltonian Monte Carlo: arXiv.org preprint, arXiv:1701.02434.

Beucher, R., Moresi, L., Giordani, J., Mansour, J., Sandiford, D., Farrington, R., et al., 2019, UWGeodynamics: A teaching and research tool for numerical geodynamic modelling: Journal of Open Source Software, v. 4, no. 36, https://doi.org/10.21105/joss.01136.

Bistacchi, A., Balsamo, F., Storti, F., Mozafari, M., Swennen, R., Solum, J., Tueckmantel, C., and Taberner, C., 2015, Photogrammetric digital outcrop reconstruction, visualization with textured surfaces, and three-dimensional structural analysis and modeling: Innovative methodologies applied to fault-related dolomitization (Vajont Limestone, Southern Alps, Italy): Geosphere, v. 11, no. 6, p. 2031–2048, https://doi.org/10.1130/GES01005.1.

Bond, C.E., and Cawood, A.J., 2021, A role for virtual outcrop models in blended learning–improved 3D thinking and positive perceptions of learning: Geoscience Communication, v. 4, no. 2, p. 233–244, https://doi.org/10.5194/gc-4-233-2021.

Brooks, S., Gelman, A., Jones, G., and Meng, X.L., eds., 2011, Handbook of Markov Chain Monte Carlo: Boca Raton, Florida, CRC Press, 619 p., https://doi.org/10.1201/b10905.

Buckley, S.J., Ringdal, K., Naumann, N., Dolva, B., Kurz, T.H., Howell, J.A., and Dewez, T.J., 2019, LIME: Software for 3-D visualization, interpretation, and communication of virtual geoscience models: Geosphere, v. 15, no. 1, p. 222–235, https://doi.org/10.1130/GES02002.1.

Carabajal, I.G., and Atchison, C.L., 2020, An investigation of accessible and inclusive instructional field practices in US geoscience departments: Advances in Geosciences, v. 53, p. 53–63, https://doi.org/10.5194/adgeo-53-53-2020.

Carbonell Carrera, C., and Bermejo Asensio, L.A., 2017a, Augmented reality as a digital teaching environment to develop spatial thinking: Cartography and Geographic Information Science, v. 44, no. 3, p. 259–270, https://doi.org/10.1080/15230406.2016.1145556.

Carbonell Carrera, C., Avarvarei, B.V., Chelariu, E.L., Draghia, L., and Avarvarei, S.C., 2017b, Map-reading skill development with 3D technologies: The Journal of Geography, v. 116, no. 5, p. 197–205, https://doi.org/10.1080/00221341.2016.1248857.

Cardoso, A., Leitão, J., and Teixeira, C., 2018, Using the Jupyter notebook as a tool to support the teaching and learning processes in engineering courses, *in* Auer, M.E., and Tsiatsos, T., eds., The Challenges of the Digital Transformation in Education: Proceedings of the 21st International Conference on Interactive Collaborative Learning, v. 2: Cham, Springer, Advances in Intelligent Systems and Computing, v. 917, p. 227–236.

Cawood, A.J., and Bond, C.E., 2019, eRock: An open-access repository of virtual outcrops for geoscience education: GSA Today, p. 36–37, v. 29, no. 2, https://doi.org/10.1130/GSATG373GW.1.

Cawood, A.J., Bond, C.E., Howell, J.A., Butler, R.W., and Totake, Y., 2017, LiDAR, UAV or compass-clinometer?: Accuracy, coverage and the effects on structural models: Journal of Structural Geology, v. 98, p. 67–82, https://doi.org/10.1016/j.jsg.2017.04.004.

Chen, A., Darbon, J., and Morel, J.-M., 2014, Landscape evolution models: A review of their fundamental equations: Geomorphology, v. 219, p. 68–86, https://doi.org/10.1016/j.geomorph.2014.04.037.

Christen, M., Kowalski, J., and Bartelt, P., 2010, RAMMS: Numerical simulation of dense snow avalanches in three-dimensional terrain: Cold Regions Science and Technology, v. 63, no. 1–2, p. 1–14, https://doi.org/10.1016/j.coldregions.2010.04.005.

de la Varga, M.I., Schaaf, A., and Wellmann, F., 2019, GemPy 1.0: open-source stochastic geological modeling and inversion: Geoscientific Model Development, v. 12, no. 1, p. 1–32, https://doi.org/10.5194/gmd-12-1-2019.

De Paor, D.G., 2016, Virtual rocks: GSA Today, v. 26, no. 8, p. 4–11, https://doi.org/10.1130/GSATG257A.1.

Dramsch, J.S., 2020, 70 years of machine learning in geoscience in review, *in* Schmelzbach, C., ed., Inversion of Geophysical Data: Elsevier, Advances in Geophysics, v. 61, p. 1–55, https://doi.org/10.1016/bs.agph.2020.08.002.

Edwards, B.I., Bielawski, K.S., Prada, R., and Cheok, A.D., 2019, Haptic virtual reality and immersive learning for enhanced organic chemistry instruction: Virtual Reality (Waltham Cross), v. 23, no. 4, p. 363–373, https://doi.org/10.1007/s10055-018-0345-4.

Fangohr, H., Kluyver, T., and DiPierro, M., 2021, Jupyter in Computational Science: Computing in Science & Engineering, v. 23, no. 2, p. 5–6, https://doi.org/10.1109/MCSE.2021.3059494.

Galin, E., Guérin, E., Peytavie, A., Cordonnier, G., Cani, M.P., Benes, B., and Gain, J., 2019, A review of digital terrain modeling: Computer Graphics Forum, v. 38, p. 553–577, https://doi.org/10.1111/cgf.13657.

Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F.J., and Marín-Jiménez, M.J., 2014, Automatic generation and detection of highly reliable fiducial markers under occlusion: Pattern Recognition, v. 47, no. 6, p. 2280–2292, https://doi.org/10.1016/j.patcog.2014.01.005.

Giles, S., Jackson, C., and Stephen, N., 2020, Barriers to fieldwork in undergraduate geoscience degrees: Nature Reviews Earth & Environment, v. 1, no. 2, p. 77–78, https://doi.org/10.1038/s43017-020-0022-5.

Giorgis, S., Mahlen, N., and Anne, K., 2017, Instructor-led approach to integrating an augmented reality sandbox into a large-enrollment introductory geoscience course for nonmajors produces no gains: Journal of Geoscience Education, v. 65, no. 3, p. 283–291, https://doi.org/10.5408/17-255.1.

Granger, B.E., and Pérez, F., 2021, Jupyter: Thinking and Storytelling With Code and Data: Computing in Science & Engineering, v. 23, no. 2, p. 7–14, https://doi.org/10.1109/MCSE.2021.3059263.

Gunderson, K.L., Holmes, R.C., and Loisel, J., 2020, Recent Digital Technology Trends in Geoscience Teaching and Practice: GSA Today, v. 30, no. 1, p. 39–41, https://doi.org/10.1130/GSATG404GW.1.

Haario, H., Saksman, E., and Tamminen, J., 2001, An adaptive Metropolis algorithm: Bernoulli, v. 7, p. 223–242, https://doi.org/10.2307/3318737.

Hod, Y., and Twersky, D., 2020, Distributed spatial Sensemaking on the augmented reality sandbox: International Journal of Computer-Supported Collaborative Learning, v. 15, p. 115–141, https://doi.org/10.1007/s11412-020-09315-5.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A.A., 2017, Image-to-image translation with conditional adversarial networks: Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, v. 1, p. 5967–5976, https://doi.org/10.1109/CVPR.2017.632.

Jacobs, C.T., Gorman, G.J., Rees, H.E., and Craig, L.E., 2016, Experiences with efficient methodologies for teaching computer programming to geoscientists: Journal of Geoscience Education, v. 64, no. 3, p. 183–198, https://doi.org/10.5408/15-101.1.

Jenkins, H.S., Gant, R., and Hopkins, D., 2014, Shifting sands and turning tides: Using 3D visualization technology to shape the environment for undergraduate students: AGU Fall Meeting, San Francisco, California, 15–19 December, Abstract ED53B-3489.

Kastens, K.A., and Ishikawa, T., 2006, Spatial thinking in the geosciences and cognitive sciences: A cross-disciplinary look at the intersection of the two fields, *in* Manduca, C.A., and Mogk, D.W., eds., Earth and Mind: How Geologists Think and Learn about the Earth: Geological Society of America Special Paper 413, p. 53–76, https://doi.org/10.1130/2006.2413(05).

Kolb, D.A., 1984, Experiential Learning: Experience as the Source of Learning and development: Prentice-Hall, 256 p.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J.B., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., and Willing, C. (Jupyter Development Team), 2016, Jupyter Notebooks—A publishing format for reproducible computational workflows: ELPUB, p. 87–90, https://doi.org/10.3233/978-1-61499-649-1-87.

Kundu, S.N., Muhammad, N., and Sattar, F., 2017, Using the augmented reality sandbox for advanced learning in geoscience education: 2017 IEEE 6th International Conference on Teaching, Assessment, and Learning for Engineering (TALE), p. 13–17.

Lajaunie, C., Courrioux, G., and Manuel, L., 1997, Foliation fields and 3D cartography in geology: Principles of a method based on potential interpolation: Mathematical Geology, v. 29, no. 4, p. 571–584, https://doi.org/10.1007/BF02775087.

Liben, L.S., and Titus, S.J., 2012, The importance of spatial thinking for geoscience education: Insights from the crossroads of geoscience and cognitive science, *in* Kastens, K.A., and Manduca, C.A., eds., Earth and Mind II: A Synthesis of Research on Thinking and Learning in the Geosciences: Geological Society of America Special Paper 486, p. 51–70, https://doi.org/10.1130/2012.2486(10).

Louboutin, M., Lange, M., Luporini, F., Kukreja, N., Witte, P.A., Herrmann, F.J., Velesko, P., and Gorman, G.J., 2018, Devito: An embedded domain-specific language for finite differences and geophysical exploration: https://arXiv.org preprint, arXiv:1808.01995.

Mathiesen, D., Myers, T., Atkinson, I., and Trevathan, J., 2012, Geological visualisation with augmented reality: 15th International Conference on Network-Based Information Systems, p. 172–179.

McNeal, K.S., Ryker, K., Whitmeyer, S., Giorgis, S., Atkins, R., LaDue, N., Clark, C., Soltis, N., and Pingel, T., 2020, A multi-institutional study of inquiry-based lab activities using the Augmented Reality Sandbox: Impacts on undergraduate student learning: Journal of Geography in Higher Education, v. 44, no. 1, p. 85–107, https://doi.org/10.1080/03098265.2019.1694875.

Mergili, M., Fischer, J.-T., Krenn, J., and Pudasaini, S.P., 2017, r. avaflow v1, an advanced open-source computational framework for the propagation and interaction of two-phase mass flows: Geoscientific Model Development, v. 10, no. 2, p. 553–569, https://doi.org/10.5194/gmd-10-553-2017.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., and Teller, E., 1953, Equation of state calculations by fast computing machines: Journal of Chemical Physics, v. 21, no. 6, p. 1087–1092, https://doi.org/10.1063/1.1699114.

Milgram, P., and Kishino, F., 1994, A taxonomy of mixed reality visual displays: IEICE Transactions on Information and Systems, v. 77, no. 12, p. 1321–1329.

Minogue, J., and Jones, M.G., 2006, Haptics in Education: Exploring an Untapped Sensory Modality: Review of Educational Research, v. 76, no. 3, p. 317–348, https://doi.org/10.3102/00346543076003317.

Neal, R.M., 2011, MCMC Using Hamiltonian Dynamics, in Brooks, S., Gelman, A., Jones, G., and Meng, X.-L., eds., Handbook of Markov Chain Monte Carlo: Chapman and Hall/CRC, p. 139–188.

Nesbit, P.R., Boulding, A.D., Hugenholtz, C.H., Durkin, P.R., and Hubbard, S.M., 2020, Visualization and sharing of 3D digital outcrop models to promote open science: GSA Today, v. 30, no. 6, p. 4–10, https://doi.org/10.1130/GSATG425A.1.

Newcombe, N.S., 2012, Two ways to help students with spatial thinking in geoscience, *in* Kastens, K.A., and Manduca, C.A., eds., Earth and Mind II: A Synthesis of Research on Thinking and Learning in the Geosciences: Geological Society of America Special Paper 486, p. 85–86, https://doi.org/10.1130/2012.2486(14).

Oldenburg, D.W., and Li, Y., 2005, Inversion for applied geophysics: A tutorial, *in* Butler, D.K., ed., Near-Surface Geophysics: Society of Exploration Geophysicists, Investigations in Geophysics, v. 13, p. 89–150, https://doi.org/10.1190/1.9781560801719.ch5.

Ormand, C.J., Shipley, T.F., Tikoff, B., Dutrow, B., Goodwin, L.B., Hickson, T., Atit, K., Gagnier, K., and Resnick, I., 2017, The Spatial Thinking Workbook: A research-validated spatial skills curriculum for geology majors: Journal of Geoscience Education, v. 65, no. 4, p. 423–434, https://doi.org/10.5408/16-210.1.

Perkel, J.M., 2018, Why Jupyter is data scientists' computational notebook of choice: Nature, v. 563, no. 7729, p. 145–146, https://doi.org/10.1038/d41586-018-07196-1.

Randles, B.M., Pasquetto, I.V., Golshan, M.S., and Borgman, C.L., 2017, Using the Jupyter notebook as a tool for open science: An empirical study: 2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL), p. 1–2.

Reades, J., 2020, Teaching on Jupyter: Region, v. 7, no. 1, p. 21–34, https://doi.org/10.18335/region.v7i1.282.

Reed, S., Kreylos, O., Hsi, S., Kellogg, L., Schladow, G., Yikilmaz, M.B., Segale, H., Silverman, J., Yalowitz, S., and Sato, E., 2014, Shaping Watersheds Exhibit: An interactive, augmented reality sandbox for advancing Earth science education, Abstract ED34A-01 presented at Fall Meeting, AGU, San Francisco, California, 15–19 December.

Reed, S., Hsi, S., Kreylos, O.,Yikilmaz, M.B., Kellogg, L.H., Schladow, S.G., Segale, H., and Chan, L., 2016, Augmented reality turns a sandbox into a geoscience lesson: Washington, D.C., Eos, v. 97, p. 18–22, https://doi.org/10.1029/2016EO056135.

Roberts, J.L., 2013, Introduction to Geological Maps and Structures: Oxford, England, Pergamon International Library of Science, Technology, Engineering and Social Studies, Elsevier, 340 p.

Rücker, C., Günther, T., and Wagner, F.M., 2017, pyGIMLi: An open-source library for modelling and inversion in geophysics: Computers & Geosciences, v. 109, p. 106–123, https://doi.org/10.1016/j.cageo.2017.07.011.

Shinneman, A.L., Loeffler, S., and Myrbo, A.E., 2020, Self-guided field trips allow flexibility in undergraduate student introductory field experiences: Journal of Geoscience Education, v. 68, no. 4, p. 371–379, https://doi.org/10.1080/10899995.2020.1768006.

Soltis, N.A., McNeal, K.S., Atkins, R.M., and Maudlin, L.C., 2020, A novel approach to measuring student engagement while using an augmented reality sandbox: Journal of Geography in Higher Education, v. 44, no. 4, p. 512–531, https://doi.org/10.1080/03098265.2020.1771547.

von Hagke, C., Wellmann, F., and Urai, J.L., 2019, From Google Earth to 3D Geology Problem 1: Understanding Structures and Generating Geological Maps using Google Earth, *in* Billi, A., and Fagereng, Å., eds., Problems and Solutions in Structural Geology and Tectonics: Elsevier, Developments in Structural Geology and Tectonics, v. 5, p. 181–188.

Wellmann, F., and Caumon, G., 2018, 3-D Structural geological models: Concepts, methods, and uncertainties, in Schmelzbach, C., ed., Advances in Geophysics, v. 59: Elsevier, p. 1–121, https://doi.org/10.1016/bs.agph.2018.09.001.

Whitmeyer, S.J., and Dordevic, M., 2021, Creating virtual geologic mapping exercises in a changing world: Geosphere, v. 17, no. 1, p. 226–243, https://doi.org/10.1130/GES02308.1.

Woods, T., Woods, J., and Woods, M., 2015, Using the Kreylos augmented reality sandbox to teach topographic maps and surficial processes in an introductory geology lab at East Carolina University: Geological Society of America Abstracts with Programs, v. 47, no. 7, p. 111.

Woods, T.L., Reed, S., Hsi, S., Woods, J.A., and Woods, M.R., 2016, Pilot study using the augmented reality sandbox to teach topographic maps and surficial processes in introductory geology labs: Journal of Geoscience Education, v. 64, no. 3, p. 199–214, https://doi.org/10.5408/15-135.1.

Zhao, J., LaFemina, P., Carr, J., Sajjadi, P., Wallgrün, J.O., and Klippel, A., (2020, March). Learning in the field: Comparison of desktop, immersive virtual reality, and actual field trips for place-based STEM education: 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), p. 893–902.