Workshop on Visualisation in Environmental Sciences (EnvirVis) (2014)
O. Kolditz, K. Rink and G. Scheuermann (Editors)

Point-Based Rendering of Forest LiDAR

Lance Simons[1] Stewart He[1] Peter Tittman[2] Nina Amenta[1]

[1] University of California, Davis [2] University of California, Berkeley

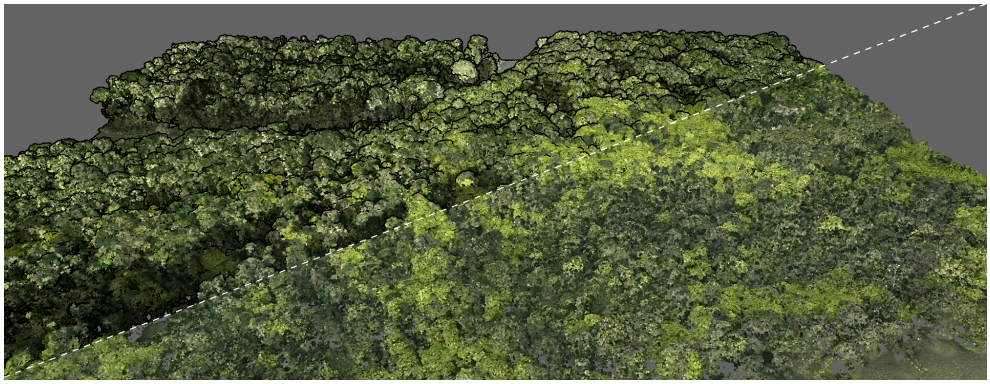

Figure 1: Tropical forest from La Selva Biological Research Station in Costa Rica. Bottom, a standard rendering of an airborne forest LiDAR point-cloud, with a photographic texture map. Top, our point-based rendering adds silhouettes, occlusion and shadow mapping to enhance the structure of the complex forest canopy, particularly gaps.

Abstract

Airborne Light Detection And Ranging (LiDAR) is an increasingly important modality for remote sensing of forests. Unfortunately, the lack of smooth surfaces complicates visualization of LiDAR data and of the results of fundamental analysis tasks that interest environmental scientists. In this paper, we use multi-pass point-cloud rendering to produce shadows, approximate occlusion, and a non-photorealistic silhouette effect which enhances the perception of the three-dimensional structure. We employ these techniques to provide visualizations for evaluating two analysis techniques, tree segmentation and forest structure clustering.

Categories and Subject Descriptors (according to ACM CCS):
I.3.3 [Computer Graphics]: Picture/Image Generation—Viewing algorithms

1. Introduction

One of the challenges environmental scientists confront as part of the process of documenting, mitigating, and adapting to climate change is monitoring the status of forests and jungles. Detailed information about forest health and structure is needed to monitor deforestation and/or reforestation efforts, and to protect warmer and dryer forests from wildfire and infestation. The three-dimensional point cloud information produced by airborne *Light Detection And Ranging* (LiDAR) is an important source of information; see [HHL*08] for a fairly recent review of LiDAR analysis in forestry. New initiatives, such as the airborne LiDAR component in the massive US NSF National Ecological Observatory Network (NEON) [Nat12] project, emphasize the increasing need for LiDAR analysis tools.

Exploratory visualization of forest LiDAR and of the results of analyses are both important tasks. They are not straightforward since the three-dimensional point clouds produced by forest LiDAR essentially contain no smooth surfaces, except for the ground, so many point-based rendering techniques, particularly involving normals or splats, are not useful. We use a combination of multi-pass real-

DOI: 10.2312/envirvis.20141105

time rendering techniques that do not rely on the presence of surfaces. This provides enhanced occlusion and an intuitive, non-photorealistic silhouette effect, as well as real-time shadows, to greatly enhance the perception of the three-dimensional forest structure.

2. Prior Work

Most software for processing and analyzing forest LiDAR, such as Fusion/LDV [McG09], ArcGIS [Sum11], lastools [IS07], TerraScan [Ter13], or LiDAR Viewer [Kre13], include point cloud visualization. But because forest LiDAR data lacks smooth surfaces, interactive motion, slicing, and coloring the points by height are the main techniques employed; Fusion/LDV and LiDAR Viewer can also use stereo. These visualizations are, in general, quite difficult to interpret.

The lack of smooth surfaces, and hence of point normals, prevents the use of standard physically-based rendering models. In particular, common point-cloud renderings based on lighting and blending point splats to give the appearance of surfaces, eg. [ZPVBG01, RPZ02], are not helpful. Most point-based NPR algorithms [ZS04, XC04, PKG03] also need to estimate surface normals or detect edges. Our multi-pass NPR silhouette rendering technique is similar to an approach for terrestrial LiDAR data due to Xu *et al.* [XNYC04]. We improve the interaction between neighboring objects using a depth-aware approach also used for NPR rendering of lines [EBRI09]. The silhouette technique is also related to a well-known z-buffer trick for hidden line removal, and to the more modern idea of depth buffer priming for *z-culling*.

An important issue in point-based rendering is managing large point clouds through level-of-detail (LOD) data structures, e.g. [RL00, BWK02, PGK02]. This issue is very relevant to forest LiDAR, but it is not the subject of this paper; we find that reasonably large LiDAR data sets can be handled directly by current graphics hardware.

3. Rendering

The input to our rendering algorithm is a set of points, potentially with color information, either from a photographic texture or representing the output of some analysis operation. No point normals are provided. Our first rendering pass generates shadow maps for lighting, and our second pass is a depth-buffer technique for the silhouette effect and enhanced occlusion. A final pass completes both the shading and the silhouettes, along with color.

Shadow map pass: We employ a point-based version of standard shadow mapping, a venerable technique [Wil78, RSC87] well-suited to modern graphics hardware [ERC01]. Simply put, if an object is visible from the point of view of the light source, then there is a path from the light to the object and that object is illuminated. Since the scene is static, we update the shadow map, only when the user changes the direction of the light source, improving performance.

Silhouette and occlusion pass: The entire scene is rendered in an initial silhouette and occlusion pass, storing both color and depth information as usual. Using the stored depth information in a z-test in the final pass (below) prevents occluded fragments from appearing in the output image. We render the points larger in this depth pass than they will be in the final pass. This occludes points that are further away from the viewer. We color the enlarged points black, so that when the points are rendered with their proper color in the final pass, they appear to have a black border.

Figure 3: Rendered as disks normal to the viewing direction (downwards), points near each other can form an occluding sequence and appear all black. When rendered as paraboloids (right), like the tapering used in [EBRI09], this is much less likely.

One issue we found is that when a dense set of points have the same (or very similar but monotonically changing) depth values, they can create a sequence of points for which p_1 occludes p_2, which occludes p_3, which occludes p_4, and so on, creating thick black lines or regions; see Figure 3. With forest LiDAR, this occurred on trees in the distance, where depth values are low-resolution, and for LiDAR scan lines of points on the ground in some views. We solved this problem by increasing the depth value of each fragment in the silhouette depth buffer by its squared distance to the center of the rendered point, effectively replacing each silhouette disc with a parabolic cap, with the rounded end pointing toward the camera. Neighboring points are no longer occluded if they have the same depth value.

Points very near the camera each have their own silhouette, so that benefits of the rendering technique (occlusion, indication of depth discontinuities) are lost. We get a more comprehensible visualization by increasing the size of the silhouettes for near-field points, which tends to fill the gaps between foreground points, maintaining the impression of a solid object.

Shading pass: Our final pass uses the depth buffer from the silhouette pass as well as the stored shadow map from the shadow pass. Depending on the rendering mode, the shader selects the per-fragment color from a choice of the original color data stored with the LiDAR data, color data representing the results of some analysis, or coloring based on the elevation of the point. The z-buffer test determines whether the fragment will be rendered. The shadow-map depth test is performed, and the fragment is shaded accordingly.

Design Decisions: We settled on this combination of silhouettes, occlusion, and shadows after experimenting with

Figure 2: A comparison of several rendering techniques in isolation. From left to right: raw point cloud, shadowing, depth cueing, silhouetting, and ambient occlusion. All examples are rendered with textures supplied with the data. Depth cueing is almost completely drowned out when rendered with colored points.

several point cloud visualization methods. Most point-based rendering techniques compute and uses a normal vector at each point. We tried generating normals by fitting a plane to a Gaussian-weighted neighborhood around each point. This inherently introduces some smoothing. Using small neighborhoods gave normals that were quite noisy, creating a distracting glittery effect when used for lighting. Enough smoothing to produce coherent lighting eliminated useful detail.

While depth cueing works well on it's own, when combined with silhouetting its effects are largely overridden. It also interferes with shadowing and obscures data in the background.

While the images produced by ambient occlusion with forest LiDAR are aesthetically pleasing and do enhance the perception of surface shape better than simple shadowing, especially in still images, the computational cost resulted in a significant drop in performance. Common performance improvements for ambient occlusion, such as calculating shading at a lower resolution than the framebuffer, resulted in increased noise and graininess and did not provide sufficient performance gain.

Data, Implementation, and Performance: The dataset in Figure 1 is a wet tropical area, provided by the Tropical Ecology Assessment and Monitoring (TEAM) Network [Tro09]. The Northern coniferous forests in Figures 4 and 5 are from Huboldt County, California and Panther Creek, Oregon. We use tiles containing 2.9, 1.7, and 2.2 million points respectively, each taken from larger datasets containing several hundred million points. Point density varied from 2.9 pt/m^2 in Costa Rica to 19.8 pt/m^2 in Panther Creek.

Our code uses OpenGL 3.3, with Python 2.7 and Pyglet 1.1.4. To maintain interactive performance, we try to keep as much processing on the GPU as possible, so much of the work is done in fragment shaders. On all of our three example datasets, we achieve rendering frame rates of at least 20fps on an NVIDIA GTX 480.

4. Applications

One of the interesting research frontiers in forest LiDAR analysis is the segmentation of individual trees from the forest canopy, and from sub-canopy vegetation to the extent possible [HI99, PW04, KHW06, KLL*07, LHN13]. Tree segmentation is challenging, and since ground-truth data is expensive to collect and difficult to align with LiDAR (GPS measurements tend to be inaccurate under heavy forest canopies), tree segmentations are also difficult to validate.

With our improved rendering, we can use visualization to estimate whether a computed tree segmentation is plausible. In Figure 4 we visualize the results of the *StarSac* tree segmentation program of Shafii *et al.* [SHH*09, TSHH11] on LiDAR from the Humboldt county forest. In Figure 4, we examine an output segmentation, indicated by color, with our visualization. This visual inspection can then be used as feedback to tune parameters in the segmentation algorithm.

Another form of analysis is classifying regions of the forest based on their three-dimensional structure. For instance, in [SHTA13] the three-dimensional point distributions in small vertical sections of the LiDAR point cloud are clustered based on a metric for comparing distributions. In Figure 5 we an unsupervised clustering result, using slicing to see "into" the dense forest from the side. With our rendering it is easy to verify that the clustering does identify patches with differing forest structure, and also allows the user to interpret the meaning of the clusters.

5. Evaluation

We asked 12 PhD students in forest ecology to evaluate this visualization as a verification tool and as a tool for general forest browsing. As an example of a gestalt visualization task, each participant was asked to classify a forest plot as riparian (near a river) or not riparian. As an example of a verification task, we asked them to identify spurious trees generated by Fusion/LDV's automatic canopy peak finding al-

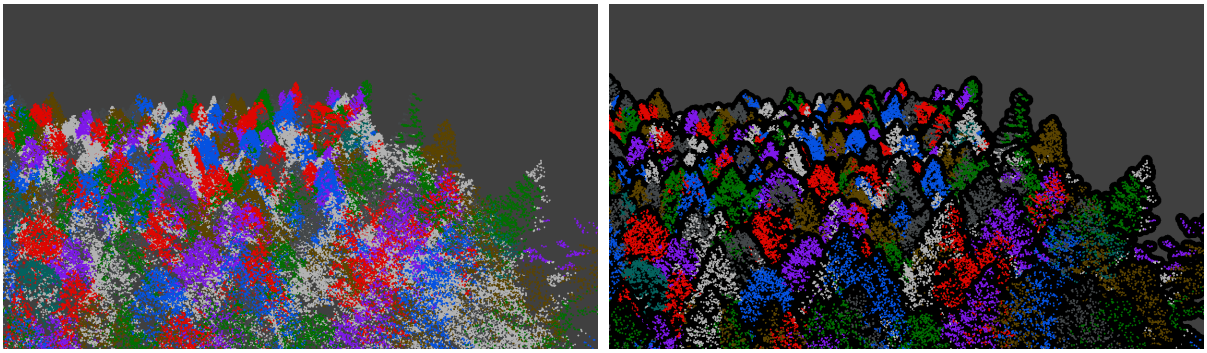

Figure 4: Part of the the colored point cloud indicating a segmentation of the California forest LiDAR dataset into individual trees, output by the StarSac software. Light gray points belong to no tree, other colors are chosen using a graph coloring heuristic to show the separation between trees. On the right, the same point cloud is rendered with silhouettes and occlusion using our technique. The enhanced rendering clarifies the three-dimensional structure of the point cloud, making it more apparent where the segmentation works nearly perfectly, where it misses some trees, and where it divides a few trees into two or more segments.

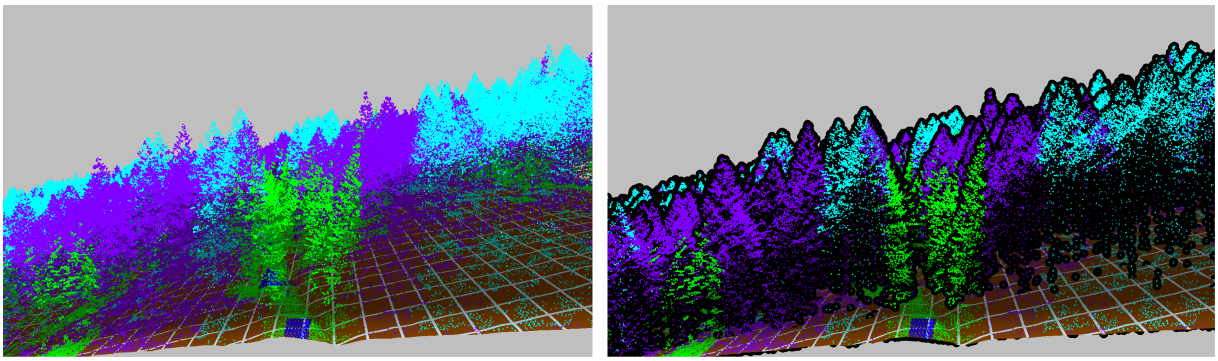

Figure 5: Our visualization makes it easier to validate and interpret the results of an unsupervised clustering algorithm. For example, the outlining (especially the lack thereof on points too close to the ground for the silhouette to appear) makes it clear that most clusters have points scattered in different regions of the canopy while the blue cluster is made up of only ground points. Trees in the teal cluster have most of their points concentrated in the dense higher canopies while trees in the green cluster have points scattered in the lower areas. The grid texture on the ground improves the perception of relative depth.

gorithm. Their answers were compared to ground-truth collected by one of us (PT). Each test was performed several times using either a simple point cloud or our outlining technique. We did not use color maps, lighting, or fog in an effort to minimize variables, but we did allow interactive motion so that some depth perceptions was always possible.

The average scores for both tests were good as shown in Table 1. Identification of riparian plots was improved with the silhouette rendering, while success in the more difficult spurious tree identification was unchanged. There was also a decrease in time spent on both tasks (Table 2). Feedback from a short user survey was overwhelmingly positive, reporting that participants found the silhouette visualizations clearer and easier to work with.

	Without Silhouettes		With Silhouettes	
Task	Correct	Incorrect	Correct	Incorrect
Riparian	53%	47%	65%	35%
Peaks	89%	11%	87%	13%

Table 1: Effect of silhouettes on task accuracy.

Task	Without Silhouettes	With Silhouettes
Riparian	23.6 *sec*	22.8 *sec*
Peaks	44.9 *sec*	36.1 *sec*

Table 2: Effect of silhouettes on task duration.

References

[BWK02] BOTSCH M., WIRATANAYA A., KOBBELT L.: Efficient high quality rendering of point sampled geometry. In *Proceedings of the 13th Eurographics workshop on Rendering* (2002), Eurographics Association, pp. 53–64. 2

[EBRI09] EVERTS M., BEKKER H., ROERDINK J. B. T. M., ISENBERG T.: Depth-dependent halos: Illustrative rendering of dense line data. *Visualization and Computer Graphics, IEEE Transactions on 15*, 6 (Nov 2009), 1299–1306. doi:10.1109/TVCG.2009.138. 2

[ERC01] EVERITT C., REGE A., CEBENOYAN C.: Hardware shadow mapping. White paper, 2001. 2

[HHL*08] HYYPPÄ J., HYYPPÄ H., LECKIE D., GOUGEON F., YU X.: Review of methods of small-footprint airborne laser scanning for extracting forest inventory data in boreal forests. *International Journal of Remote Sensing* (2008), 37–41. 1

[HI99] HYYPPÄ J., INKINEN M.: Detecting and estimating attributes for single trees using laser scanner. *The photogrammetric journal of Finland 16*, 2 (1999), 27–42. 3

[IS07] ISENBERG M., SHEWCHUK J.: Lastools: converting, viewing, and compressing lidar data in las format. *avaliable at: http://www. cs. unc. edu/ isenburg/lastools* (2007). 2

[KHW06] KOCH B., HEYDER U., WEINACKER H.: Detection of Individual Tree Crowns in Airborne Lidar Data. *Photogrammetric Engineering & Remote Sensing 72*, 4 (2006), 357–363. 3

[KLL*07] KWAK D., LEE W., LEE J., BIGING G., GONG P.: Detection of individual trees and estimation of tree height using lidar data. *Journal of Forest Research 12*, 6 (2007), 425–434. 3

[Kre13] KREYLOS O.: LiDAR Viewer software. http://http://idav.ucdavis.edu/~okreylos/ResDev/LiDAR/, 2013. 2

[LHN13] LI J., HU B., NOLAND T. L.: Classification of tree species based on structural features derived from high density LiDAR data. *Agricultural and Forest Meteorology 171-172* (Apr. 2013), 104–114. URL: http://linkinghub.elsevier.com/retrieve/pii/S0168192312003498, doi:10.1016/j.agrformet.2012.11.012. 3

[McG09] MCGAUGHEY R.: Fusion/ldv: Software for lidar data analysis and visualization. *US Department of Agriculture, Forest Service, Pacific Northwest Research Station: Seattle, WA, USA* (2009), 123. 2

[Nat12] NATIONAL SCIENCE FOUNDATION: The national ecological observatory network. http://www.neoninc.org/, 2012. 1

[PGK02] PAULY M., GROSS M., KOBBELT L. P.: Efficient simplification of point-sampled surfaces. In *Visualization, 2002. VIS 2002. IEEE* (2002), IEEE, pp. 163–170. 2

[PKG03] PAULY M., KEISER R., GROSS M.: Multi-scale feature extraction on point-sampled surfaces. In *Computer Graphics Forum* (2003), vol. 22, pp. 281–289. 2

[PW04] POPESCU S. C., WYNNE R. H.: Seeing the Trees in the Forest : Using Lidar and Multispectral Data Fusion with Local Filtering and Variable Window Size for Estimating Tree Height. *Photogrammetric Engineering & Remote Sensing 24061*, 0324 (2004). 3

[RL00] RUSINKIEWICZ S., LEVOY M.: Qsplat: A multiresolution point rendering system for large meshes. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (2000), ACM Press/Addison-Wesley Publishing Co., pp. 343–352. 2

[RPZ02] REN L., PFISTER H., ZWICKER M.: Object space ewa surface splatting: A hardware accelerated approach to high quality point rendering. In *Computer Graphics Forum* (2002), vol. 21, pp. 461–470. 2

[RSC87] REEVES W. T., SALESIN D. H., COOK R. L.: Rendering antialiased shadows with depth maps. In *ACM SIGGRAPH Computer Graphics* (1987), vol. 21, ACM, pp. 283–291. 2

[SHH*09] SHAFII S., HAMANN B., HUTCHINSON R., KREYLOS O., VIERS J.: Tree detection and delineation of the cosumnes river preserve. In *Proceedings of the 10th International Conference on GeoComputation* (University of New South Wales, Sydney, Australia, 2009). 3

[SHTA13] SHAH Z., HE S., TITTMANN P., AMENTA N.: Analysis of airborne laser scanning data with regional shape descriptors. *SilviLaser* (2013), 248–255. 3

[Sum11] SUMERLING G.: Lidar analysis in ArcGIS 10 for forestry applications. *ESRI (Hrsg.), An Esri® White Paper. Redlands, CA* (2011), 92373–8100. 2

[Ter13] TERRASOLID LTD.: Terrascan – software for lidar data processing and 3d vector data creation. http://www.terrasolid.com/products/terrascanpage.html, 2013. 2

[Tro09] TROPICAL ECOLOGICAL ASSESSMENT AND MONITORING NETWORK: Lidar data. http://www.teamnetwork.org/data/lidar, 2009. 3

[TSHH11] TITTMANN P., SHAFII S., HARTSOUGH B., HAMMAN B.: Tree detection, delineation, and measurement from lidar point clouds using ransac. In *Proceedings of Eleventh International Conference on LiDAR Applications for Assessing Forest Ecosystems (SilviLaser 2011)* (University of Tasmania, Hobart, Australia, 2011). 3

[Wil78] WILLIAMS L.: Casting curved shadows on curved surfaces. In *ACM SIGGRAPH Computer Graphics* (1978), vol. 12, ACM, pp. 270–274. 2

[XC04] XU H., CHEN B.: Stylized rendering of 3d scanned real world environments. In *Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering* (2004), NPAR '04, ACM, pp. 25–34. URL: http://doi.acm.org/10.1145/987657.987662. 2

[XNYC04] XU H., NGUYEN M. X., YUAN X., CHEN B.: Interactive silhouette rendering for point-based models. In *Proceedings of the First Eurographics conference on Point-Based Graphics* (2004), PBG'04, pp. 13–18. URL: http://dx.doi.org/10.2312/SPBG/SPBG04/013-018. 2

[ZPVBG01] ZWICKER M., PFISTER H., VAN BAAR J., GROSS M.: Surface splatting. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM, pp. 371–378. 2

[ZS04] ZAKARIA N., SEIDEL H.-P.: Interactive stylized silhouette for point-sampled geometry. In *Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia* (New York, NY, USA, 2004), GRAPHITE '04, ACM, pp. 242–249. 2