A Real Time Immersive Virtual Reality Testbed - Project Completion Report
HMN1582-09-1-0013

Thomas J. Pingel
Department of Geography
Northern Illinois University, DeKalb, IL 60115

Keith C. Clarke
Department of Geography
University of California, Santa Barbara, CA 93106

March 2013

Abstract
This paper summarizes recent efforts in the use of automation to create geographic immersive virtual environments on a testbed of the University of California, Santa Barbara campus and nearby neighborhood of Isla Vista. In particular, we discuss how aerial LiDAR point clouds can be processed and fused with aerial video sources, and distributed on the web via the X3D language.

Keywords
Virtual Reality, Immersion, LiDAR, Automation, X3D

Introduction

Immersive virtual reality descriptions of geographically related content remain a relatively underdeveloped area for research, particularly when it comes to producing a scene which adds value to the task of analyzing a recent or ongoing event at a particular locality. The UCSB virtual reality test bed project seeks to uncover aids to the construction and communication of just such scenes. Here, we deal with some of the more technical and representational aspects of the construction, with an emphasis on automation. Automation is an important aspect of construction; most methods currently in use require a good deal of manual labor by technicians and artists. However, given the unpredictability of events, we assume that although many kinds of geodata will be available to provide context for an event at any given location, detailed models about the terrain, the three-dimensional building structure, and the movement patterns of humans are likely not to exist. Therefore, we investigate the construction of such scenes from technology that can be rapidly deployed and processed – airborne laser scans and video imagery and audio streams captured from drones or other autonomous vehicles.

In this case we intend "immersive" to refer to any virtual reality (VR) representation in which the user views his or her environment from a perspective view, and can freely move around in that environment – by simulated walking, flying, or other locomotion – as if he or she were moving in a real environment. This would exclude more commonly used representations in geography – like maps and satellite images – since they are typically displayed orthographically. It would also exclude perspective representations such as images taken directly from a camera source, since the user has no active control over his or her locomotion. A commercial product like Google Street View - where panoramic images have been stitched together to form a nearly cohesive whole - comes close to immersion, but locomotion through the Street View is cumbersome. Within a single panoramic image, the user can look about in a reasonably natural way, but the interpolated phase in transition between images is largely unnatural. The definition of immersion presented above nearly corresponds with Björk and Holopainen's concept of "spatial immersion" (2004). It is worthwhile to note that spatial immersion is not the only kind of immersion: Björk and Holopainen describe sensory-motor immersion, cognitive immersion, and emotional immersion. Any of these may ultimately be important for the communication of a geographic virtual reality; in this case we largely confine ourselves to those problems of immersion where perspective and locomotive fidelity are important.

Immersion in this sense has been the focus of both academic and commercial efforts. One of the more prominent academic efforts is the Virtual Tübengen Project (e.g., Riecke et al., 2006; Meilinger et al., 2008; Trutoiu et al., 2009). The model currently has approximately 200 structures covering an area approximately 500 m x 150 m. It is constructed using largely in-house developed software, and stored in relatively common 3D formats (e.g., 3ds Max). Virtual Tübengen was created primarily for the purpose of exploring human spatial cognition in a naturalistic setting. Its construction was derived from building blueprints which were converted manually to 3D models, and which were then textured using rectified photographs of individual facades taken by the researchers. The scene is configurable, though static – a kind of "museum" virtual reality in which individual models are not interactive (Hartman et al., 1996). As a point of comparison, our model of the UCSB campus and surrounding area comprises an area 3.25 km by 1.6 km with thousands of buildings of various sizes on moderately differentiated terrain. At this scale, manual generation of models becomes prohibitively expensive in terms of time and monetary cost.

Commercially, the digital game community has driven most of the advancement in immersive VR. Computer gaming is a $65 billion dollar global industry, and individual titles can bring in staggering amounts of revenue: World of Warcraft has grossed over $10 billion in sales and 10 million copies sold since its 2004 release, and benefits largely from its extended gameplay and longevity (Digital Battle, 2012). A more traditional console game like “Call of Duty: Black Ops” has a shorter game duration and a different (though equally impressive) sales profile with 23 million copies sold worldwide for a gross revenue of $1.5 billion. Budgets for these titles are similarly large – Grand Theft Auto 4 had a reported budget of $100 million and Halo 3 had a reported budget of $55 million (Digital Battle, 2010). High-end gaming – an area in which immersion is standard – has both high development costs and potential revenues.

Not only does the gaming industry drive technological development in representation, but it often does so for input/output device design. The Nintendo Wii system advanced the state of the art in 2006 by the use of “Wiimote” – a wireless remote control that used optical tracking in conjunction with accelerometers to increase the kinetic fidelity of gameplay. Wiimotes allow the player to control on-screen avatars via actions that closely resemble those in real life. For instance, to hit a virtual tennis ball, one holds the remote like a tennis racquet handle and swings it, just as one would swing a tennis racquet. In 2010, Microsoft released the Kinect for its Xbox 360 platform. This system used three cameras to track user movement without the need for the player to hold a controller at all, in what Microsoft calls a “natural user interface.” The Kinect has recently been appropriated by academic researchers not only for immersive VR purposes (Kamel Boulos et al., 2011), but in order to build three-dimensional models of objects and structures in the real world, such as the interiors of caves (Mankoff & Gulley, 2012).

“Digital Earths” – those in which a virtual model of the globe is used to transmit geographic information – are now commonplace thanks to the widely successful Google Earth (GE) project. GE features fused satellite imagery with global terrain data to provide a very realistic experience of interacting with a model of the Earth. Interaction with GE can take several forms, including an object-like interaction where the model is interacted with as if it were a globe, a ground-level interaction in which the user can walk around a particular place, and a flight-simulator interaction in which the user pilots a virtual airplane. GE pioneered the use of the Keyhole Markup Language (KML) to annotate the globe with points, lines, and polygons, and it was an early adopter of the COLLADA 3D representation system to place models of buildings in the landscape.

Although the commercial sector has driven many important aspects of immersive VR, the uses for these technologies are still somewhat limited. Games, for instance, typically do not represent real places, and the terrain and building models for these structures are often highly simplified for both narrative and technological reasons. Unlike replicas of real places – often termed “mirror worlds” (Kamel Boulos and Burden, 2007), game spaces are often carefully crafted to give the appearance of a fully detailed landscape, while in reality they highly constrain the movement of players toward their next game objective. Technologically, simplified terrain and building structures reduce computational demands on the system by allowing the same structures to be re-used many times. Different texturing schemes can hide these differences, giving the appearance of a varied land or cityscape. Even so, many of these games feature environments that are paradoxically real-looking but undifferentiated. Many users of these games report feeling “lost” inside them, presumably because cues that exist in the real-world to help them encode location are missing inside the gaming world.

GE suffers from its own limitations. First, details of the terrain cannot be altered, and therefore higher-resolution terrain models derived from local laser scans cannot be incorporated. In fact, although images, annotations and static models are relatively easy to include, the incorporation of higher-level adjustments to the interface are difficult or impossible. As such, GE is mainly used descriptively rather than analytically. Finally, the coverage of three-dimensional models of structures is unevenly distributed. While high-density, high-population urban areas like New York City or Seattle are modeled quite well, smaller cities and other urban areas contain models only when individual users have created and submitted them.

While efforts to create immersive VR models of the Earth are important and useful, there are good reasons to focus attention on immersive VR models of particular localities as well. First, inclusion of a high-resolution local terrain could be quite important. With regard to our testbed site, the University of California, Santa Barbara and the surrounding neighborhood of Isla Vista are located next to the Pacific Ocean on a peninsula which features cliffs approximately 20 meters above sea level. The ability to include the correct location and elevation of these cliff structures in detail was important, but perhaps more important was the inclusion of very subtle terrain differences associated with human paths like roadways, sidewalks, and other footpaths. These features are retained in the basic data for LiDAR scans, but are missing in the lower resolution ~10 m DEMs upon which larger extent models (e.g., GE) are based.

Second, we required the ability to incorporate video and audio sources into the model – something not currently supported by any Digital Earth product of which we are aware. Live video sources allow for on-the-fly texturing of the terrain and building surfaces, while the incorporation of spatially sensitive audio sources can add a great deal of information to the scene. Together with the "static" model of the physical space, a reasonably accurate, and more importantly, *fully automatable* scene may be created and viewed in real time.

Three aspects of the model which we wish to share with the mapping community at this time are 1) our thought process regarding the code base for the project, 2) a newly designed LiDAR filter aimed at retaining subtle ground features that may be important for visual guidance in an automatically constructed terrain, and 3) our process of video overlay for the model.

Code Base

Our model has been constructed using the Extensible 3D (X3D) language. X3D is the successor to the Virtual Reality Markup Language (VRML), developed in the 1990s as a standard for the communication of three-dimensional vector graphics (Brutzman & Daly, 2007; Geroimenko & Chen, 2005). The GeoVRML project was incorporated into the VRML97 specification to include a convenient way to reference geographic information, including elevation grids. X3D replaced VRML by utilizing the same basic syntax, but in an XML format. Since X3D natively supports both standard kinds of 3D vector formats as well as geographic coordinate systems and objects, we believe it represents a solid foundation as a code base for the project.

X3D documents technically describe scenes, which are aggregations of not only objects, but information about lighting, viewpoints, and locomotion. X3D scenes are read and rendered by browsers in much the same way as the more familiar Internet browsers render HTML content. Most of these X3D browsers are capable of reading a wide variety of objects encoded in languages other than X3D (e.g., COLLADA or older VRML-based models), which can be referenced by URL via an *Inline* node. In this way, X3D is intended to provide a means to bring together a wide variety of three-dimensional content, irrespective of format (Arnaud & Parisi, 2007). Further, X3D natively works well with JavaScript, which has emerged as the standard for HTML5 and many dynamic aspects of user interface and database integration. Finally, native video texturing and spatialized audio sources were part of the X3D specification from its beginning, thus sparing the effort of incorporating third-party applications to handle these elements.

X3D benefits from a strong research community. Evidence of this comes from German browser producer Instant Reality Labs, which is developing interesting new features as proposals to the next X3D version (Avalon). Some of these interesting developments include: better input/output device support for new technology like Wiimotes, Kinect Sensors, and the Space Mouse, an input device that allows the user to move objects with six degrees of freedom. In addition, many background computational tasks have been included as node objects, including automated steering behaviors and physics that make modeling object movement much less daunting for the average researcher. Entirely new nodes like Browser Textures that enable web-pages to be used as interactive textures open up new frontiers for design and interaction.

A second promising direction for X3D comes from its easy integration with HTML5 and WebGL via the X3DOM project. In this way, X3D scenes can be incorporated directly into web pages without the need for third-party plugins and their associated overhead. As a royalty-free standard, the effort finally places X3D in a position to become the lingua franca of the three-dimensional web-based graphics community. Under this model, three-dimensional scenes are blended seamlessly into web content without any action necessary on the client side.

X3D is a good choice not only for representing both static and dynamic objects, but also for more abstract visualization tasks (Geroimenko & Chen, 2005). To simply copy the real world into a similar digital form is not enough – ultimately, like maps, we want our immersive geographic virtual realities to enhance our understanding of the real world. In much the same way as augmented reality seeks to enhance our understanding of consensus reality, we aim to make our immersive geographic virtual worlds analytically interesting.

The Simple Morphological Filter to Produce Terrain for Immersive 3D Maps

One significant advantage of creating local models is the ability to define a ground source of much finer quality than that typically provided by any global-scale mapping software like Google Earth. A critical source for such high-resolution terrain data are laser scans taken from airborne platforms in the form of Light Detection and Ranging (LiDAR) technology. The typical base format of a LiDAR scan is a "point cloud" – a relatively unsorted collection of x,y,z points within the scan range. Such point clouds must be processed to classify individual returns – typically as a ground, vegetation, or building return. Several commercial software packages exist to do this, as well as dozens of academic algorithms in the literature (Meng et al., 2010).

Performance varies somewhat based on algorithms and their interaction with the point cloud. Typically, areas with a great deal of vegetation mixed with high relief prove the most troublesome for classification algorithms. Mistakes in classification can take the form of either a Bare Earth return misclassified as an Object (Type I) or as an Object return misclassified as a Bare Earth return (Type II). Many filters attempt to reduce the total error, or a weighted average of these two types of errors. When the DEMs produced from such algorithms are used in a purely quantitative way, this approach makes sense. In fact, it is often crucial to minimize Type II errors, since these kinds of errors often result in a highly distorted DEM, especially with the tops of buildings are mistakenly classified as ground. In contrast, Type I errors are generally more permissible, since gaps in the record can be filled in by interpolation techniques, so long as there are enough nearby ground points for a valid interpolation.

The minimization of Type II errors is similarly important for DEMs derived from LiDAR that are to be used visually. The visual errors associated with these kinds of errors produce landscapes that bear little resemblance to their real life counterparts. In many cases, other objects that are intended to be placed on or near the ground may be placed incorrectly as a result of such errors. An example of this may be found on the UCSB campus, where the very large library building (approximately 75 m x 175 m) causes a local error in the National Elevation Dataset derived ground layer used in Google Earth. When a COLLADA model of the library is snapped to this ground layer, it appears to float about 6 meters (i.e., the height of the local error) above the ground (Figure 1).

The minimization of Type II error is certainly important, but a profound reduction of Type I error offers many benefits of its own for the DEM consumed visually. Generally speaking, only a relatively small proportion of ground points from the original cloud are needed to create a DEM. For instance, the aerial scan from the UCSB campus area has 2-3 returns per square meter, depending on the particular location. As the campus is not particularly dense with either buildings or vegetation, most of these returns are ground. And, since we sought to produce a 1 m resolution DEM of campus as our ground layer for the immersive virtual reality, a large number of these returns are superfluous. To falsely reject some of these true ground points as objects would not interfere with the creation of the DEM unless they cluster into a large areal group – and particularly so when they occur at a location with an important feature of relief. These important features of relief may be quite large – cliffs and hilltops, for instance – but they might also be as small as the relief surrounding a subtle footpath. Reduction of Type I error means that these features are more likely to be retained. Large features are important to retain because they add character and realism to the landscape. In the analysis of movement, for instance, those features may be of paramount importance in determining why a particular agent moved through or around one location versus another. Since LiDAR has the capability to penetrate the vegetative canopies where visible wavelengths of light cannot, the retention of the more subtle features may be the only indication of footpaths that may otherwise be obscured.

Since most ground classification algorithms for LiDAR points clouds are typically oriented toward a balanced reduction in Type I and Type II error, we developed a ground filtering algorithm called the Simple Morphological Filter (SMRF). SMRF follows other progressive morphological filters (e.g., Zhang et al. 2003; Chen et al. 2007) but features some important differences with regard to the way tolerance values are calculated, the rate of increase for the opening window, and in the interpolation basis. The result is a filter that features an extremely low Type I error rate, while still providing a reasonable Type II error rate. No other published filter of which we are aware features either a lower total error or Type I error rate when tested against a benchmark set of LiDAR scans developed by the International Society of Photogrammetry and Remote Sensing (Sithole & Vosselman, 2004). The SMRF filter is described in full (Pingel et al., 2013) and is published online (http://www.tpingel.org/code/smrf).

The LiDAR point cloud from an aerial scan is often dense enough to recreate the forms of buildings, but reconstruction of vegetation can be a problem, especially when the number of species is artificially high as is often the case in manicured urban environments. Overlay of aerial imagery is, of course, a viable option. One interesting application of the LiDAR data is the transformation of the Digital Surface Model (DSM) into an image that better represents the structure of objects on the ground. Traditionally, with terrain, this is accomplished by means of a hillshaded relief map (Imhoff, 2007). In the case of urban areas, we find that a hillshaded image can obscure the structure on the ground to some degree. This happens because a hillshaded image takes into account the aspect of the slope. Aspect is quite important in representing natural environments since these are typically very irregular. In contrast, the built environment benefits from a multitude of right angles and standardized slopes. For this reason, aspect becomes much less important, and its explicit inclusion into the visualization adds information that can be quite distracting and in some cases suppresses the visibility of edges that may help to visually delineate structures.

An alternative is to simply map slope to a gray tone. In this way aspect is lost, but the shading of built slopes and vegetation generated from the DSM creates an illusion of a hand-drawn rendering of the environment. Ongoing work on this project attempts to assess empirically the drawbacks and benefits to this visualization, including a non-linear transformation of slope to gray-tone that emphasizes shallow slopes and de-emphasizes steeper ones by means of a vertical exaggeration transformation. Evidence suggests that humans overestimate slope largely because they overestimate vertical heights in environmental-scale features (Proffitt et al., 1995; Yang et al., 1999; Pingel, 2010). For this reason, we transform the raw slope values by exaggerating the vertical dimension by a factor of between two and three times before mapping slope to gray tone. The result of such a transformation is shown below in Figure 2. We call such vertically exaggerated slope-shaded images *bonemaps*. Bonemaps can be created from DSMs or for middle stages of processing by the SMRF algorithm, which removes most of the vegetation while retaining structures.

The benefit of bonemaps for use in geographic immersive virtual reality applications is that since the image and structure of the environment come from the same source (an airborne laser scan) there are never any issues of image registration that can be challenging for automated generation of true orthophotos. Correction of orthophotos commonly in use will take into account a terrain model derived from an elevation layer, but this elevation layer is typically much coarser in resolution than that of the image. This usually results in error near areas that have not been modeled accurately (e.g., areas where the terrain changes far more rapidly than the resolution can adequately describe). Importantly, this means that buildings are not taken into account, and as such distortion is typically noticeably poor around such structures. True orthophotos attempt to take this into account. In this case, bonemap images – while replicating the visual experience – really only describe the spatial structure. Still, much information is encoded in the spatial structure, and since it is always accurate to within the limits of the GPS associated with the original encoding of the LiDAR data, bonemap (and other LiDAR-derived image) visualizations have the benefit of being highly spatially accurate compared to orthorectified imagery.

Empirical work testing bonemaps against several other popular geovisualization formats (hillshades, elevation shading, and orthophotos) suggest that bonemaps feature a higher throughput of data per unit time compared to other geovisualizations. Details of these findings are to be presented at the American Association of Geographers Annual Meeting in 2013, and subsequently submitted for publication in a peer-reviewed journal. Bonemaps developed from this project were used to great effect on a LiDAR test-site in Guatemala and Belize (Pingel, 2012).

Incorporation of Real-Time Video in X3D Scenes

In terms of automated production of a geographic immersive virtual reality scene, video overlay provides a key conduit in the production of a local urban scene. X3D supports video texturing natively, although there remain serious obstacles in terms of the ability to support multiple high definition overlays at full frame rates. Still, imagery captured from an aerial platform (e.g., an unmanned aerial vehicle or UAV) can provide a wide area of coverage at very high resolutions on demand. These sources can be imported directly into X3D, though real-time, non-local sources remain somewhat difficult to smoothly integrate into a scene. Planned advancements to the X3D specification - as well as some proposed standards available in Avalon - address some of these issues.

In terms of video acquisition, our model makes use of some recent advancements in simple aerostat designs published by the Grassroots Mapping organization. Simple aerostats can be constructed with relatively common materials. In our case, we use metallized Biaxially-oriented polyethylene terephthalate (BoPET) as an envelope and helium as the lift source. BoPET sheeting is commonly available as lightweight emergency blankets and for greenhouse insulation, since it retains heat very well. Balloon and kite photography has undergone a renaissance in recent years thanks to the ready availability of materials, and especially to advancements in lightweight digital camera equipment. Balloon-based images have the additional advantage of being able to fly in calm to moderately windy conditions, and some helium aerostats (e.g., the Heli-Kite) have been designed to fly in very heavy winds. One crucial advantage of helium aerostats in urban areas lies with the fact that since they are tethered, they are free of many of the restrictions that govern the more maneuverable rotary-wing UAVs. Although the Federal Aviation Administration (FAA) has recently widely expanded the role of UAVs in United States airspace, and may continue to do so in the future, helium-based systems still serve a vital service in that they can provide continuous coverage of a single area for an extremely extended duration of time. This particular capability makes them an invaluable tool for monitoring ongoing events like local natural disasters.

In order to provide geographic context information, we use a smartphone on the Android platform with video, audio, GPS, compass, and inertial sensors, and relay that information back to the ground in real time via a WiFi connection. By separating the sensor platform from the vehicle, we believe we can keep the sensor package as current as possible at the lowest cost. At the same time, the flexibility afforded by the operating system allows us to write customized software for the embedded sensors, and to capture, preprocess, and relay that information back to the ground station in real time. Untethered UAVs offer exciting new capabilities of their own, but given their pronounced costs and limitations, we believe the smartphone / aerostat combination provides the best value, and allows us to shift our focus from the technical side of video and audio acquisition toward investigating how such data can be integrated with three-dimensional models in a way that adds value to discrete event analysis.

Future Work

One important direction for future work is the extraction of three dimensional objects from the video feed. While video overlay is a useful way to view a dynamic scene in a very automated way, the extraction of geospatial objects from the video feed will greatly enhance the analytic possibilities. We are also excited to incorporate some of the latest 3D building reconstruction techniques, especially when fused with very high resolution terrestrial laser scan sources.

We are currently in the process of empirically testing the efficacy of several of our visualization schemes for three-dimensional models. One key aspect of the construction of three-dimensional scenes for event analysis is that they must be built rapidly. It also suggests that hyper-realism, often the goal in game development, is probably not feasible given the limitations of development costs and the likely inferiority of automated methods versus manual development by skilled 3D artists. However, humans are very used to dealing with abstract representations, and there is good evidence to suggest that such abstractions may actually be of more benefit in understanding a scene than complete realism. Cartography has long concerned itself with identifying appropriate and useful generalization and symbolization methods, and there is no reason to suppose that such a focus will be needed any less in the virtual world than it has in the printed and digital spheres thus far.

References

Arnaud, R. and Parisi, T. (2007) Developing Web Applications with COLLADA and X3D: A White Paper. http://www.khronos.org/ collada/presentations / Developing_Web_Applications_with_COLLADA_and_X3D.pdf Last visited 8/15/2012.

Brutzman, D. and Daly, L. (2007) *X3D: 3D Graphics for Web Authors*. Morgan Kaufman Publishers.

Björk, S. and Holopainen, J. (2004) *Patterns in Game Design*. Hingham, MA: Charles River Media.

Chen, Q. Gong, P. Baldocchi, D. and Xin, G. (2007) Filtering Airborne Laser Scanning Data with Morphological Methods. *Photogrammetric Engineering and Remote Sensing*. 73, pp. 175-185.

Digital Battle (2010) Top 10 Most Expensive Video Game Budgets Ever. http://digitalbattle.com/2010/02/20/top-10-most-expensive-video-game-budgets-ever/ Last visited 8/15/2012.

Digital Battle (2012) Top 10 Highest Grossing Video Games Ever. http://digitalbattle.com/2012/02/21/top-10-highest-grossing-video-games-ever/ Last visited 8/15/2012.

Geroimenko, V. and Chen, C. (2005) *Visualizing Information Using SVG and X3D: XML-based technologies for the XML-based Web.* London: Springer-Verlag Ltd. Hartman, J. Wernecke, J. and Silicon Graphics. The VRML 2.0 Handbook: Building Moving Worlds on the Web. Boston: Addison-Wesley.

Imhoff, E. (2007) *Cartographic Relief Presentation*. Redlands, CA: ESRI Press.

Kamel Boulos, M., Blanchard, B., Walker, C., Montero, J., Tripathy, A., and Gutierrez-Osuna, R. (2011) Web GIS in Practice X: A Microsoft Kinect Natural User Interface for Google Earth Navigation. *International Journal of Health Geographics*, 10(45). http://www.ij-healthgeographics.com/content/10/1/45. Last accessed 3/5/2013.

Kamel Boulos, M., and Burden, D. (2007) Web GIS in Practice V: 3-D Interactive and Real-Time Mapping in Second Life. *International Journal of Health Geographics*. 6, 51. http://www.ij-healthgeographics.com/content/6/1/51. Last accessed 3/5/2013.

Mankoff, K., Gulley, J. (2012) A 3D Surface of the Interior of a Sub-Glacial Cave Acquired Using a Microsoft Kinect as a LIDAR Sensor. Paper presented at the Association of American Geographers Annual Meeting, New York City, New York.

Meilinger, T. Knauff, M. and Bülthoff, H. (2008) Working Memory in Wayfinding - A Dual Task Experiment in a Virtual City. *Cognitive Science*. 32, 4, pp. 755-770.

Meng, X. Currit, N. and Zhao, K. (2010) Ground Filtering Algorithms for Airborne Lidar Data: A Review of Critical Issues. *Remote Sensing.* 2, pp. 833-860.

Pingel, T. (2010) Modeling Slope as a Contributor to Route Selection in Mountainous Areas. *Cartography and Geographic Information Science*. 37, 2, pp. 137-148.

Pingel, T. (2012) Automatic Methods of LIDAR Visualization: A Test Case in the El Pilar Archaeological Reserve for Maya Flora and Fauna. Paper presented at West-East Lakes Joint Meeting of the American Association of Geographers, DeKalb, IL, 25-27 October.

Pingel, T., Clarke, K., McBride, W. (2013). An improved simple morphological filter for the terrain classification of airborne LIDAR data. *ISPRS Journal of Photogrammetry and Remote Sensing.* 77, 21-30.

Proffitt, D. Bhalla, R. Gossweiler, R. and Midgett J. (1995) Perceiving Geographical Slant. *Psychonomic Bulletin and Review*. 2, 4, pp. 409-428.

Riecke B. Schulte-Pelkum J. Avraamides, M. von der Heyde, M. and Bülthoff, H. (2006) Cognitive Factors can Influence Self-Motion Perception (Vection) in Virtual Reality. *ACM Transactions on Applied Perception.* 3, 3, pp. 194-216.

Sithole, G. and Vosselman, G. (2004) Experimental Comparison of Filter Algorithms for Bare-Earth

Extraction from Airborne Laser Scanning Point Clouds. ISPRS Journal of Photogrammetry and Remote Sensing. 59, pp. 85-101.

Trutoiu, L. Mohler, B. Schulte-Pelkum, J. and Bülthoff, H. (2009) Circular, Linear, and Curvilinear Vection in a Large-screen Virtual Environment with Floor Projection. *Computers and Graphics*. 33, 1, pp. 47-58.

Yang, T. Dixon, M. Proffitt, D. (1999) Seeing Big Things: Overestimation of Heights is Greater for Real Objects than for Objects in Pictures. *Perception*. 28, 4, pp. 445-467.

Zhang, K. Chen, S. Whitman, D. Shyu, M. Yan, J. and Zhang, C. (2003) A Progressive Morphological Filter for Removing Non-Ground Measurements from Airborne Lidar Data. *IEEE Transactions on Geoscience and Remote Sensing*. 41, pp. 872-882.

Figures

Figure 1: An error in the underlying DEM causes the Davidson Library on the UCSB campus to appear to float 6 meters above the ground within Google Earth. This type of error highlights the need for enhanced ground filtering algorithms such as SMRF.

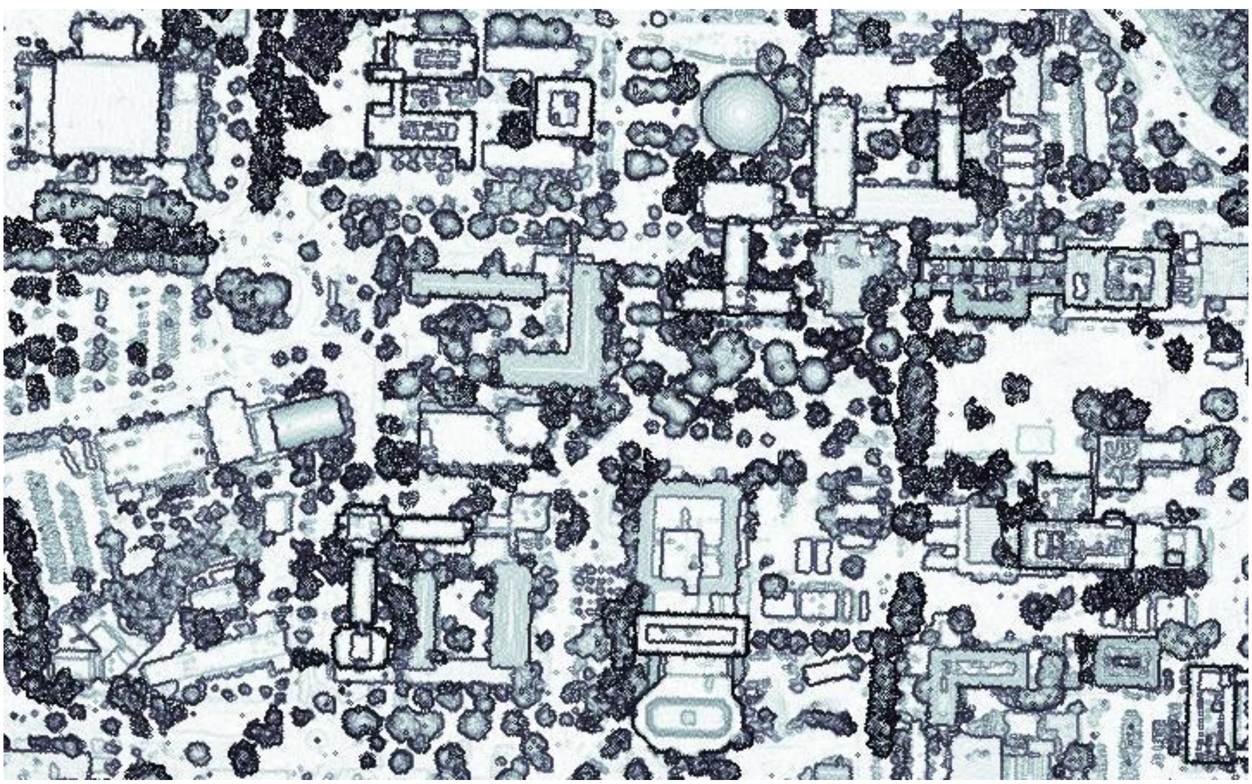

Figure 2: A mapping of slope to gray tone of the LiDAR-derived 1 m DSM of the UCSB central campus. Although the image is generated entirely automatically, it appears hand-drawn, suggesting a correct parsing of the environment in a human-readable way.