Immersive Tangible Geospatial Modeling

Payam Tabrizian
ptabriz@ncsu.edu

Anna Petrasova
akratoc@ncsu.edu

Brendan Harmon
baharmon@ncsu.edu

Vaclav Petras
vpetras@ncsu.edu

Helena Mitasova
hmitaso@ncsu.edu

Ross Meentemeyer
rkmeente@ncsu.edu

Center for Geospatial Analytics
North Carolina State University
Raleigh, North Carolina 27695

ABSTRACT

Tangible Landscape is a tangible interface for geographic information systems (GIS). It interactively couples physical and digital models of a landscape so that users can intuitively explore, model, and analyze geospatial data in a collaborative environment. Conceptually Tangible Landscape lets users hold a GIS in their hands so that they can feel the shape of the topography, naturally sculpt new landforms, and interact with simulations like water flow. Since it only affords a bird's-eye view of the landscape, we coupled it with an immersive virtual environment so that users can virtually walk around the modeled landscape and visualize it at a human-scale. Now as users shape topography, draw trees, define viewpoints, or route a walkthrough, they can see the results on the projection-augmented model, rendered on a display, or rendered on a head-mounted display. In this paper we present the Tangible Landscape Immersive Extension, describe its physical setup and software architecture, and demonstrate its features with a case study.

CCS Concepts

•**Human-centered computing** → **Human computer interaction (HCI); Geographic visualization;** *Virtual reality;* Mixed / augmented reality;

Keywords

immersive virtual environments; augmented reality; tangible user interfaces; tangible interaction; landscape modeling; head mounted display; Oculus Rift

1. INTRODUCTION

Geographic information systems (GIS) are used to model, analyze, simulate, and visualize spatial patterns and processes. While GIS provide powerful computational tools for studying spatial phenomena and solving spatial problems, they can be challenging to use – requiring training and expertise. GIS's complicated interfaces and steep learning curve make it hard to rapidly test ideas, collaborate, or engage the public. Tangible interfaces for GIS enable multiple users to naturally, directly interact with geospatial data and analyses with their bodies affording an intuitive, collaborative environment that does not require training to use [7]. See Petrasova et al. and Amburn et al. for reviews of tangible interfaces for spatial modeling [5, 1].

Tangible Landscape – a tangible interface for GRASS GIS – couples a physical and digital model of a landscape through a real-time cycle of 3D scanning, geospatial computation, and projection [5]. Multiple users can interact with geospatial data, models, analyses, and simulations by manipulating a physical model of the landscape. Users' changes to the physical model are 3D scanned in real-time and imported into GRASS GIS, an open source GIS [4], for modeling, analysis, or simulation. The result is then projected back on the physical model so that users can evaluate their modifications. As a collaborative environment for exploring geographic data and solving spatial problems Tangible Landscape enables experts and non-experts alike to test how changes will impact the landscape.

Because Tangible Landscape represents landscape with projection-augmented, scale models that are seen from a bird's-eye view at a geographic scale, users do not experience the landscape at a human-scale. While early tangible interfaces for geospatial modeling like Illuminating Clay [6] used renderings to visualize human-scale views, we implemented an immersive virtual environment for a more engaging human-scale experience. We developed the Tangible Landscape Immersive Extension[1] that couples Tangible Landscape with an immersive virtual environment so that human-scale views can be rendered on devices like head-mounted displays (HMD). Users can define viewpoints and routes for walkthroughs with the tangible interface, see the view rendered on a display, and then experience the view with a HMD. For digital 3D modeling and animation we use Blender [2], a free and open source program for 3D modeling, rendering, simulation, animation, and game design. Blender has add-ons for working with georeferenced data and immersive virtual environments such as HMDs.

2. SYSTEM OVERVIEW

Figure 1 shows the physical setup of Tangible Landscape and the proposed extension. Tangible Landscape is com-

[1]http://tangible-landscape.github.io/immersive.html

SIGSPATIAL'16, October 31 - November 03, 2016, Burlingame, CA, USA

ACM ISBN 978-1-4503-4589-7/16/10.

DOI: http://dx.doi.org/10.1145/2996913.2996950

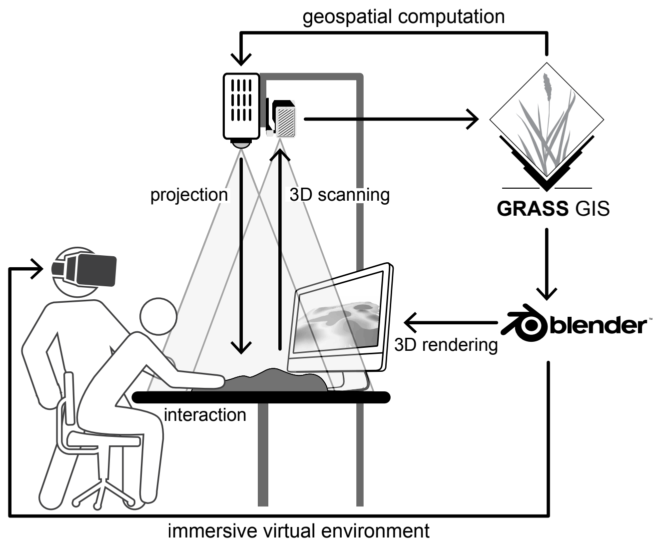


Figure 1: Physical setup of Tangible Landscape and the proposed extension

prised of 4 primary components: a malleable physical model, a 3D scanning device (Kinect for Windows v2), a projector, and a computer with GRASS GIS and software we developed for scanning and synchronizing all of the system components[2,3]. The scanner captures the shape of the physical model as the user modifies it and the point cloud is then georeferenced and transformed into a digital elevation model in GRASS GIS where all the subsequent geospatial analyses are performed. The resulting GIS layers are then projected back on the physical model to provide users with feedback. This extension adds two components to Tangible Landscape: 3D modeling software and an immersive virtual reality headset.

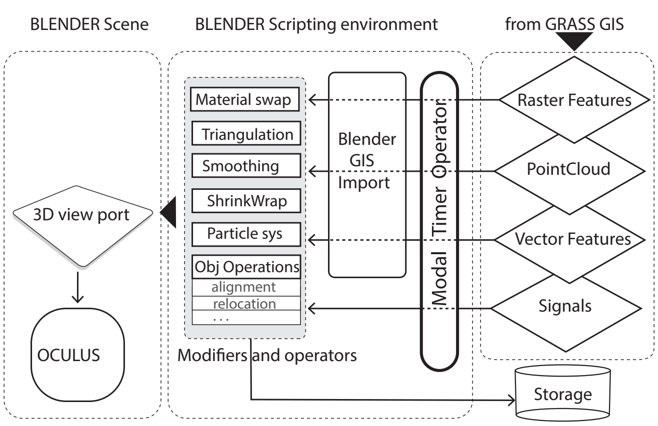


Figure 2: The application process flow diagram

Figure 2 illustrates the architecture of the Tangible Landscape Immersive Extension. GRASS GIS and Blender are loosely coupled through file-based communication. GRASS GIS (either continuously or when prompted) generates files with geospatial information in a specified system directory. Blender constantly monitors the incoming data and imports it for further processing using the *BlenderGIS* add-on module[4]. The history of the user's modifications can easily be preserved by storing the time-stamped, generated files in GIS formats. We setup and synchronized the monitored directory on a network drive to distribute the processing between two computers.

The system supports several modes of interaction (Figure 3) that generate four types of data: i) a point cloud in binary PLY format representing topography, ii) vector data in shapefile format including user defined tree patches, water features, walking routes, camera position, etc., iii) raster data in PNG format for texturing such as elevation maps, water flow, cost surfaces, orthophotos, etc., and iv) empty files signaling actions such as clearing the scene.

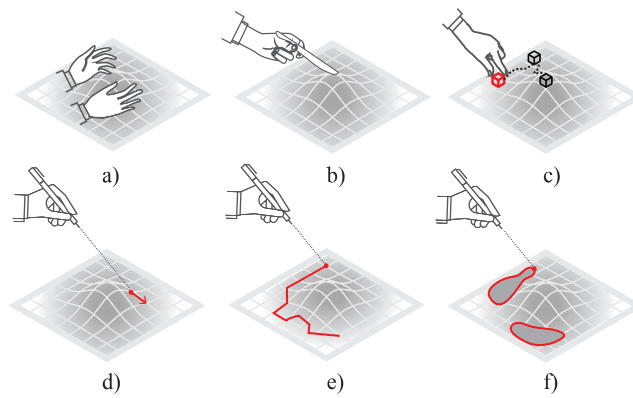


Figure 3: Interaction modes supported by Tangible Landscape: a) Sculpting topography by hand or b) with a tool. c) Placing markers to designate waypoints. d) Drawing viewpoints, e) walking routes, or f) patches with a laser pointer

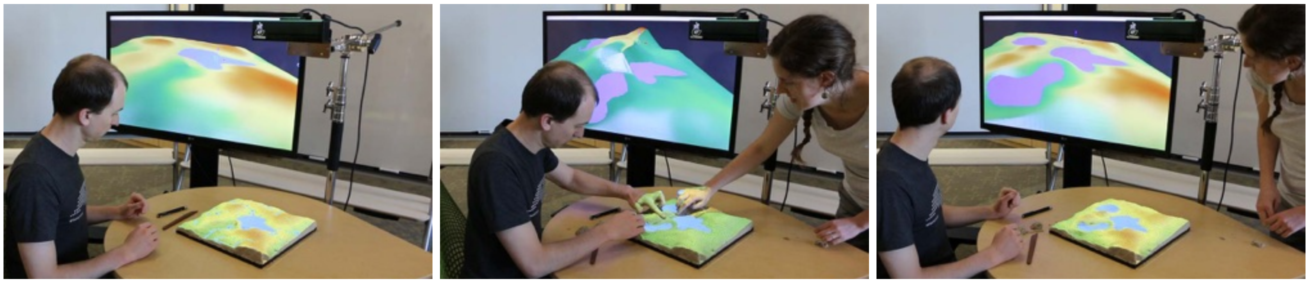
Point cloud data. The core of the virtual 3D model is the terrain mesh, which is processed from a point cloud. As the user reshapes the physical model by hand (Figure 3a) or with a sculpting tool (Figure 3b), the depth data are acquired by the 3D scanner, processed by the GRASS GIS add-on *r.in.kinect*[3], and exported as a georeferenced point cloud in binary PLY format. Once the point cloud file has been detected in Blender, a *Delaunay triangulation* function is applied to generate a 3D mesh, which is then smoothed with the *Smooth* modifier. All scene objects that were dependent or attached to the previous mesh are aligned to the new mesh using Blender's *ShrinkWrap* modifier.

Raster data. Existing raster data such as orthophotos or raster-based geospatial analyses are projected on the physical model and can be imported into Blender and applied to the 3D terrain as a surface texture. For example, when modeling surface water flow, the raster projections represent the evolving spatial pattern of water depth. In coastal storm surge modeling the raster data represent water depth in flooded areas.

Vector features. Vector features are generated when a user draws a path (Figure 3d) or a boundary (Figure 3e) on the physical model using a laser pointer or markers (Figure 3c). The 3D scanner senses color information as RGB

[2]https://github.com/tangible-landscape/grass-tangible-landscape
[3]https://github.com/tangible-landscape/r.in.kinect

[4]https://github.com/domlysz/BlenderGIS



a) b) c)

Figure 4: Sculpting topography to form ponds

values for each point in the scanned point cloud. While the user is drawing with the laser pointer, *r.in.kinect* iterates over all points in each scan and searches for the brightest point. If the brightness, computed as sum of RGB values, is above certain threshold, *r.in.kinect* starts to record the positions of the brightest points. When the user stops drawing, no point above the threshold is found and the shape is written in a vector file as a line or area. By placing markers we can designate waypoints for a trail across the landscape. Their position is detected from the difference between the scan before and after placing the markers. Once their positions are identified, they are connected by a least cost path. Cost can be defined, for example, as the slope of the terrain. The designed features are then projected back on the physical model and simultaneously imported and rendered in Blender. Once features have been imported into Blender and aligned, they can be enhanced with Blender modifiers and operators. A water texture, for example, can be applied to polygons representing water bodies. Blender's *Particle System* modifier can be applied to polygons representing tree patches with given species, density, and diversity to generate trees on the 3D terrain model.

Viewpoints. Users can designate a viewpoint by drawing a line on the physical model. The starting point represents the view-origin and the end point determines the direction of the view (Figure 3d). After importing the line, the camera is aligned 6 feet above the vector's first vertex to keep the view-origin at the eye level.

Flythroughs and walkthroughs. Users can draw a curve on the physical model using markers or a laser pointer to design a flythrough or walkthrough animation in Blender. The curve represents the route the camera will follow during the animation. Users select either a perspective view or bird's-eye view and the camera is aligned to the first vertex of imported vector at the given height. The scene is animated to simulate a walkthrough or flythrough. Since the model will continue to update while the animation is playing, the user can continuously observe how landscape changes as it is being modified.

Immersive experience. We used the Blender add-on module *virtual_reality_viewport* to render the camera view as a separate viewport for the immersive HMD[5]. Since this add-on supports multiple modes of visualization in separate viewports, a group of users can simultaneously experience different views of the same scene. For example while one user interacts with the physical model and sees the scene rendered on a display, another user can explore the scene from another view with the HMD.

[5]https://github.com/dfelinto/virtual_reality_viewport/

3. DEMONSTRATION

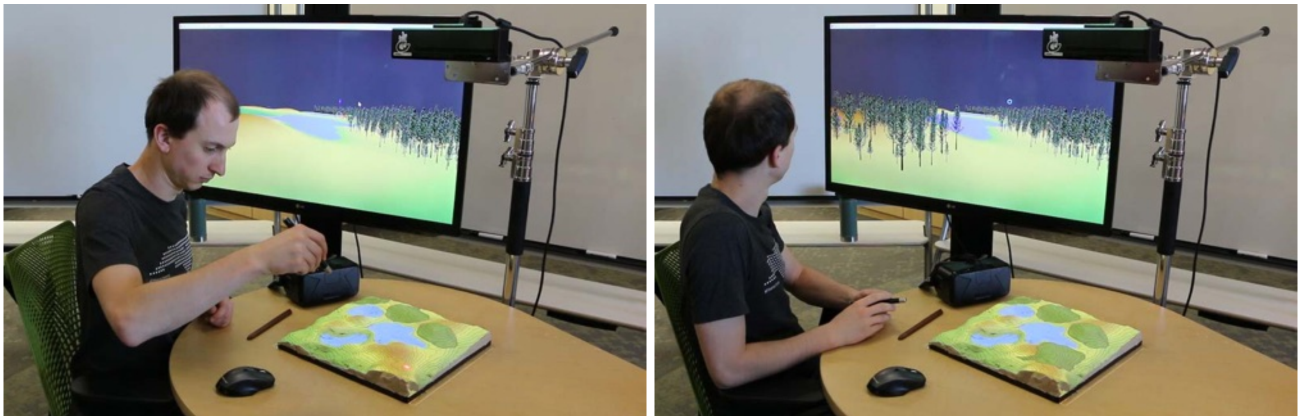
We demonstrate the coupling of Tangible Landscape with an immersive virtual environment with a case study in which users design a small park with water features, planting, and a recreational trail using the technology. Their goals were to control storm water runoff, create an aesthetic landscape, and plan a comfortable walking route with good views. They used polymer-enriched sand to build a malleable physical model that was easy to sculpt, yet would hold its form. They sculpted a scale model of a hypothetical, 50 ha landscape with several hills and a valley opening to the southwest.

Step 1: Ponding and landform design. Retention ponds are used by landscape planners and civil engineers to control and remediate storm-water runoff. Well designed retention ponds with planting can have aesthetic and recreational value. The users' first task was to design ponds for storm water management and aesthetics. The users modified the topography to create ponds and store runoff. Any soil excavated to create a pond had to be placed on site. Water flow and accumulation simulations were continuously projected onto the physical model to provide real-time feedback. The scanned and simulated data were also continuously imported into Blender for visualization. To simulate water flow during a storm event we used the module *r.sim.water*[6] to solve the shallow water flow continuity equation using a path sampling technique [3]. Water accumulation in the retention ponds was simulated by filling depressions in the terrain using the *r.fill.dir* module[7]. The Blender script processed the terrain point cloud, surface texture, and water boundary polygon and rendered the scene in 3D on a display. One of the users built a dam to form a large pond in the middle of the landscape (Figure 4a). Then both users collaboratively excavated two more ponds and built two small hills with the excavated soil (Figure 4b). They explored different views with the display to appraise the aesthetics of their design (Figure 4c).

Step 2: Vegetation design. In the second step one of the users designed patches of trees to frame views of the ponds. As the user drew patches with laser pointer the polygons were rendered on the physical model and filled with clusters of individual trees in the Blender scene (Figure 5).

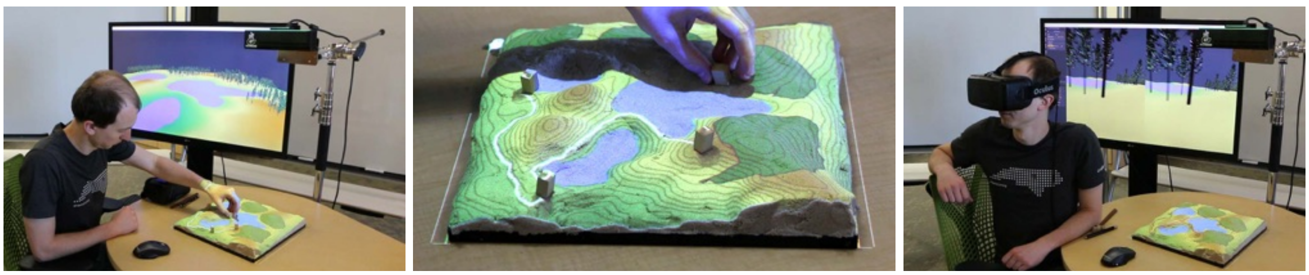
Step 3: Trail design. In the third step the users' goal was to design a cost-effective, easy to hike trail with good views. First the users placed markers on the physical model to identify waypoints along the new trail (Figure 6a, b). The optimal route between the waypoints was automatically

[6]https://grass.osgeo.org/grass72/manuals/r.sim.water.html
[7]https://grass.osgeo.org/grass72/manuals/r.fill.dir.html



a) b)

Figure 5: Drawing patches of trees with a laser pointer



a) b) c)

Figure 6: Designing an optimized trail

computed using the *r.walk*[8] algorithm to find the least cost walking path. *r.walk* can be parameterized to avoid high slopes and obstacles such as water bodies. The optimal trail route was continuously recomputed and projected onto the model to provide users feedback during the design process. The trails were also continuously updated in the Blender scene for visualization and evaluation.

Then one of the users adjusted the trail manually, drawing a new route with a laser pointer. The user adjusted the route of the computationally optimized trail to pass through the new forested patches and cross the saddle between the two new small hills. Finally the user put on the HMD, began the walkthrough simulation along the trail, and immersively experienced the landscape (Figure 6c).

4. CONCLUSION

Our application is a blueprint for an emerging model of geospatial education, collaborative decision making, and participatory design that synthesizes geographic scale analytics and human-scale perception. The hardware and software is inexpensive, open-source, and can be easily replicated. In the future we intend to enhance the fidelity of the rendering with improved texture mapping and a library of landscape elements such as different tree species. We also intend to evaluate user interaction and experience by assessing cognitive and meta-cognitive processes in landscape modeling and design problems.

[8]https://grass.osgeo.org/grass72/manuals/r.walk.html

5. REFERENCES

[1] C. R. Amburn, N. L. Vey, M. W. Boyce, and M. J. R. Mize. The Augmented REality Sandtable (ARES). Technical report, US Army Research Laboratory, 2015.

[2] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Blender Institute, Amsterdam, 2016.

[3] H. Mitasova, C. Thaxton, J. Hofierka, R. McLaughlin, A. Moore, and L. Mitas. Path sampling method for modeling overland water flow, sediment transport, and short term terrain evolution in Open Source GIS. *Developments in Water Science*, 55:1479–1490, 2004.

[4] M. Neteler, M. Bowman, M. Landa, and M. Metz. GRASS GIS: a multi-purpose Open Source GIS. *Environmental Modelling & Software*, 31:124–130, 2012.

[5] A. Petrasova, B. Harmon, V. Petras, and H. Mitasova. *Tangible Modeling with Open Source GIS*. Springer International Publishing, 2015.

[6] B. Piper, C. Ratti, and H. Ishii. Illuminating clay: a 3-D tangible interface for landscape analysis. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '02*, Minneapolis, 2002. ACM Press.

[7] C. Ratti, Y. Wang, H. Ishii, B. Piper, D. Frenchman, J. P. Wilson, A. S. Fotheringham, and G. J. Hunter. Tangible User Interfaces (TUIs): A Novel Paradigm for GIS. *Transactions in GIS*, 8(4):407–421, 2004.