A Linked Feature Space Approach to Exploring LIDAR Data

Lane Harrison, Thomas Butkiewicz, Xiaoyu Wang, William Ribarsky, Remco Chang
University of North Carolina at Charlotte

ABSTRACT

A typical approach to exploring Light Detection and Ranging (LIDAR) datasets is to extract features using pre-defined segmentation algorithms. However, this approach only provides a limited set of features that users can investigate. To expand and represent the rich information inside the LIDAR data, we introduce a linked feature space concept that allows users to make regular, conjunctive, and disjunctive discoveries in non-uniform LIDAR data by interacting with multidimensional transfer functions. We achieve this by providing interactions for creating multiple scatter-plots of varying axes, establishing chains of plots based on selection domains, linking plots using logical operators, and viewing selected brushing results in both a 3D view and selected scatter-plots. Our highly interactive approach to visualizing LIDAR feature spaces facilitates the users' ability to explore, identify, and understand data features in a novel way. Our approach for exploring LIDAR data can directly lead to better understanding of historical LIDAR datasets, and increase the turnaround time and quality of results from time-critical LIDAR collections after urban disasters or on the battlefield.

Keywords: LIDAR, visualization, interaction, feature extraction, scatter-plots

1. INTRODUCTION

Existing LIDAR feature extraction approaches typically isolate features by applying pre-defined segmentation algorithms. Using this approach, users are limited to a small sandbox of feature spaces with which they can interact. Typically, users can only interact with terrain information that is embedded in the data, or they can reconstruct a 3D model based on the LIDAR grids. Although these features are practically important, after thorough study of LIDAR data, we found these typical approaches only cover a limited set of features embedded in the LIDAR data. Therefore, in this paper, we argue that LIDAR data actually contains more information that could of great help in understanding urban disasters or on the battlefield.

To help users discover the embedded rich information inside LIDAR datasets, we introduce our visualization system that depicts LIDAR data in multiple feature spaces, each of which represents a unique aspect of the LIDAR data (i.e. the certain geometry of a terrain). Our approach significantly expands the number and richness of interaction possibilities by allowing users to specify new feature space domains with varying axes. Segmentation algorithms may still be used to create various initial feature spaces. Our approach facilitates not only exploratory analysis, but also predictive analysis. In addition, users can decide beforehand which feature space path (which is essentially a visual multidimensional transfer function) to follow and verify their results in the 3D view.

By interactively linking feature spaces with Boolean logic operators, users are able to make conjunctive and disjunctive queries. Additionally, selections made in given plots can be used as new domains for derived plots. Redrawing a plot based on a new domain can uncover patterns that would be hidden by keeping a consistent domain.

By applying our methods to historical datasets, we can use the ability to identify once elusive features to extract additional knowledge which can then be used to strengthen predictive models based on these historical datasets. Due to the acceleration of feature discovery over other time-intensive techniques, one can produce timely and relevant results shortly after or even during LIDAR data collection. This is very attractive for time-critical applications, such as after natural or urban disasters and battlefield use where it is important to quickly assess the overall situation as well as track changes over time.

2. RELATED WORK

Previous approaches to LIDAR feature extraction focuses on computer-assisted and fully-automatic techniques. Computer-assisted methods typically use a small set of scatter-plots that allow users to brush subsets of interest[1]. Fully automatic approaches attempt to extract features of interest by using image segmentation algorithms[2]. Both of these approaches play a significant role in assisting user to understand features stored in the LIDAR data, however, either approaches only represents the features on a limited aspect. In contrast, our approach combines these two approaches to gather and presents user more complete features. In our approach, the system enables not only to start analysis with automated techniques but also allows users to interactively explore data feature spaces while simultaneously creating multidimensional transfer functions.

Research in volume rendering indicates that multidimensional transfer functions are essential for selecting features of interest in volumetric data[3]. Because it is easier for users to interact in 2D, these transfer functions are usually defined in a 2D feature space[4]. Thus the majority of our key interactions are done within the scatter-plot views. We define several novel interaction techniques for this view.

Zhou uses an edge-detection algorithm to detect building boundaries in a re-sampled grid of LIDAR data[5]. Our initial algorithm is related in that it computes gradients to detect building boundaries and other features of interest, except that we achieve a similar effect using raw unstructured LIDAR data, which does not sacrifice accuracy by requiring interpolation between sample points.

Recent developments in parallel coordinates visualization have shown the importance of conjunctive discoveries in real-world applications with several case studies[6,7]. Piringer introduces a system that links three 2D scatter-plots to a 3D scatter-plot[8]. Tweedie links histograms via union and negation operators to new histograms to facilitate conjunctive discoveries[9]

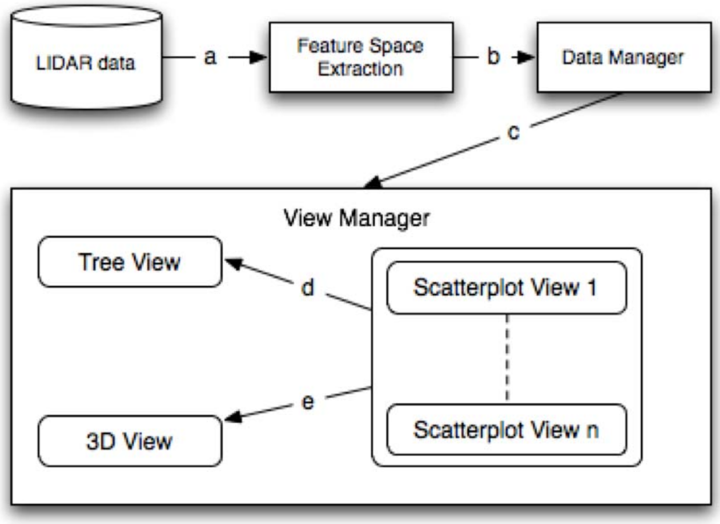


Fig. 1 Diagram showing basic system design. a) The LIDAR data is fed into a feature space extraction algorithm. b) Then this algorithm outputs the computed metrics into a data manager. c) The data manager is used by the view manager to control the data flow in the views. d) The creation of new scatter-plot views is reflected in the tree view. e) The 3D view reflects the transfer function specified by the user in the scatter-plot views.

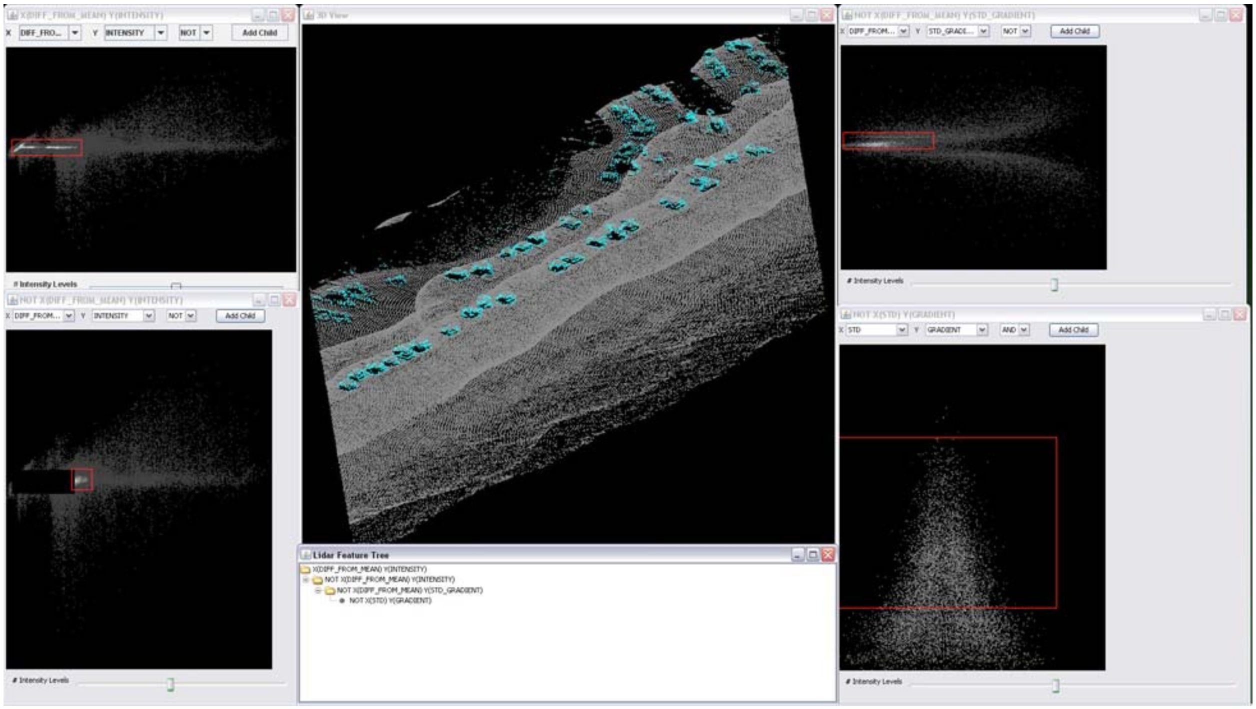
3. SYSTEM DESIGN AND IMPLEMENTATION

3.1 Data

It is important to know how our data is structured before we can apply automated and computer aided approaches to help depict the embedded features. In our current system, we focus on extracting and understanding features in 2.5 dimensional, unstructured Light Detection and Ranging (LIDAR) data. More specifically, for each (x,y) coordinate, there is an associated height (or intensity) value. Such LIDAR data is obtained by flying aircraft back and forth across a terrain, using laser-based range finders to collect sample points on that terrain. Due to the way in which the data is collected, in its original form it does not have a regular grid structure. In other words, each consecutive (x,y) pair is not evenly spaced. Instead there are areas of varied density (including overlapping scans), as well as areas without any coverage at all. The unique characteristics increase the difficulty in utilizing such data and therefore largely limits the features that can be extracted. Traditionally, this data is interpolated into a 2D raster to facilitate analysis. This process however, can remove accuracy and discards the extra sample points in areas of higher density[10]. Therefore, we use the unstructured data in our system to preserve as many features as possible.

3.2 Feature Space Acquisition

To extract meaningful features from the above datasets, our system focuses on applying computational algorithms to precede the initial extraction of our feature space. Specifically, our system computes the nearest neighbors for each LIDAR point and prepares each point for its gradient and standard deviation. Our system will also calculate more advanced metrics such as the standard deviation of the gradient. Any combination of these computations is referred to as a feature space, where different choices can produce various interesting results in the scatter-plot view. Users can develop an intuitive sense of the physical meanings of these metrics by making selections in the scatter-plot view, and seeing the results reflected in the 3D view.



X(DIFF_FROM_MEAN) Y(INTENSITY)
X DIFF_FRO... Y INTENSITY NOT

3D View

NOT X(DIFF_FROM_MEAN) Y(STD_GRADIENT)
X DIFF_FROM... Y STD_GRADE... NOT

NOT X(DIFF_FROM_MEAN) Y(INTENSITY)
X DIFF_FROM... Y INTENSITY NOT

NOT X(STD) Y(GRADIENT)
X STD Y GRADIENT AND

Lidar Feature Tree
X(DIFF_FROM_MEAN) Y(INTENSITY)
NOT X(DIFF_FROM_MEAN) Y(INTENSITY)
NOT X(DIFF_FROM_MEAN) Y(STD_GRADIENT)
NOT X(STD) Y(GRADIENT)

Fig. 2 Coastal LIDAR usage scenario. Selecting buildings in highly unstructured coastal LIDAR data. Note the bright areas in the scatter-plots that are made possible by the heat-map functionality.

3.3 Views

Our system utilizes a set of three types of views to depict features from difference perspectives. A 3D LIDAR rendering window is used to interactively represent the selected features. A 2D scatter-plot view represents different individual feature information. Throughout the interaction process, users create multiple scatter-plot views. All scatter-plot views are managed by a interactive tree view which is designed to facilitate user's exploration processes. Using our system, users can leverage the interactive capabilities of these views to meet their analysis goals. Since LIDAR are important data on multiple platforms, we also designed our system to be compatible with multiple platforms, as our implementation is based strictly on Java and OpenGL.

3.3.1 3D View

Throughout the user's exploration of the feature spaces, the 3D-view provides a way to see the results of the multidimensional transfer function in a geospatial context. All the LIDAR points are rendered based on its distribution in the dataset. User can easily zoom, pan and rotate the 3D view to examine LIDAR information from different angles and to help them focus on certain areas of the terrain.

This 3D view is coordinated with scatter plot views in a manner that the changes in scatter plot will view immediately reflect in the 3D view. The coordination is based on a one-to-many mantra, where system only keeps one 3D rendering visual state. This approach not only significantly reduces the use of computational resources, but more importantly, it will help the user to more efficiently explore the data set and reduce confusion during the exploration process. With the assistance of our interactive scatter-plot view, the 3D view is of great importance to help user visually depict the features and enable them to examine and compare individual LIDAR areas interactively.

3.3.2 Scatter-Plot View

This is the "linked" component of our approach. Users begin with a scatter-plot showing all LIDAR points and their choice of feature space (e.g. intensity vs. gradient). Then, users can change axes to any of the metrics mentioned in the data section, make two-dimensional selections on the plot itself, highlight their selection in the 3D-view, or create a new child scatter-plot based on their selections.

The creation of the child scatter-plots demonstrates the uniqueness of our approach, since it allows users to perform simple intuitive interactions, while actually making complex multi-dimensional transfer functions. Users may apply the Boolean operators AND or NOT when creating a child scatter-plot. Creating a child using the AND operator takes all of the LIDAR points within the two-dimensional selection range and creates a new scatter-plot view with these points as the domain. Semantically, this operation is similar to zooming in to a feature of interest. In contrast, using the NOT operator will create a scatter-plot with all of the LIDAR points outside of the selection range. This is useful for filtering out unwanted data parts. Multiple iterations of creating children using NOT and AND operators while switching axes allows users to hone in on areas of interest. However, what if a user wants to go back several steps? To solve problems like this, we allow users to hide unneeded scatter-plot views and store scatter-plot view interactions in a tree data structure.

Each child creation operation can be thought of as adding a child to a tree. For example, the root will contain all LIDAR points, and the first child could contain LIDAR points within a certain standard deviation and gradient range (or whatever range and axes the user selects). Users may then add another child to the root or a child to the first child and continue interacting with any scatter-plot. We store this interaction information so users can go back to previous nodes in the tree and create new branches of exploration.

Since scatter-plots tend to become over-plotted, meaning that several points are drawn on top of one another, we enhance the traditional scatter-plot view by adding a heat-map view. Such a view will show brighter colors for over-written points, which will help users to identify this issue and could even suggest places to look for more features. This enhancement is illustrated in the scatter-plots in Figure 2.

Two main issues are faced when translating a scatter-plot view into a density-based view. First, should the intensity in each cell of the heat-map be continuous or discretized? Second, how many pixels should there be per cell? To address the latter issue, we simply set one cell to each pixel. This way the heat-map looks more similar to the original scatter-plot. The continuous or discretized issue is more complex. The discretized approach can show different features in the plot depending on the cutoff points for each brightness level. In contrast, the continuous approach will only show one set of features per plot. Because of this, we chose to use the discretized method with a variable number of brightness levels that can be set by the user.

3.3.3 Tree View

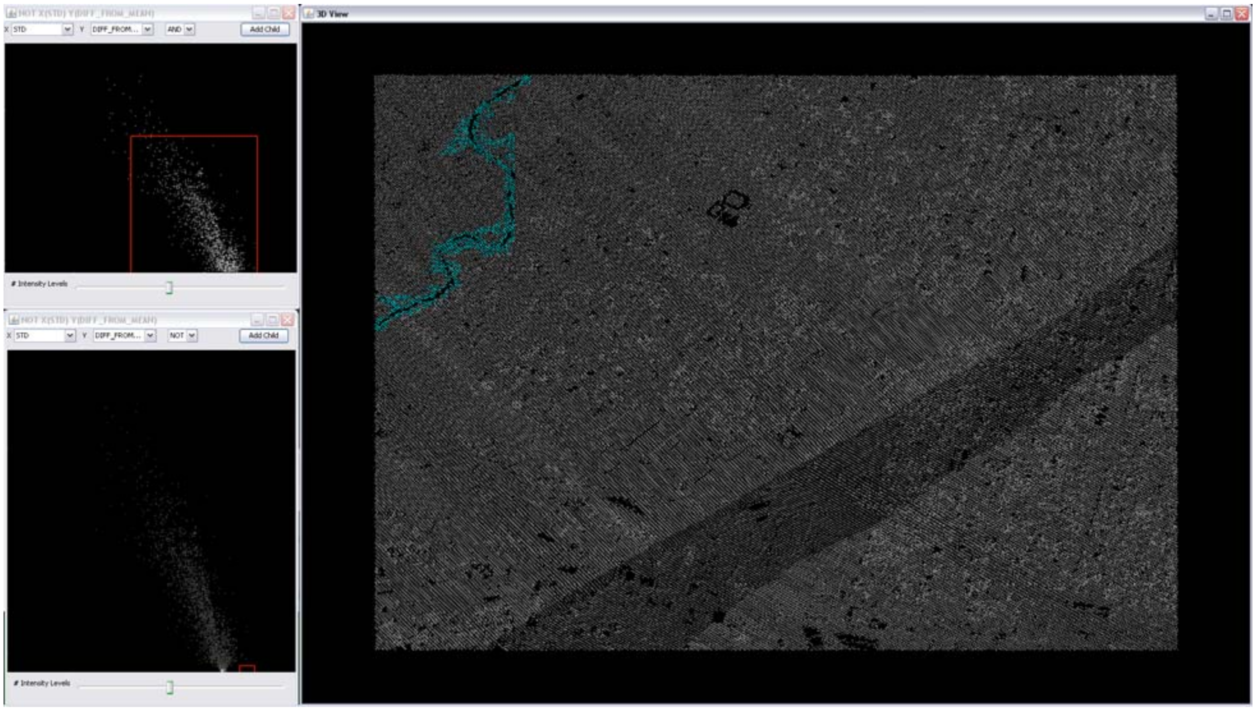
Since the scatter-plot child-creation process naturally takes the form of a tree, we provide a tree view similar to ones found in file-browsers to allow quick access to the child-creation interactions thus far. Basic expand and collapse operations are available, and double-clicking on any given node will show the corresponding scatter-plot view window. Also, should users have a large tree, they may want to show the scatter-plot view which they are currently using in the tree so they can see their previous steps. Users could do this by simply selecting a scatter-plot window, which will be highlighted in the tree view.

3.4 Scalability

We initially expected the performance of the system to slow dramatically as the tree size increased. However, by using binary arrays to track the active points in each subplot, we were able to increase performance even with deeper trees. In fact, we were able to render one-hundred scatter-plots at interactive rates on a basic laptop. Another potential bottleneck is the calculation of the nearest neighbors for each point. We have only tested the nearest-neighbor algorithm on smaller (2-3MB) files. However, this algorithm and the metrics calculations can also be pre-processed, with the resulting values loaded at runtime.

3.5 Extendability

Since the key components of our approach are linked scatter-plots, any calculation that results in one value per LIDAR point can be used with our system. The linked approach can also be used with other types of data, such as 3D volume data.



NOT X(STD) Y(DIFF_FROM_MEAN)

X STD Y DIFF_FROM... AND

NOT X(STD) Y(DIFF_FROM_MEAN)

X STD Y DIFF_FROM... NOT

Fig. 3 Analysis usage scenario: The user wishes to identify buildings that lie in the flood plane of the river. This is achieved by selecting low height and gradient values.

4. USE CASE

We will use a concrete example to better explain the queries that users can make with our system. We use an image-processing algorithm modified to work with unstructured LIDAR data as our starting point. This algorithm effectively brings out the features (walls, trees, etc.) in the data. Options are given for calculating values at each point based on the average gradient and maximum gradient of a given number of its nearest neighbors. This serves as a starting point for exploration. However, other algorithms that effectively segment unstructured data can be used, as well. In our example, the 3D view of the LIDAR data is linked to an initial 2D scatter-plot with height on the X axis and gradient on the Y axis.

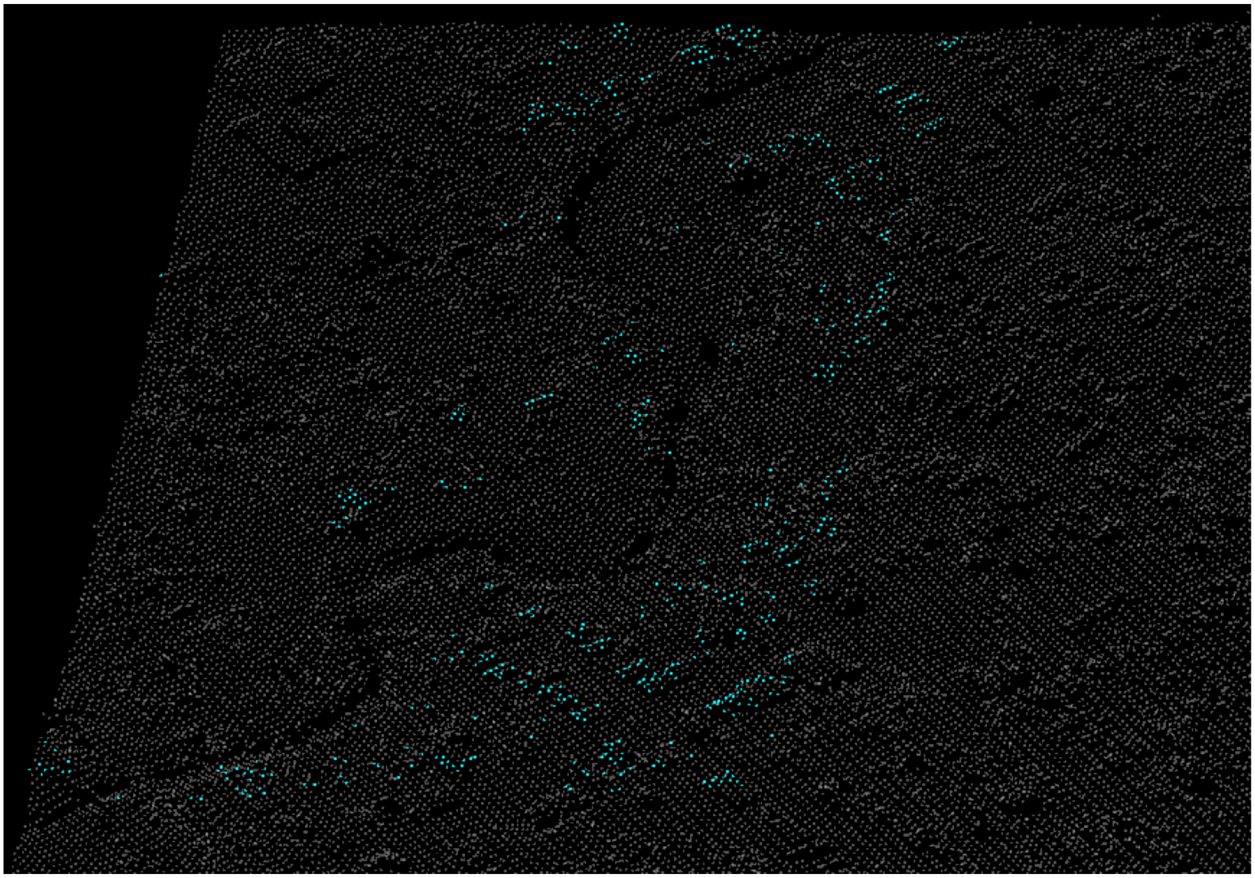


Fig. 4 Analysis usage scenario: By further interaction with the scatter-plots in Figure 3, the buildings of interest can be identified.

Users begin their analysis either by creating a new plot on the same level as the initial plot with different axes or by making a selection in the initial plot and using this selected domain to create a new plot. Users make selections by clicking and dragging to create multiple brushing boxes. All points contained in these boxes are shaded with a user-specified color. The latter option recalculates the new plot based on the new domain. This is key, as recalculating the scatter-plot for a smaller domain can reveal features in the data that would otherwise remain hidden. This “subplot” is added to a lower level in the plot tree. Users could also tweak high-level scatter-plots and see the changes reflected in the plots derived from it.

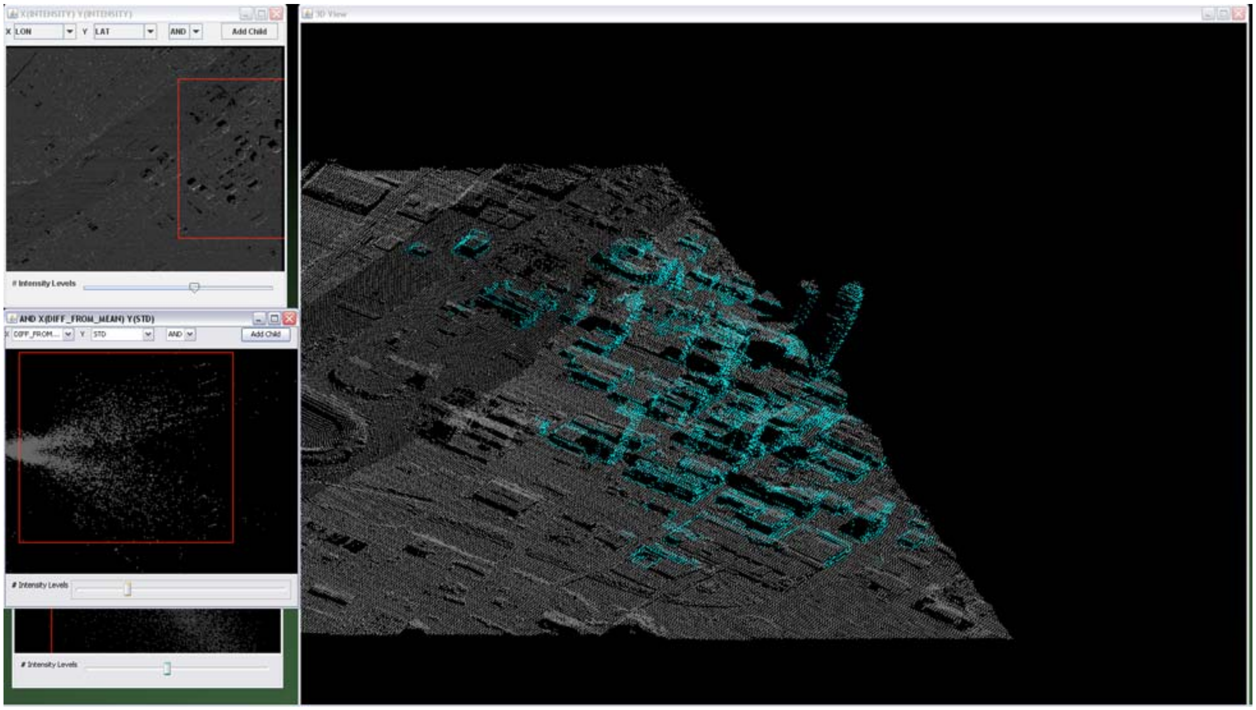


Fig. 5 Region-of-interest example. By plotting the latitude and longitude data in the scatter-plot view, users can select sub-regions in a given dataset to explore.

The 2D scatter-plots in the tree can be selected and linked together via logic operators. The process of linking and selecting plots creates a multidimensional transfer function. This also allows users to make conjunctive and disjunctive queries on the data (assuming that users know the meaning of their selections) and to get instant feedback in the 3D view. An example of this reasoning is as follows: "If the height vs. gradient falls in this range AND the variance vs. height falls within this sub-range, draw points as blue."

However, users may also explore the dataset without knowing exactly what they are selecting. For example, scatter-plots sometimes show clusters of related data items. Users may select these and color them in the 3D view or derive a new scatter-plot with this selected range as the new domain to see if any sub-clusters can be found.

Users can make disjunctive discoveries by using the NOT operator. This operator provides advantages in both semantics and interaction. The semantical advantage is that users are empowered to think and organize their analysis in natural terms. For instance, users may want to select a given height and gradient range and draw a new plot based on this range, but they also want to filter out a given gradient and variance range in the resulting plot. We expect that users will think of this query as "A but NOT B" where A and B are the ranges described, respectively. The interactive advantage of using the NOT operator comes in the form of speed and efficiency of making exclusionary selections. While these types of selections can be achieved by using multiple selection boxes to select the range outside of B, using a NOT operator necessitates only a single selection to achieve the desired result.

Once a user has achieved a desirable result, they could export either a path in the chain (a multidimensional transfer function) or the entire chain for future refinement.

4.1 Sample Scenarios

Consider a scenario where an analyst wishes to find the potential flood plane of a river and the buildings which lie in this flood plane. First, the analyst would set the axes of the starting scatter-plot to intensity (height) and gradient values, and select the lower end of this range, since the river would be low and flat in regards to the surrounding landscape. Next, the analyst would increase the range of their intensity selection, so that the area surrounding the river will be selected. This would be the flood plane. Finally, the analyst would use the standard deviation of gradient and difference from mean values to find the buildings in this area. This process is illustrated in Figures 3 and 4.

Realistically, it will be necessary to explore several combinations of axes and selections before finding an ideal time to make a child scatter-plot. However, since all child plots are stored in the tree, users can “save” intermediate results by adding children and refer to them later via the tree view.

Other possible scenarios include coastal storm surge and general urban terrain analysis, illustrated in Figures 2 and 5 respectively. In coastal storm surge analysis, one may be interested in identifying areas of potential surge inundation due to natural changes in dune structures. Urban analysts may encounter various situations, such as assessing a situation after a natural disaster via LIDAR data, which can be gathered quickly when needed.

5. CONCLUSION

Our approach is unique in that it is interaction-focused and allows users to test their intuition in a timely and flexible manner. These capabilities are essential in battlefield or disaster scenarios in which an analyst can quickly process incoming LIDAR data. By enhancing traditional scatter-plots with a heat-map view, users are able to identify more interesting features which can be explored further by creating a child scatter-plot and continuing interaction. By interacting with a 3D, tree, and scatter-plot views, users can easily drill down to features in the data and keep track of their results. Also, while interacting with these spaces, users are creating transfer functions that can be applied to the 3D-view. We demonstrate the effectiveness of our approach by providing several use cases, one of which was a scenario in which an analyst could determine which buildings lie within a river’s flood plane. For future work, we plan to apply saved interactions from a given dataset to a similar dataset, to explore the transferability of such transfer functions.

REFERENCES

[1] Butkiewicz, T. Chang, R. Wartell, Z. Ribarsky, W. “Visual Analysis and Semantic Exploration of Urban LIDAR Change Detection” EuroVis 2008 / Computer Graphics Forum, vol. 27, num. 3, 2008.

[2] Grun, A. Kubler, O. and Agouris, P. “Use of DTMs/DSMs and Orthoimages to Support Building Extraction.“ Automatic Extraction of Man-Made Objects from Aerial and Space Images, Birkhauser, Basel, pp. 199–210. . Baltsavias, E., Mason, S. and Stallmann, D., 1995.

[3] Kindlmann, G. and Durkin, J.W. “Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering”, Proc. IEEE Symp. Volume Visualization, pp. 79-86, 1998.

[4] Kniss, J. Kindlmann, G. and Hansen, C. “Interactive Volume Rendering Using Multi-Dimensional Transfer Functions and Direct Manipulation Widgets.” Proc. IEEE Visualization, pp. 255-262 and 562, 2001.

[5] Zhou, G. Song, C. Simmers, J. and Cheng, P. “Urban 3D GIS From LIDAR and digital aerial images. “ Computers & Geosciences, vol. 30, pp. 345-353, 2004.

[6] Akiba H. and Ma, K. L. “A Tri-Space Visualization Interface for Analyzing Time-Varying Multivariate Volume Data” 2007 Proceedings Eurographics/IEEE VGTC Symposium on Visualization. p 115.

[7] Steed, C. A. Swan II, J. E. Jankun-Kelly, T.J. and Fitzpatrick, P.J. “Guided Analysis of Hurricane Trends Using Statistical Processes Integrated with Interactive Parallel Coordinates.” In Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST 2009), October 2009.

[8] Piringer, H. Kosara, R. and Hauser, H. “Interactive focus+context visualization with linked 2D/3D scatter-plots.” In Proc. of the Intl. Conference on Coordinated & Multiple Views in Exploratory Visualization (CMV 2004), pages 49–60, 2004.

[9] Tweedie, L. Spence, B. Williams, D. Bhogal, R. “The Attribute Explorer” CHI '94: Conference companion on Human factors in computing systems (1994), pp. 435-436.

[10] Vu T. T., Matsuoka M., Yamazaki F.: “Lidar-based change detection of buildings in dense urban areas.” In Geoscience and Remote Sensing Symposium, 2004. IGARSS ’04. Proceedings. 2004 IEEE International (Sept 2004), vol. 5, pp. 3413–3416.