Computers and Geosciences 109 (2017) 25–31

Contents lists available at ScienceDirect

Computers and Geosciences

journal homepage: www.elsevier.com/locate/cageo

Research paper

TouchTerrain: A simple web-tool for creating 3D-printable topographic models

Franciszek J. Hasiuk [a,*], Chris Harding [b], Alex Raymond Renner [c], Eliot Winer [c]

[a] Department of Geological and Atmospheric Sciences, 253 Science I, Iowa State University, Ames, IA, USA
[b] Department of Geological and Atmospheric Sciences, 253 Science I, Human-Computer Interaction Program, 1620 Howe Hall, Iowa State University, Ames, IA, USA
[c] Department of Mechanical Engineering, Human-Computer Interaction Program, 1620 Howe Hall, Iowa State University, Ames, IA, USA

ARTICLE INFO

Keywords:
3D printing
Topography
Terrain
Google Earth Engine
Python
Direct Digital Manufacturing

ABSTRACT

An open-source web-application, TouchTerrain, was developed to simplify the production of 3D-printable terrain models. Direct Digital Manufacturing (DDM) using 3D Printers can change how geoscientists, students, and stakeholders interact with 3D data, with the potential to improve geoscience communication and environmental literacy.

No other manufacturing technology can convert digital data into tangible objects quickly at relatively low cost; however, the expertise necessary to produce a 3D-printed terrain model can be a substantial burden: knowledge of geographical information systems, computer aided design (CAD) software, and 3D printers may all be required. Furthermore, printing models larger than the build volume of a 3D printer can pose further technical hurdles.

The TouchTerrain web-application simplifies DDM for elevation data by generating digital 3D models customized for a specific 3D printer's capabilities. The only required user input is the selection of a region-of-interest using the provided web-application with a Google Maps-style interface. Publically available digital elevation data is processed via the Google Earth Engine API. To allow the manufacture of 3D terrain models larger than a 3D printer's build volume the selected area can be split into multiple tiles without third-party software. This application significantly reduces the time and effort required for a non-expert like an educator to obtain 3D terrain models for use in class. The web application is deployed at http://touchterrain.geol.iastate.edu, while source code and installation instructions for a server and a stand-alone version are available at Github: https://github.com/ChHarding/TouchTerrain_for_CAGEO.

1. Introduction

Terrain has a profound influence on many Earth processes and human activities, such that a thorough understanding of it is vital to many geoscience and engineering disciplines. Despite this importance, the nature and scale of terrain often places it outside simple comprehension, which leads to difficulty in the classroom when students are asked to make qualitative and quantitative measurements using traditional topographic maps (Tversky, 2003; Taylor et al., 2004; Ishikawa and Kastens, 2005; Rapp et al., 2007). 3D-printed models can overcome this problem by putting data directly into the hands of students, educators, citizens, and stakeholders (Hasiuk, 2014; Hasiuk and Harding, 2016). The goal of the TouchTerrain project is to overcome the most technically challenging barriers to more widespread adoption in the classroom by providing a web application for easily generating 3D-printable terrain model files of any area on Earth.

Elevation data is widely available in digital elevation models (DEMs) derived from remote sensing techniques with meter-scale accuracy or better (Bellian et al., 2005). DEMs can be visualized in a variety of ways (Buckley et al., 2004; Mach and Petschek, 2007; Mitasova et al., 2012). Traditional 2D visualizations include contour lines, color sequences/-ramps, and hillshading. To trained geoscientists, the spacing between the lines also suggest the slope, i.e., smaller gaps indicate a steeper hill; however, students often find it difficult to make the leap from reading a contour map to visualizing a terrain's 3D shape.

Visualizing terrain data in 3D leads to additional visualization techniques (Johnson et al., 2006; Mach and Petschek, 2007). 3D viewers, such as ESRI's ArcScene, and digital globes, such as Google Earth, combine visualization of terrain properties in 2D space with interactive viewpoint navigation to enable users to explore terrain data in ways that

* Corresponding author.
E-mail addresses: franek@iastate.edu (F.J. Hasiuk), charding@iastate.edu (C. Harding), arenner@iastate.edu (A.R. Renner), ewiner@iastate.edu (E. Winer).

http://dx.doi.org/10.1016/j.cageo.2017.07.005
Received 3 October 2016; Received in revised form 23 April 2017; Accepted 16 July 2017
Available online 25 July 2017

they cannot in the real world. However, it is still not clear for what use-cases 3D maps are best suited (Schobesberger and Patterson, 2007; Popelka and Brychtova, 2013). Augmented reality sandboxes allow students to move sand and witness the resulting effects on topographic maps digitally superimposed on the sand and updated in real-time (e.g., Woods et al., 2016).

The intuitive and material nature of 3D-printed terrain models give them advantages over 2D maps and 3D computer visualizations. Actions such as zooming and rotating are accomplished via hand positioning of the model, and surface details are easily discerned. The models can be directly annotated with pens and can help visually-impaired users to comprehend terrain (Wild et al., 2013). Although the use of 3D-printed terrain models as instructional material is still in its infancy, early research has shown that such models have value, either on their own, or in concert with 2D and 3D terrain visualization methods (Rule, 2011; Williams et al., 2013; Horowitz and Schultz, 2014; Hasiuk and Harding, 2016).

As the equipment, software, and material costs for printing 3D models have come down, the greatest cost lies in generating a 3D model file of the chosen area that reliably prints well on a specific type of 3D printer. The expertise and time required to do this, along with the specialized software which may be required, is a hurdle to widespread use of 3D terrain models. The TouchTerrain project aims to remove this barrier, empowering educators to use 3D-printed terrain models from any area on the Earth as a basis for novel teaching methods in the classroom and the field.

2. Methods

2.1. 3D printers

A 3D printer is designed to take information from any digital 3D model and make a tangible 3D model (Pham and Gault, 1998). "Fused Filament Fabrication" (FFF) is one 3D printing method that uses plastic filament with a small circular cross-section (often 1.75 mm or 3 mm in diameter). The filament is heated to a semi-molten state and then extruded through a nozzle with an orifice smaller than the original filament diameter (e.g., 0.4 mm). After the filament is added to a model, it quickly cools until fully solidified, gaining strength and rigidity in the process. Motors position the nozzle in a plane that is orthogonal to the direction of extrusion. When the nozzle changes position, a specified amount of material is extruded. The result of the manufacturing method is a "3D printed segment" (Renner et al., 2015), that has a specific thickness, length, and shape. These dimensions depend on the orifice diameter, changes in nozzle position (both velocity and direction), and the temperature of the material as it exits the nozzle. After the first printed segment is laid down, the build plate lowers (typically 0.025–0.4 mm) and the algorithm is repeated, fusing a new layer of printed segments to the previously printed layer.

Final model quality is affected by a 3D printer's hardware specifications, software settings used for printing, digital model quality, and build-material. Digital model quality has a particularly significant impact on the printing process because the digital model must be processed by a "slicing" algorithm to create the series of 2D horizontal cross-sections (Fig. 1A) that are the fundamental instructions to the printer. These cross-sections are created at discrete intervals that correspond to the distance the build plate moves away from the nozzle. A cross-section consists of a series of positions along a particular XY-plane through the digital model that will be sent as commands to the printer. If there is a problem in creating even one cross-section, the slicing software may fail to generate the required instructions for the printer to begin.

2.2. Initial attempts at "hand-crafting" 3D printable terrain

When a pre-made 3D model is not available, or needs to be adjusted, users have had to rely on "hand-crafting" techniques including how to find and download the appropriate elevation data as well as how to process it through different software packages to create the 3D digital model (Fig. 2). Although, hand-crafting provides a high degree of flexibility, it presents substantial barriers in terms of operator expertise, software-hardware requirements, and cost. In order to develop TouchTerrain, we experimented with a number of hand-crafting techniques, as described below.

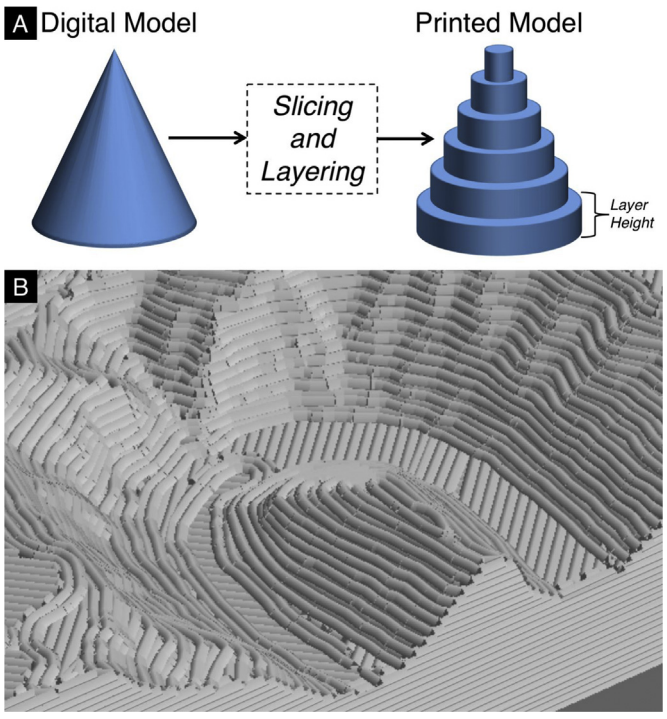

Fig. 1. Basic workflow for 3D printing a 3D digital model. A slicing algorithm turns the digital 3D model into a series of sequential 2D layers, each with a uniform height (h). This layer thickness is one of the fundamental specifications of a 3D printer. B. Preview of a 3D print showing stacked horizontal slices. Every element approximates a segment of extruded plastic filament.

2.2.1. Vertical exaggeration

Because of its inherent layering process, 3D printed terrain has the visual quality of stacked 3D contours (Fig. 1B). Although this visual quality differs from the data's on-screen 3D rendering, we consider this a benefit because it highlights the most salient features of the terrain's morphology and it is analogous to topographic map contours. With this limitation in mind, it proved important to consider exaggerating the vertical scale (z-scale) for effective 3D printing of gently sloping terrain (e.g., Iowa). We found that a vertical exaggeration of 2-5x usually ensured a sufficiently detailed model.

2.2.2. Resolution

Printing a 10 cm by 10 cm (lateral size) model on a printer where the smallest movement the extruder can realistically perform is around 0.25 mm, results in an underlying lateral grid of about 400 by 400 steps ($400 \times 0.25\ \text{mm} = 10\ \text{cm}$). We found that trying to force the 3D printer to move on a finer lateral grid (e.g., 0.1 mm) yields barely perceptible improvements in print detail at the cost of longer slicing time, longer print time, and an increased chance of failure due to the mechanics of the printer trying to print very fine features. Overly complex 3D model files can be avoided by intelligently down-sampling a raster before creating a 3D mesh. For instance, down-sampling a 800×800 pixel DEM raster to 400×400 pixels prior to meshing reduces the file size by approximately 75%.

2.2.3. Maximum object size

Each 3D printer has certain inherent limitations, including the

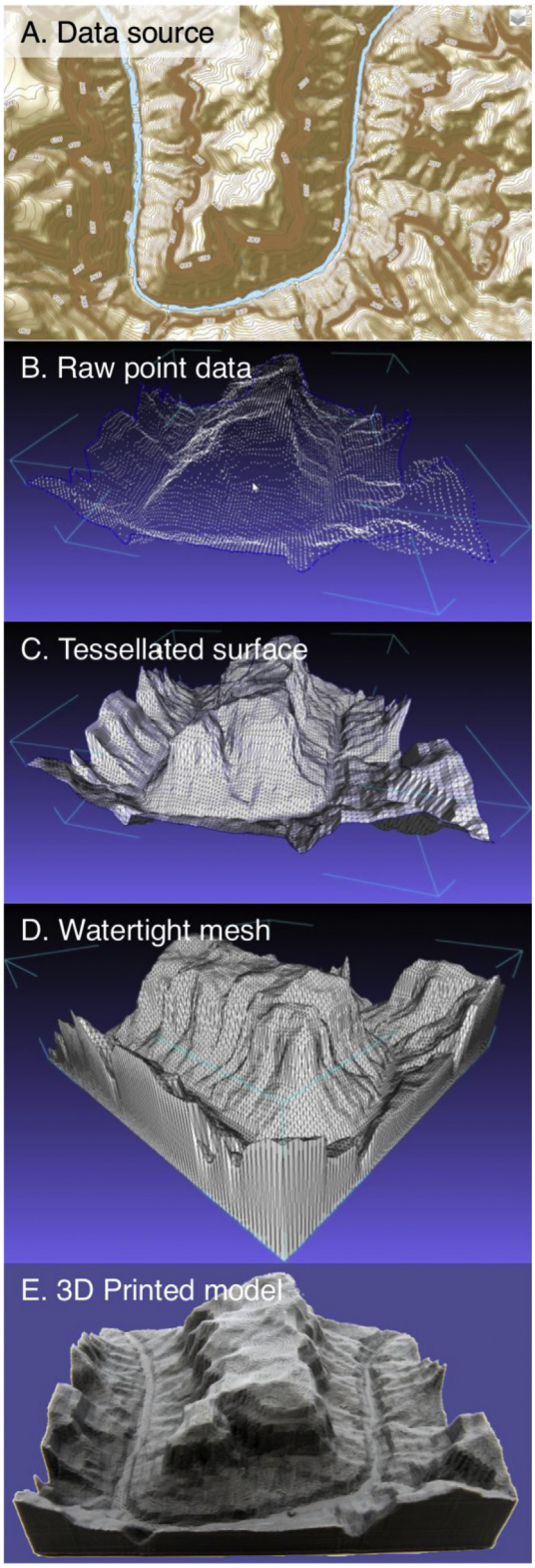

Fig. 2. Steps used to transform elevation data in raster form to a 3D printable model with Google Earth Engine (image from USGS National Map Viewer). A. Elevation data is downloaded from internet data repository. B. Elevation data is converted to a 3D point cloud. C. A mesh surface is interpolated from the point data. D. Additional triangles are added to form the sides and base. E. Digital model is 3D-printed.

maximum size object it can create. For instance, the largest possible build volume for the MakerBot Replicator 2X is 246 mm wide, 152 mm deep and 155 mm high. Any digital 3D models created for this 3D printer must fit into this envelope. This limitation can be overcome by tiling digital models and then assembling them after 3D printing into a larger model. For example, an 800 × 800 pixel DEM raster could be divided into four 400 × 400 pixel tiles. Each tile could be printed as a 10 × 10 cm model and all four tiles could be assembled to create a 40 × 40 cm 3D map. However, tiling a 3D model manually by either splitting the raster prior to meshing or by splitting the larger mesh into several smaller meshes proved to be complex and laborious.

In summary, the complexity of this hand-crafting process and its requirement for expertise in multiple types of software tends to create a rather steep learning curve, which makes it unsuitable for non-experts. With the lessons learned while creating 3D digital terrain models and 3D printing them, we designed and deployed an automated process that would hide these complexities from the user and deliver 3D models to the user ready for printing (Table 1, Fig. 3).

3. TouchTerrain architecture and Google Earth Engine

TouchTerrain has a client-server architecture (Fig. 3). The server is written in Python and, in conjunction with Google Earth Engine (GEE), performs heavy computations that would not be feasible to run on the frontend, in the client's browser. The client is a webpage written with JavaScript which communicates with both the server and GEE.

We use Google Earth Engine (GEE) to access and process a variety of DEM rasters. Unrelated to Google Earth, Google Earth Engine is a development environment for analyzing raster data via a command-line tool, a JavaScript API or a Python API. GEE's data catalog (explorer.earthengine.google.com/#search/tag:elevation) offers several seamless geospatial data sets, including several types of elevation data (Table 2). The GEE API offers a wide range of raster and vector processing functions (cf. developers.google.com/earth-engine/). Processing is performed on Google's cloud servers and results are returned as GeoTIFF files. The easiest way to explore GEE data and functionality is via the GEE Code Editor, a web-based IDE for the Earth Engine JavaScript API (developers.google.com/earth-engine/playground).

On the client-side (browser frontend), we use the GEE JavaScript API to overlay a semi-transparent hillshade version of the selected DEM in a Google Maps window (Fig. 3B). The region of interest (ROI) selection frame is also provided by the Google Maps API. Once the ROI, the DEM source, and the 3D printer parameters are defined, the Python server builds the 3D printable model(s).

On the server-side (backend), we use the GEE Python API to request a DEM raster covered by the ROI from the GEE data catalog (Fig. 3D). All elevation rasters provided by GEE initially store their elevation in meters and their x-y coordinates in decimal degrees (latitude/longitude, WGS84 datum). To minimize distortions and to set the horizontal units to the unit of the elevation (meters), the server instructs GEE to reproject the raster into the closest UTM zone. GEE is also perform a bi-linear re-sampling of the raster to match the lateral resolution of the 3D printer. For example, with a printer resolution of 0.25 mm, the user might select a square ROI on the map for printing as a single tile 10 × 10 cm model. After UTM projection, the raster might contain 1200 × 1200 cells (depending on the

Table 1
Comparison of workflow for creating 3D-printable terrain models via expert abilities ("Hand-crafted") and TouchTerrain.

Step	"Hand-Crafted" Workflow	TouchTerrain Workflow
Select Region of Interest	User via data-owner (e.g., USGS)	User via TouchTerrain GUI
Down-sample data to 3D Printer Resolution	User via GIS	TouchTerrain Server
Interpolate Surface ("Meshing")	User via GIS/Mesh-software	TouchTerrain Server
Add Sides and Base	User via CAD (e.g. MakeMixer)	TouchTerrain Server

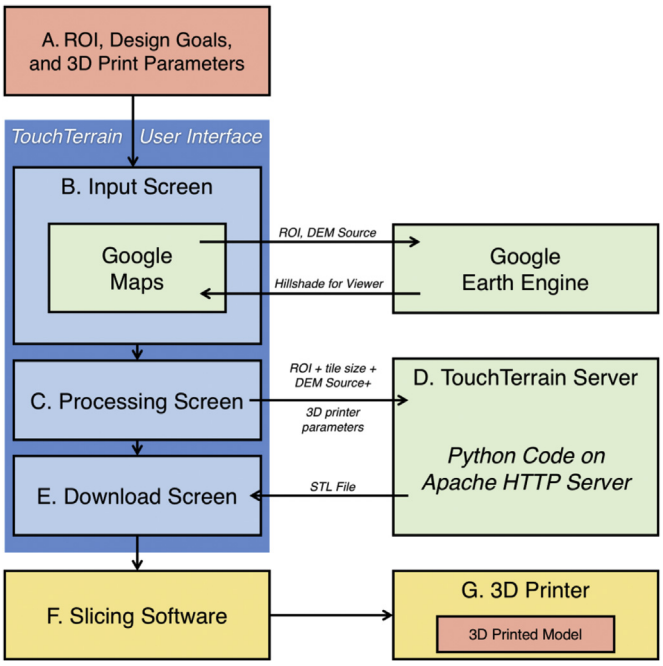

Fig. 3. Workflow for TouchTerrain web application. A) User selects the Region of Interest (ROI), identifies design goals (e.g., tile configuration, vertical exaggeration), and 3D printer specifications in the TouchTerrain User Interface (B). The UI's Google Maps window automatically updates the hill-shaded terrain view using Google Earth Engine. C) ROI, model and printer parameters are fed to the Touch Terrain server (D), which creates an STL file per slice that can be downloaded (E) and 3D printed (F,G).

Table 2
Publically available digital elevation data used in TouchTerrain.

Name	Resolution	Areal Extent	Bathymetry Included	Reference
USGS NED	1/9 arc-sec (~10 m)	USA	No	USGS Nationalmap.gov
SRTM GL1	1/3 arc-sec (~30 m)	Global	No	NASA, 2013
GMTED2010	7.5 arc-sec (~225 m)	Global	No	Danielson and Gesch, 2011
ETOPO1	60 arc-sec (~1 km)	Global	Yes	Amante and Eakins, 2009

size of the ROI, the latitude and the resolution of the DEM source). In this case, the raster would be re-sampled to 400 × 400 cells, to avoid creating an overly complex 3D model. Alternatively, to 3D-print this ROI as four 10 × 10 cm tiles, the ROI would be re-sampled to 800 × 800 cells by GEE and then split into four 400 × 400 pixel tiles by the server.

To use GEE as a developer it is necessary to authenticate with a GEE developer account, which can be used free of charge for research or educational purposes, albeit with some restrictions. The authentication requires only the installation of a file on the server as well as a few Python libraries (cf. developers.google.com/earth-engine/python_install). Query request API calls made to GEE, such as downloading the projected and re-sampled GeoTIFF, have a quota limit of a few queries per second. Only one query is required to create all 3D models for a given ROI. End-users of the web application do not require a GEE account.

Once the UTM-projected and re-sampled raster of the ROI has been received by the server, the server converts it into a 2D array and, if necessary, splits it into tiles. The elevation value of each cell is considered to be at its center. To make a mesh, a "quad" (square) data structure is created containing the 3D coordinates of each cell's corners. The x/y coordinates of these corners follow from the cell size (i.e., they are always half-way between the centers of adjacent cells). The elevation (z-coordinate) of each quad corner is bi-linearly interpolated from its four surrounding cell elevations via simple averaging. For border and corner cells, these surrounding cells are either part of an adjacent tile or initially missing. To solve this, a tile's array is temporarily padded with a one-cell-wide fringe with elevations set to the edge values of the neighbouring tiles. Fig. 4A shows a 4-by-2 cell raster of an ROI (red outlined cells), that has been padded (gray cells) and divided into two 2-by-2-cell tiles (blue and green cells). The elevations of the four corners of the dash-outlined quad is the average of the elevation of its adjacent four cells, some of which are part of the adjacent tile (green) or the padding (gray).

In addition to getting the x/y/z coordinate of each corner, each quad also records which of its sides (if any) has no adjacent cell(s) representing terrain. These edge quads will be used in subsequent steps to define where the sides of the model will be built. In the future, this principle could also be used to deal with irregular ROIs selected by digitizing a polygon, rather than the simple rectangular ROI currently used.

The quads are bisected into two triangles (Fig. 4B) that form a watertight 3D triangular mesh surface representing the terrain (Fig. 2D). Triangle vertices are listed in counterclockwise sequence to encode its normal. For the triangles on the terrain mesh ("top-triangles"), x/y/z coordinates are given by the corners of the quads. Triangles for the flat base of the model ("bottom-triangles") are created using the x/y coordinates of the top-triangles and the minimum elevation of all cells in the ROI as the z-coordinate. Using the minimum of all tiles as bottom elevation ensures that the upper surfaces of all printed tiles fit together properly. Triangles for the sides ("walls") are created by going over all the previously flagged edge quads and connecting them vertically to the bottom elevation to create walls. Although it would be possible to lower the number of triangles used in the bottom and the sides through better tessellation, this carries the danger of creating issues like non-manifold edges or vertices (i.e., edges or vertices that do not connect to the rest of the mesh), which may lead to problems later during slicing and 3D printing.

Finally, the model for each tile is scaled from real world meter-based (UTM) coordinates to the requested size with units in millimeters and centered at 0/0/0 (i.e., the center of the 3D printer's build-plate). With this, the user does not need to manually scale the tiles before printing, ensuring that all printed tiles will fit together to create a large 3D terrain map.

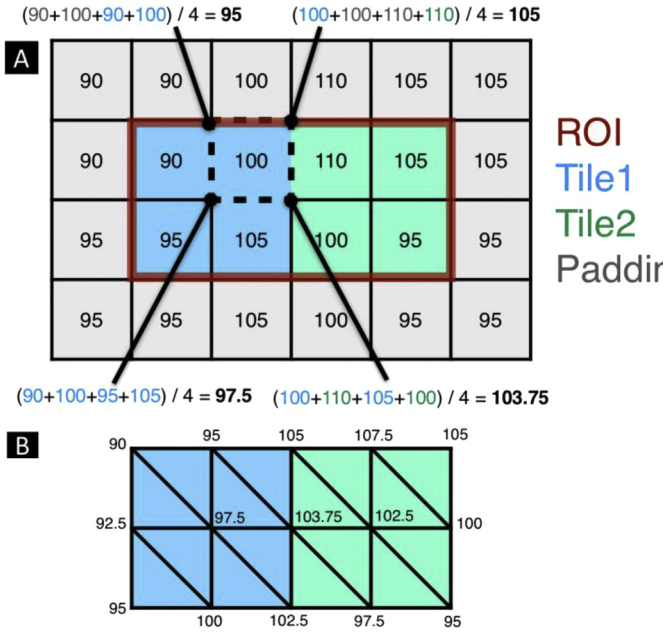

Fig. 4. A. Interpolating the elevation of the corners of a quad (cell) within a tile from the cell values of the ROI 2D array. B. Interpolated and tessellated 2D ROI as it appears in TouchTerrain output STL file. The bottom elevation for both tiles would be lower than 90.

3.1. Digital 3D model files and 3D printing

Each triangle's mesh is written to its own file, and all tiles (plus an info file) are placed into a zip folder for download. The standard 3D print file format is STL (an abbreviation of "stereolithography," one of the first 3D printing methods), which only describes the mesh geometry as a cloud of triangles, each with a sequence of vertices and a normal. STL does not support vertex indexing so a vertex used by multiple triangles is listed multiple times in the file. OBJ, another file format popular for 3D modelling applications, uses vertex indexing as well as having the ability to include color information that could be used for full-color 3D printing on compatible 3D printers.

To print the tiles, each tile's file is loaded into a 3D print application, typically supplied by the printer's manufacturer. For our MakerBot Replicator 2X, we used MakerBot Desktop. Once loaded, no scaling is needed because the tile already has the desired size and vertical exaggeration. A common 3D printing phenomenon encountered on models with 90° corners is a bending-up effect of the lower layers at the corners due to the plastic detaching from the build platform. To counteract this effect, we placed 20-mm-diameter discs under each corner, which are clipped off after the tile is printed (Fig. 5A). To assemble the tiles into a larger model, the sides can be slightly smoothed with sandpaper for better fit and glued together with acetone or cyanoacrylate (Fig. 5B). As an alternative to printing tiles oneself, the model files can be sent to an online 3D print shop.

3.2. Our web server infrastructure

Currently, we use the open source Apache server architecture to connect the browser front-end with the Python server, i.e., to transmit the user-chosen ROI and print parameters to the server and serve the printable model to the user after processing. ("nginx" is a lighter-weight open-source alternative.) We use webapp2 (webapp-improved.appspot.com/) as a Python web framework together with Jinja2 (jinja.pocoo.org/docs/dev/) for templating. The current infrastructure is functional, but rather simplistic and misses several convenience and performance scaling features we hope to add in the future (e.g., an email notification once the processing is done). Iowa State runs a Linux server with 6 CPU's and 12 Gb of RAM. During the 2 weeks following the 1.0 release date (March 14, 2017) roughly 2000 users accessed the website to download 3D terrain models.

3.3. Usage example

The Iowa State TouchTerain server is accessible at: touchterrain.geol.iastate.edu. In the Google Maps window, the user selects an ROI with a red area selection box (Fig. 6A). DEM data source is selected from a drop-down menu (Fig. 6B, Table 2). The background map type can be set to street-map or satellite. A semi-transparent hillshade layer (Fig. 6C) of the currently selected DEM source can be overlaid, to show data availability because not all areas are available for each source. Varying the transparency helps to identify interesting terrain features. If the user panned away from the red-selection box, it can be re-centered on the current view (Fig. 6D).

The following options are available in TouchTerrain (Fig. 6E) to match the specifications of the user's 3D printer:

- *Tile width*: the width of each tile after printing, its height follows from the aspect ratio of the ROI and the tile configuration. The height is automatically calculated from the aspect ratio of the ROI. Both must be smaller than the build plate the model will be printed on.
- *3D print resolution*: Distance the 3D printer will move to print one cell of the resampled elevation raster and its approximate real-world distance in meters. Small distances (i.e., high resolutions) will create larger 3D model files and require longer processing. The thickness of the extruded (hot) filament is a good value for this parameter.
- *Tile configuration*: Number of tiles in x and y the ROI is divided into. The red area selection box is automatically updated to show the tile configuration.
- *Base thickness*: Extra thickness to be added to the base of a model to provide structural rigidity and strength beneath the lowest elevation features (e.g., a river).
- *Vertical exaggeration*: Vertical scale can be exaggerated to accentuate the terrain in areas with subtle topography (e.g., Iowa). In most cases a factor of 2–3 is sufficient.
- *File format*: STL (ASCII or binary) and OBJ (ASCII only) are common file formats accepted by 3D printer software. Binary STL files are recommended as they have the smallest file size.

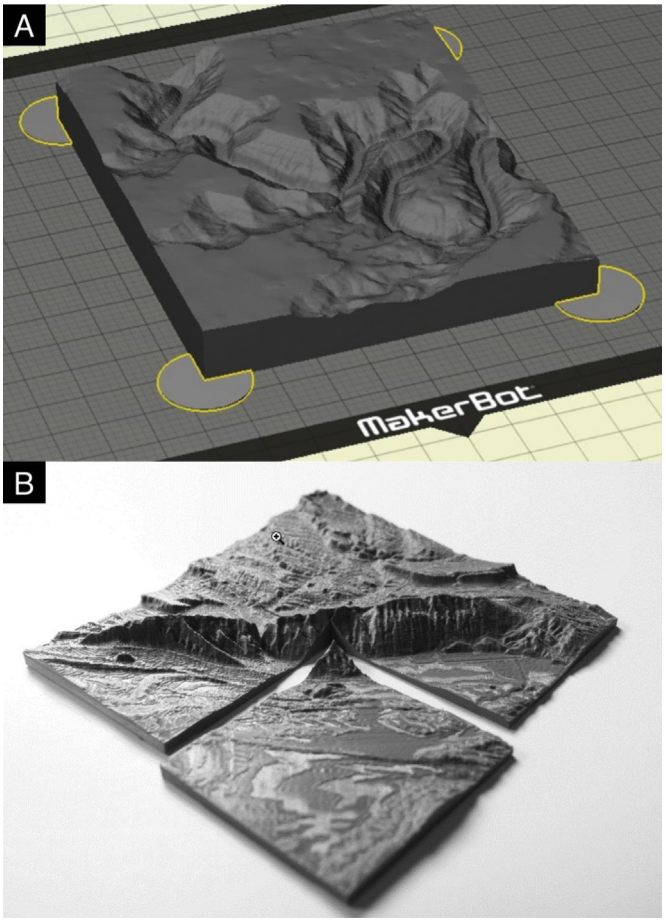

Fig. 5. A. 3D terrain model output from TouchTerrain v3 prepared for 3D printing by adding "helper discs" at the corners of the model to reduce shrinkage during printing (Makerbot Desktop 3.8.1 software). B. Four-tile model of Sheep Mountain, Wyoming (ROI in Fig. 6) corresponding to a 1:81,000 map scale with 2x vertical exaggeration. Each tile is 10 × 10 cm. A (lateral) 3D printer resolution of 0.25 was used, for which the USGS NED 10 m DEM was down-sampled to 20 m per cell. Printing each tile on a Replicator 2X took around 2 h and cost around $2 in material.

Clicking the export button (Fig. 6F) begins the process of generating the 3D printable model. When completed, the user is given a link to download a zipped folder containing the tiles in the requested file format. A summary text file is provided containing metadata about the tiles, such as date, resolution, real-world horizontal scale, vertical exaggeration. We initially experimented with embossing this information into the bottom of the tile, but it proved detrimental to the overall build quality for the first few layers to not be continuous. Instead, we typically write this information with a marker on the bottom of the tile or print it on paper and glue to the bottom of the tile.

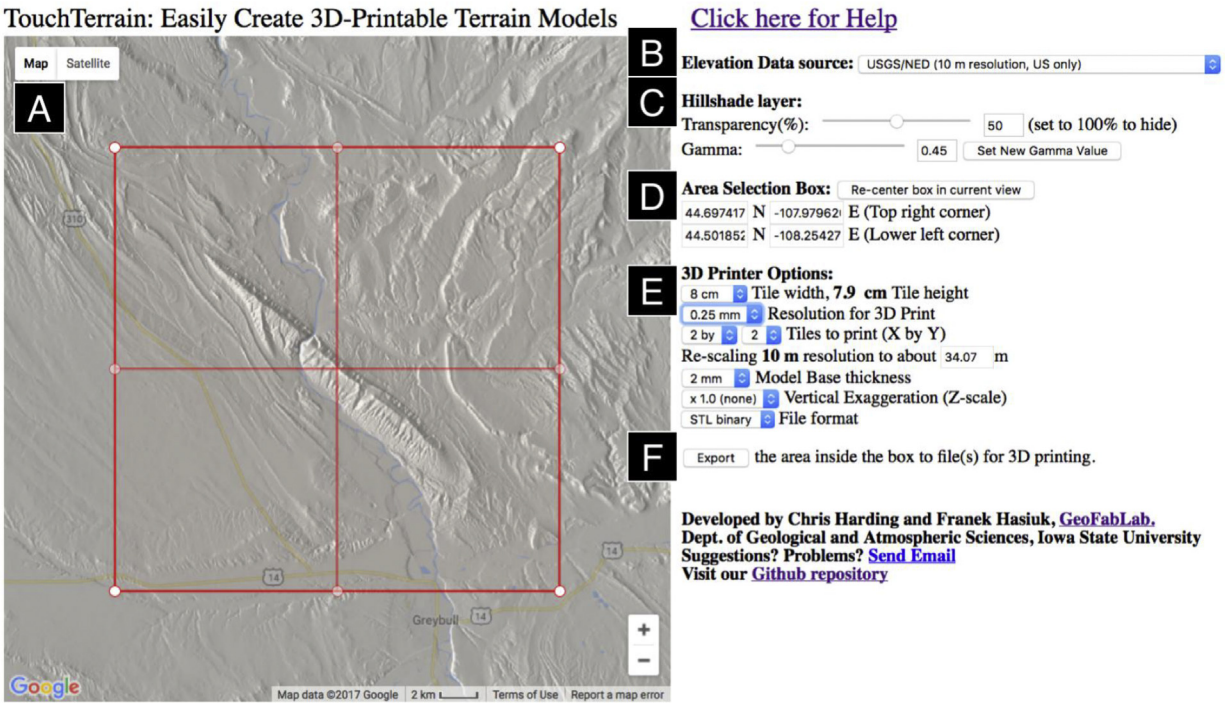

Fig. 6. Main user-interface for TouchTerrain. A: Google Maps-interface for selecting ROI, B: drop-down menu to select the elevation dataset. C: slider to set the transparency of the hill-shaded terrain, D: coordinates of the region of interest, E: options related to tiling and 3D printing; F: Button that submits the ROI and model parameters to the TouchTerrain server for processing.

3.4. Applications

TouchTerrain has a variety of applications to the student and educator which could be easily transferred to use by researchers, farmers, or outdoor enthusiasts, in business and government. TouchTerrain at its core provides greater familiarity with a location's terrain. 3D printed terrain could be used when presenting potential plans for local infrastructure to stakeholders and the general public. A classroom-bound teacher could include TouchTerrain models of the Grand Canyon in an exercise on geology, geography, history, or even poetry. In a field mapping course, an instructor could give students TouchTerrain models to describe key features of the field area, to communicate safety risks, or to serve as a base map for recording observations or drawing fault lines, rock unit boundary polygons, etc. Finally, wherever possible, the instructor should include the student in the process of 3D printing the TouchTerrain model. This will give them a greater sense of ownership and educate them on the technical aspects of 3D printing.

3.5. Future developments

While developing TouchTerrain and communicating with users, several ideas have been generated that have yet to make their way into the application.

3.5.1. GUI improvement

ROI selection could permit the digitizing of a bounding polygon instead of a quadrilateral.

3.5.2. User community

Because of the "maker culture" that surrounds 3D printing, it would be useful to supplement the web-application with a user community, where users would be able to see the ROIs that others have made models from. This could be easily accomplished by storing the coordinates of user-requested ROIs. If storage space was not cost-prohibitive, the digital 3D models themselves could be stored. In addition, users could share tips for how to make their models print the best on different 3D printers on a user forum or listserv.

3.5.3. Additional terrain data

Terrain data is available for numerous extraterrestrial bodies (e.g., Mars, the Moon). This data, when available in GEE, could easily be made 3D printable through TouchTerrain. One of the most significant ways this technology could be developed would be to add the ability to 3D print paleogeographies that show how Earth's terrain looked in the past when continents and oceans had different locations and morphologies.

3.5.4. Full color and image overlay

While not a standard feature on most 3D printers, the ability to print in multiple colors is becoming more common. TouchTerrain could be modified to allow the overlay of map data or image data on a terrain model so links between terrain and other properties (e.g. geologic units, faults, population density) could be studied.

3.5.5. Upload link to 3D printing services

While some users may want to maintain their own 3D printer, there are numerous online services (e.g., 3DHubs.com, RedEye.com, Shapeways.com) that allow 3D printable models to be uploaded and printed for a fee. Establishing a direct link from TouchTerrain to such services would further democratize the access to terrain models by freeing users from the need to own and maintain a 3D printer.

4. Conclusions

We explored creating of digital 3D terrain models suitable for 3D printing on personal 3D printers. After initially hand-crafting the digital models via a complex, manual workflow utilizing several software tools, we created a web-application that hides much of the complexity from the end-user. As result, the user receives 3D-printable, multi-tiled model files that cover the requested area.

The TouchTerrain server can be used at http://touchterrain.geol.iastate.edu. Open-source code can be downloaded from the Github online code repository (https://github.com/ChHarding/TouchTerrain_for_

CAGEO) and either deployed as a server or run as a stand-alone version with parameters (ROI coordinates, DEM source, etc.) supplied via text (JSON) file.

TouchTerrain offers a simple method for producing a 3D printable terrain models, with functionality that is not easily provided elsewhere (e.g., tiling, multiple DEM sources, 3D printer resolution). We believe that 3D terrain models can play a valuable teaching role and that providing easy access to tangible terrain models 3D will play a vital role in increasing spatial literacy among students, researchers, and the broader public.

References

Amante, C., Eakins, B.W., 2009. ETOPO1 1 Arc-minute Global Relief Model: Procedures, Data Sources and Analysis. NOAA Technical Memorandum NESDIS NGDC-24. National Geophysical Data Center, NOAA. http://dx.doi.org/10.7289/V5C8276M.

Bellian, J.A., Kerans, C., Jennette, D.C., 2005. Digital outcrop models: applications of terrestrial scanning lidar technology in stratigraphic modeling. J. Sediment. Res. 75, 166–176. http://dx.doi.org/10.2110/jsr.2005.013.

Buckley, A., Hurni, L., Kriz, K., Patterson, T., Olsenholler, J., 2004. Cartography and visualization in mountain geomorphology. In: Bishop, M.P., Shroder, J.F. (Eds.), Geographic Information Science and Mountain Geomorphology. Springer-Praxis, pp. 253–287.

Danielson, J.J., Gesch, D.B., 2011. Global Multi-resolution Terrain Elevation Data 2010 (GMTED2010): U.S. Geological Survey Open-file Report 2011–1073, p. 26.

Google Earth Engine Team, 2015. Google Earth Engine: a Planetary-scale Geospatial Analysis Platform. earthengine.google.com.

Hasiuk, F.J., 2014. Making things geological: 3-D printing in the geosciences. GSA Today 24, 28–29. http://dx.doi.org/10.1130/GSATG211GW.1.

Hasiuk, F.J., Harding, C., 2016. Touchable topography: 3D printing elevation data and structural models to overcome the issue of scale. Geol. Today 32, 16–20.

Horowitz, S.S., Schultz, P.H., 2014. Printing space: using 3D printing of digital terrain models in geosciences education and research. J. Geosci. Educ. 62, 138–145.

Ishikawa, T., Kastens, K.A., 2005. Why some students have trouble with maps and other spatial representations. J. Geosci. Educ. 53, 184–197.

Johnson, A., Leigh, J., Morin, P., Van Keken, P., 2006. GeoWall: stereoscopic visualization for geoscience research and education. Comput. Graph. Appl. IEEE 26, 10–14.

Mach, R., Petschek, P., 2007. Visualization of Digital Terrain and Landscape Data: a Manual. Springer Science and Business Media.

Mitasova, H., Harmon, R.S., Weaver, K.J., Lyons, N.J., Overton, M.F., 2012. Scientific visualization of landscapes and landforms. Geomorphology 137, 122–137.

NASA, 2013. NASA shuttle radar topography mission global 1 arc second. NASA LP DAAC. http://dx.doi.org/10.5067/MEaSUREs/SRTM/SRTMGL1.003.

Pham, D.T., Gault, R.S., 1998. A comparison of rapid prototyping methodologies. Int. J. Mach. Tools Manuf. 38, 1237–1287.

Popelka, S., Brychtova, A., 2013. Eye-tracking study on different perception of 2D and 3D terrain visualisation. Cartogr. J. 50, 240–246.

Rapp, D.N., Culpepper, S.A., Kirkby, K., Morin, P., 2007. Fostering students' comprehension of topographic maps. J. Geosci. Educ. 55, 5–16.

Renner, A., Holub, J., Sridhar, S., Evans, G., Winer, E., 2015. A virtual reality application for additive manufacturing process training. In: ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (pp. V01AT02A033). American Society of Mechanical Engineers.

Rule, A.C., 2011. Tactile earth and space science materials for students with visual impairments: contours, craters, asteroids, and features of Mars. J. Geosci. Educ. 59, 205–218.

Schobesberger, D., Patterson, T., 2007. Evaluating the effectiveness of 2D vs. 3D trailhead maps; A map user study conducted at Zion National Park, United States. Mt. Mapp. Vis. 6, 201–205.

Taylor, H.A., Renshaw, C.E., Choi, E.J., 2004. The effect of multiple formats on understanding complex visual displays. J. Geosci. Educ. 52, 115–121.

Tversky, B., 2003. What maps reveal about spatial thinking. Dev. Sci. 3, 281–282. http://dx.doi.org/10.1080/00223980309600638.

Wild, T.A., Hilson, M.P., Farrand, K.M., 2013. Conceptual understanding of geological concepts by students with visual impairments. J. Geosci. Educ. 61, 222–230.

Williams, W.J., Williams, J., Williams, S.G.W., 2013. Universal design in geoscience education: promoting student success with accessibility using local 3D printing for learner-centered instruction. In: 2013 GSA Annual Meeting in Denver.

Woods, T.L., Reed, S., Hsi, S., Woods, J.A., Woods, M.R., 2016. Pilot study using the augmented reality sandbox to teach topographic maps and surficial processes in introductory geology labs. J. Geosci. Educ. 64, 199–214. http://dx.doi.org/10.5408/15-135.1.