Spatially Important Point Identification: A New Technique for Detail-Preserving Reduced-Complexity Representation of 3D Point Clouds

Rohit Sant[1], Ninad Kulkarni[1], Kratarth Goel[2], Salil Kapur[2] and Ainesh Bakshi[3]
[1]Department of EEE&I, BITS-Pilani K.K. Birla Goa Campus, Goa, India
[2]Department of CS/IS, BITS-Pilani K.K. Birla Goa Campus, Goa, India
[3]Department of Chemical Engineering, BITS-Pilani K.K. Birla Goa Campus, Goa, India

Abstract - *This paper describes a technique for reducing the inherent complexity of range data without discarding any essential information. Our technique implements this by searching for 'important' points in the range data and discarding intervening points, all of which may be regenerated to a good approximation by linear interpolation. The implementation uses a metric based on the 3D geometry of the scene to assign to each point an 'importance' value. We define Spatially Important Points, which are got by comparing this importance value with a customizable template and with importance values of its neighbours. The algorithm has been tested on various datasets and has been found to give, on an average, a 78% reduction in complexity while retaining almost all points of significance, as shown by reconstructing the dataset. Results have been tabulated at the end of the paper.*

Keywords: 3D, range data, terrain, compression, importance

1 Introduction

Optimal representation of the environment in terms of terrain maps has been a long-studied topic in mobile robotics. A long-standing debate exists on the benefits of either mapping technique, and the exclusive choice of either invariably results in a compromise [1], [2]. It is with this background and the subsequent motivation that we present our technique for data representation – we recognize the powerful nature of the geometric 3D model of the scene, and we work towards reducing its complexity while preserving that power. Since conventional downsampling causes an equal loss of useful information as that of redundancy, a specialized technique for reduction of complexity is needed.

We begin by detecting points essential to the representation of the terrain. This is done by comparing 'importance' values (defined in Section 3.2) of each point to those in its neighborhood, and also to a template. We define Spatially Important Points (SIPs) to be points of appreciable change in a terrain. Setting a threshold gives an output of SIPs for the scene. Our technique may be viewed as an importance based sorting algorithm; setting higher thresholds removes points in ascending order of importance. The scene may thus be represented in varying levels of detail based on the chosen value of the threshold, but the algorithm ensures that features absolutely essential to the representation of the terrain are retained for almost all realistic thresholds.

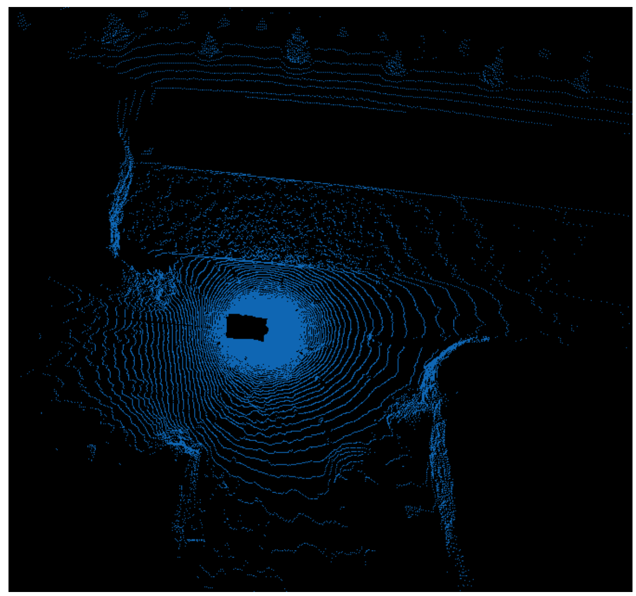

Figure 1. A sample point cloud displaying an outdoor scene. This point cloud, though it gives an accurate terrain representation, contains more than 100,000 points, making it unwieldy for easy storage and operation.

Put simply, the Spatially Important Point Representation (SIPR) of a scene results in a minimal representation with detail preservation. Its percentage retention of informative points (points which add to the information content of the scene) is far greater than that of redundant points. Detail-preserving, vastly compressed and easily reconstructable datasets result, which can be directly used in robotic applications like navigation, planning, mapping, locomotion.

The organization of the paper is as follows: Section (2) describes related work in this field, Section (3) describes, in detail, the working of our algorithm while the implementation is discussed in Section (4). Results are tabulated in Section (5) and Section (6) concludes the paper.

Note that in the +rest of this paper, the terms 'terrain', 'scene' and 'range data' are used interchangeably, with the understanding that they all imply a 3D range dataset..

2 Related Work

Reduced-complexity representation of 3D data is an implicit requirement for almost all research concerning 3D range data. Early work, for example that by Martin et al. [15] uses a median value to represent each 3D grid in an octree. Though this reduces complexity, it results in indiscriminate loss of detail. A variation on the same theme by Lee et al. [16] incorporates non-uniform grids, but the results still suffer due to the downsampling technique. A technique described in [14] uses multiple scans to identify redundancy, but this adds an unwanted restriction on the input and severely limits its usefulness in situations where multiple scans cannot be obtained. A notable implementation can be seen in [17], in which the authors approach this problem using curvature.

Since this is usually an interim step in a larger problem statement, most researchers do not bother with operating on raw data, preferring to use clustering [8], [9] usually with a k-nearest neighbors approach. In [10], a different technique involving plane detection to reduce complexity of range data has been described, but their requirement does not include the finer details in a point cloud.

Often geometric relations between points have been used in determining properties of the terrain. Labecki et al in [4] analyzed geometric configurations of a walking robot for terrain mapping. Schenker et al in [5] also use the geometric structure of scan-lines for terrain exploration. Most of the above research uses rudimentary metrics for actually detecting important points using range data, relying instead on powerful post-processing (Hough transforms [6]) and/or auxiliary peripherals for confirmation. A notable application of a similar technique is carried out by Khalifa et al [7], which uses a more refined scan-line based technique for CAD model acquisition from 3D range data. Our algorithm uses a more robust metric for determination of useful terrain points, and as a standalone tool gives a better output than current algorithms.

2.1 Our Contribution

Over the past 10-15 years, advances in high density point acquisition techniques have made datasets having the order 10^6 points commonplace in robotic navigation. This makes most of today's robotic applications like planning, mapping, and navigation computationally very expensive. Unlike [11], [15] and [16] which build upon more expensive constructs like meshes, grids, triangulations and /or expensive post-processing using ICP [14], we develop a completely new technique starting from basic geometry and build up on it. Our algorithm achieves better efficiency, due to its specialized nature. To the best of our knowledge, our algorithm for low-complexity range data representation does indeed provide a unique and useful solution for detail preserving reduction of point data – thus removing the final drawback of range data.

3 Algorithm Description

The structured nature of unprocessed range data can be effectively used to reduce the complexity of the representation without discarding essential information. We attempt to demonstrate the effectiveness of this approach through this algorithm.

3.1 Overview

Terrain data obtained from modern laser rangefinders follows a structure where points are scanned along a line of constant elevation. A Spatially Important Point (SIP), as defined by us, is a point where the terrain changes appreciably. A definition of this form has been made since all points which lie between two such consecutive constant-elevation SIPs can be replaced by a linear interpolation between the said SIPs, thus eliminating the need to store them explicitly. Spatially Important Point detection, as mentioned previously, is done by finding 'importance' values (defined in Section 3.2) of each point and then comparing with those of its neighbors. This stage involves minimal loss of actual detail. We therefore hypothesize that a representation of the terrain by its Spatially Important Points is a complete representation, from the point of view of robotic applications. The output of SIPR is still a point cloud consisting of raw points belonging to the original dataset, with no aberrations

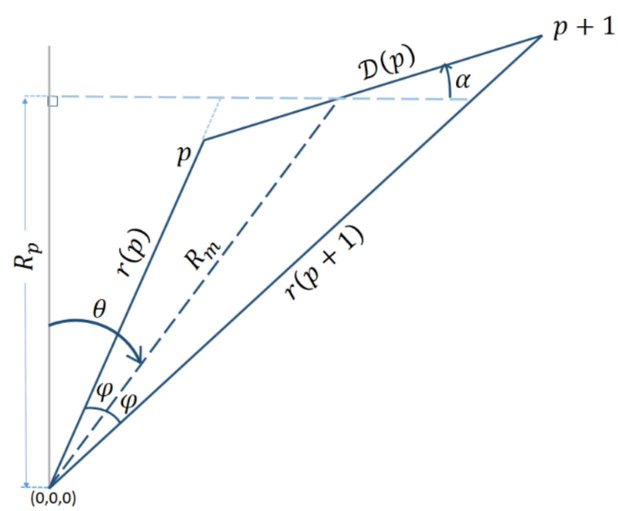

Figure 2. The schematic diagram for evaluating Spatially Important Points. A single segment from (p) to (p+1) is shown for purposes of clarity.

3.2 Importance Function

We define an importance function f for quantifying relevance of points. The requirement for this function is that it should provide a good indicator of the magnitude of change of the terrain from one point to the next.

Let $\mathbb{P}_i = \{p_{i1}, p_{i2}, \dots p_{in}\}$ be the i[th] constant-elevation scan line having n points. Then for each $\mathbb{P}_i$, points are treated pairwise, and the distances between adjacent points are stored in a set $\mathcal{D}_i$, where $\mathcal{D}_i(p_{ik})$ is the Euclidean distance between points p_{ik} and $p_{i(k-1)}$ (Figure 1). Further operations are done exclusively on $\mathcal{D}_i$. The function f is got by operating on $\mathcal{D}_i$ as follows:

$$f(p) = \frac{\mathcal{D}_i(p_{ik})}{g(r).h(\theta)} - \left|\overrightarrow{R_p}\right| \qquad (1)$$

Where (Figure 2),

$$\left|\overrightarrow{R_p}\right| = \left|\overrightarrow{R_m}\right| \cos\theta \qquad (2)$$

$$g(\vec{r}) = \left(1 + \frac{|\vec{r}(p_{i(k+1)})|}{|\vec{r}(p_{ik})|}\right) \qquad (3)$$

$$h(\theta) = \frac{\tan\varphi.\sec^2\theta}{1+\tan\varphi.\tan\theta} \qquad (4)$$

($\vec{r}(p_{ik})$ Is simply the radius vector of point p, and φ is half the constant angle subtended by each segment at the origin.)

3.2.1 Discussion

The defining characteristic of this function is that it identically equals zero for any consecutive point-pair on a flat terrain. Hence deviations of f from zero provide a good measure of the absolute change in the terrain. Comparisons with neighboring values of f is a good way of judging the relative change in the terrain.

The function $f(p_{ik})$ increases monotonically with decreasing α (Figure 1) and thus unambiguously gives a measure of deviation from local flatness.

The function $g(\vec{r})$ is a measure of the expected $\mathcal{D}_\mathrm{i}(\mathrm{p_{ik}})$ values for a flat terrain; the greater this function, the larger would be the expected value of $\mathcal{D}_\mathrm{i}(\mathrm{p_{ik}})$ for a flat terrain, since φ is constant everywhere. The ratio $\frac{\mathcal{D}_i(p_{ik})}{g(r)}$ would then describe the absolute deviation of a segment from flatness.

The function $h(\theta)$ arises out of the fact that for a flat terrain – our template for absolute comparison – $\mathcal{D}_i(p_{ik})$ and $g(\vec{r})$ values do not scale equally (φ is constant – a hardware constraint) with increasing θ.

3.3 Template

We choose a flat terrain as a basis for absolute comparison since it provides a practically desirable ideal in many cases.

For a hypothetical flat constant-elevation portion of the terrain at a given $\overrightarrow{R_p}$, each $\mathcal{D}_i(p_{ik})$ has a different value, since all of them subtend the same angle at the origin. However, the value of $f(p_{ik})$ for each $\mathcal{D}_i(p_{ik})$ is zero i.e. a constant. Thus the value of $\frac{\mathcal{D}_i(p_{ik})}{g(r).h(\theta)}$ for a flat terrain is equal to $\left|\overrightarrow{R_p}\right|$, and evaluating $f(p_{ik})$ merely measures the deviation of $\mathcal{D}_i(p_{ik})$ from a flat terrain.

An intricacy here is the choice of $\overrightarrow{R_p}$ – since the function f attains zero at an $\left|\overrightarrow{R_p}\right|$ equal to the perpendicular distance to a flat terrain constructed at the same place as each $\mathcal{D}_i(p_{ik})$, we have to choose $\overrightarrow{R_p}$ locally to ensure correct interpretation of the principle. We also assume the angle bisector to be invariant of the orientation of $\mathcal{D}_i(p_{ik})$, which is a reasonable assumption given the constant azimuth readings. As a result the angle bisector cosine is chosen as an appropriate $\overrightarrow{R_p}$.

It is worth mentioning that the algorithm can be used for very specialized detection purposes – any non-flat terrain can be compared against by merely calculating an f for that template and compare everything against this value.

3.4 Neighbor Comparison

Following this, we apply the threshold condition:

$$p_{ik} \in \mathbb{K} \;\; iff$$

$$\Big(f(p_{ik}) - f\big(p_{i(k-1)}\big) \in \big((-\infty, -\mathcal{T}) \cup (\mathcal{T}, \infty)\big)\Big)$$
$$or$$
$$\Big(f\big(p_{i(k+1)}\big) - f(p_{ik}) \in \big((-\infty, -\mathcal{T}) \cup (\mathcal{T}, \infty)\big)\Big) \qquad (5)$$

Where

$\mathbb{K}$ is the set of Spatially Important Points
f is the importance function
p_{ik} is the point index in the original structured dataset
$\mathcal{T}$ is a constant threshold set as a decision boundary

3.4.1 Analysis

For a neighbor comparison, we impose the Spatially Important Point condition, which measures a difference of the deviations from flatness of adjacent points, thus establishing a measure for observing the trend in the terrain. A point is chosen as a Spatially Important Point if its deviation is sufficiently different from that of its neighbor.

4 Implementation

We deal with an input terrain represented as a point cloud in 3D. For the purposes of this research, we used the Canadian Planetary Emulation Terrain 3D Mapping Dataset [12]. Data is acquired through a laser rangefinder (LRF) with a constant 0.36° azimuthal separation (half of this angle is later referred to as φ) between consecutive readings at constant elevation. This data is represented as co-ordinates on a spherical (Range, Elevation, Azimuth) system [12]. We take advantage of the fact that this input data is structured, enabling us to traverse the dataset in lower time complexity as compared to an unstructured set.

Since the algorithm is run for all sets of constant-elevation data, all regularity (redundancy) is correctly identified (albeit in different iterations) and removed.

5 Results

The technique was tried on various sets of range data available through the Canadian Planetary Emulation Terrain 3D Mapping Dataset [12], the results of which are discussed below. One scene – Scene A (Figure 3) has been analyzed in this section, and a few more results have been displayed in Figures 4-7. All images are screenshots of point clouds viewed

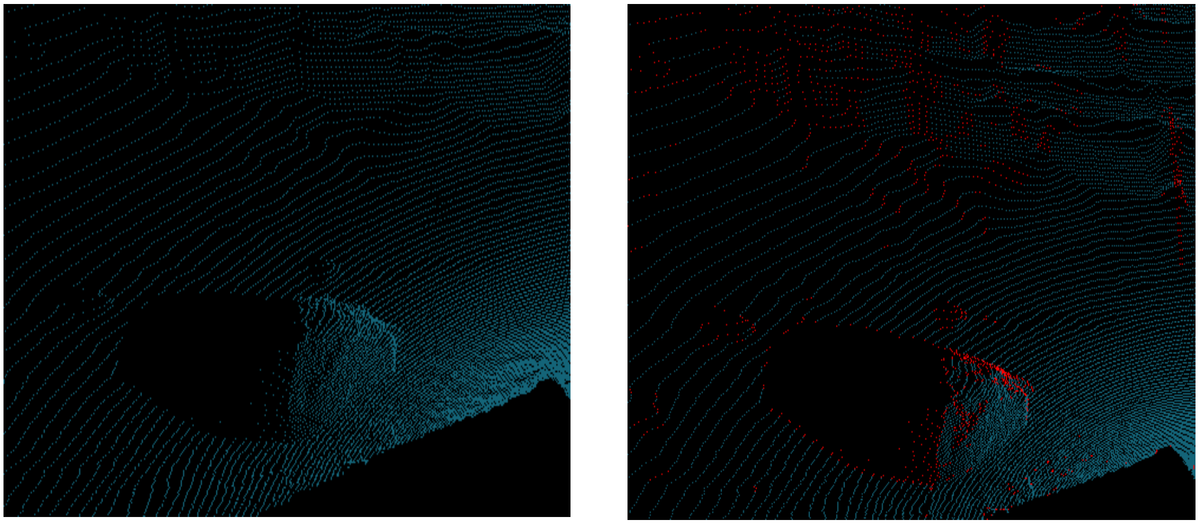

Figure 3. The figure on the left indicates a piece of terrain and the figure on the right shows its SIPR (in red) superimposed on the dataset.

in the PCD viewer (part of Point Cloud Library [13]) or Meshlab [3]. The readers are requested to consult the full-color (electronic) version for maximum clarity.

5.1 Analysis of Scene A

This scene (Figure 3) comprises of a dense flat portion in the middle of the scene adjacent to which a fairly large rock is seen. A wall made of large, discrete rocks is seen in the background. A comparative picture of a section of the original dataset and its SIPR is shown here.

The full point cloud contains 105998 points, while the SIPR is built up of just 8722 points, resulting in a size reduction of nearly 92%. The complete boundary of the centered large rock is maintained, and every single junction between the discrete rocks in the wall is well identified by the SIP algorithm. Most of the discarded points belonged originally to the flat central portion seen in the middle of Figure 3, thus removing the (easily re-constructible) regularity present in the image. The code executes in 322ms for the above scene comprising of 44 scan lines and a total of 105998 points.

For the purposes of reconstruction, as described in the section below, such aggressive compression leads to less usable terrain, and an average compression of 75% is chosen.

5.2 Reconstruction – A measure of performance

We verify our claim – of our reduced point set being a good representation of the entire terrain – by reconstructing the terrain using a simple linear interpolation between our points. Figure 4 shows one such comparative result.

An alpha-shapes surface fitting is also done, in Figures 5 and 6. Features like trees, houses and contours between the house and the trees have been well reconstructed. We observe that the interpolation is done along the same scan lines as the original dataset – since SIPI operates on these scan lines for removing points, any other surface fit on the reconstructed points (splines, triangulations) too would generate much the same surfaces as the original.

This completes the motivation of this paper – we have shown that our Spatially Important Points, keeping merely 30% of the original dataset, manage to closely approximate the original dataset through a simple interpolation. Comparisons of alignment with the original dataset were done using Iterative Closest Point (ICP) and results have been reported in Table 1.

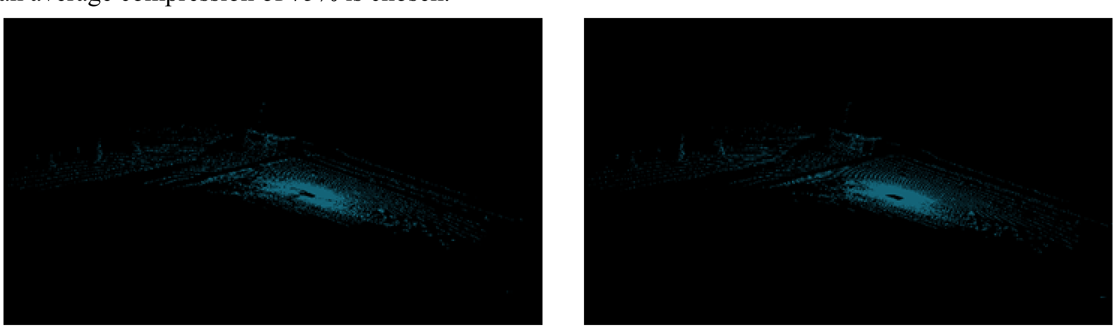

Figure 4. Reconstructed image of a point cloud on the left and the original image on the right

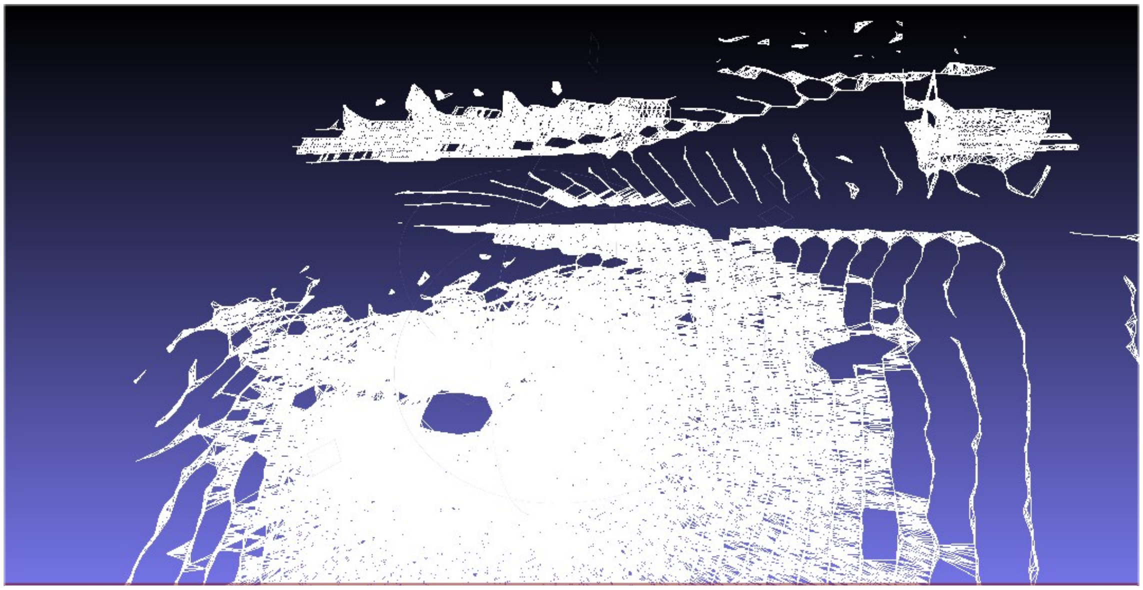

Figure 5. Alpha shapes for the original terrain

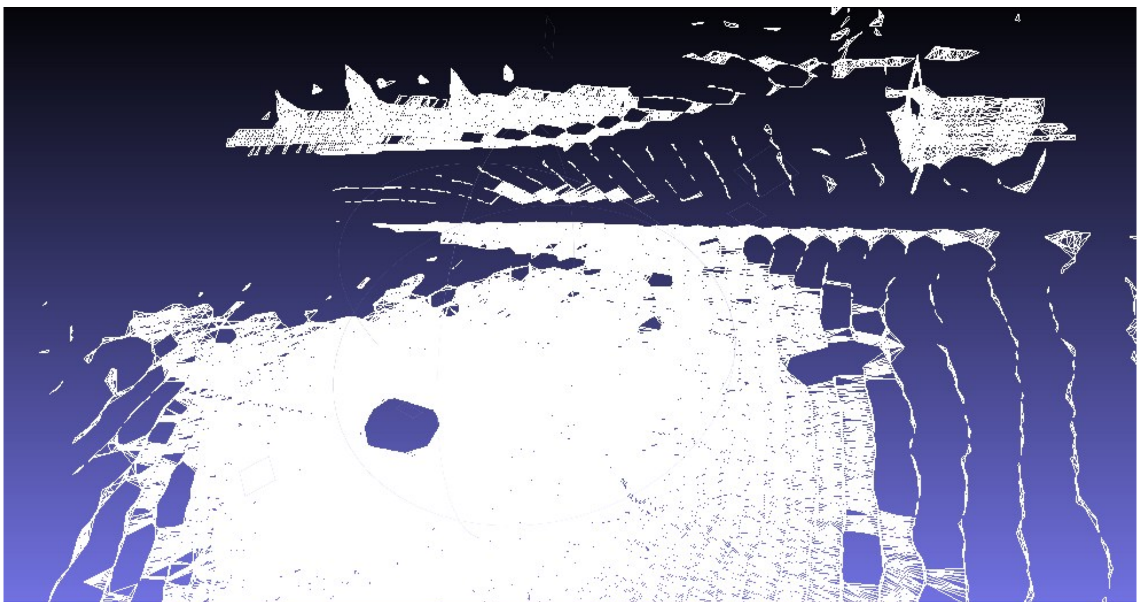

Figure 6. Alpha shape reconstruction for the terrain in Figure 4

5.3 Other Results

Figure 7 shows the variation of the alpha-shapes reconstruction with the number of retained data points. It is seen that at 27% retention, the reconstruction closely resembles the original terrain. Even at a 14% retention, the output is seen to be visually similar to the input.

These results were obtained on a machine with an Intel® Core i3-2310M processor (2.10 GHz) and 4 GB of RAM. The code

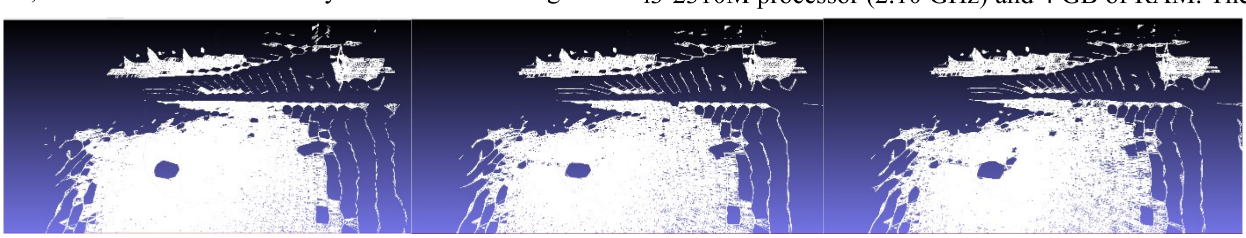

Figure 7. Alpha shape reconstruction for (L-R) (a)Original terrain (b) Reconstruction from 27% points and (c) Reconstruction from 14% points.

for executing this technique was written in C++ and compiled using version 4.4.3 of the GNU g++ compiler.

Table 1. Results of SIPI

Average data reduction (structured point clouds)	77.6%
Average runtime	314 ms
Average ICP Score of reconstructed terrain with original terrain	$1.162*10^{-3}$
Asymptotic Complexity	$O(n^2)$ (worst case)

6 Conclusion, Applications & Future Work

The Spatially Important Point Identification (SIPI) technique is a simple and fundamentally strong approach towards efficient representation of 3D data. It has been demonstrated that the Spatially Important Point Representation (SIPR) of a scene can successfully represent all defining characteristics present in the data using only, on an average, 22.4% of the complexity of the original data.

Some of the ideas highlighted in the paper are given here

1. Hypothesis of an efficient representation of range data which enhances detail while reducing redundancy
2. Adoption of a scan-lines based approach for identifying important points in a terrain
3. Definition and assignment of a unique ratio for gauging the importance of a point
4. Experiments done on terrain databases with favorable results as shown through reconstruction, with as much as 91.3% data reduction possible with no catastrophic loss of detail.

6.1 Future Work

We are implementing a dynamic-programming based planning algorithm on the basis of this data reduction technique. The current technique was developed with a view to integrating seamlessly with further robotic navigation techniques.

References

[1] Thrun, S., & Bucken, A. (1996). Integrating Grid-Based and Topological Maps for Mobile Robot Navigation. *The Thirteenth National Conference on Artificial Intelligence (AAAI'96)* (pp. 944-950). Portland: AAAI Press.

[2] Pfaff, P., Triebel, R., & Burgard, W. (2007). An Efficient Extension of Elevation Maps for Outdoor Terrain Mapping. *International Journal of Robotics Research*, 217-230.

[3] Cignoni, P., Corsini, M., Ranzuglia, G.: Meshlab: an open-source 3D mesh processing system. ERCIM News, 45–46 (2008)

[4] Łabęcki, P., Rosiński, D., & Skrzypczyński, P. (2011). Terrain Map Building for a Walking Robot Equipped with an Active 2D Range Sensor. *Journal of Automation, Mobile Robotics & Intelligent Systems*, 68-78.

[5] Schenker, P. S., & Sword, L. F. (1997). *Lightweight rovers for Mars science exploration and sample return.* Pasadena: California Institute of Technology.

[6] Borges, P., Zlot, R., Bosse, M., Nuske, S., & Tews, A. (2010). Vision-based Localization Using an Edge Map Extracted from 3D Laser Range Data. *International Conference on Robotics and Automation* (pp. 4902-4909). Anchorage,Alaska: IEEE.

[7] Khalifa, I., Moussa, M., & Kamel, M. (2003). Range image segmentation using local approximation of scan lines with application to CAD model acquisition. *Machine Vision and Applications*, 263-274.

[8] Zhao, Y., He, M., Zhao, H., Davoine, F., & Zha, H. (2012). Computing Object-based Saliency in Urban Scenes Using Laser Sensing. *International Conference on Robotics and Automation* (pp. 4436-4443). Saint Paul, Minnesota: IEEE.

[9] Zhao, H., Liu, Y., Zhu, X., Zhao, Y., & Zha, H. (2010). Scene Understanding in a Large Dynamic Environment through Laser-Based Sensing. *International Conference on Robotics and Automation, 2010* (pp. 127-133). Anchorage,Alaska: IEEE.

[10] Li, W., Wolberg, G., & Zokai, S. (2011). Lightweight 3D Modeling of Urban Buildings From Range Data. *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission* (pp. 124-131). Hangzhou: IEEE Computer Society.

[11] Moorthy, I., Millert, J. R., Hut, B., Berni, J. A., Zareo-Tejada, P. J., & Lit, Q. (2007). Extracting tree crown properties from ground-based scanning laser data. *IEEE International Geoscience and Remote Sensing Symposium* (pp. 2830-2832). Barcelona: IEEE.

[12] Tong, C., Gingras, D., Larose, K., Barfoot, T. D., & Dupuis, E. (2012). The Canadian Planetary Emulation Terrain 3D Mapping Dataset. *International Journal of Robotics Research (IJRR)*.

[13] Rusu, R. B., & Cousins, S. (2011). 3D is here: Point Cloud Library (PCL). *2011 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1-4). Shanghai: IEEE.

[14] Swadzba, A., Vollmer, A., Hanheide, M., & Wachsmuth, S. (2008). Reducing noise and redundancy in registered range data for planar surface extraction. *19th International Conference on Pattern Recognition, 2008* (pp. 1- 4). Tampa, FL: IEEE.

[15] R. R. Martin, I. A. Stroud and A. D. Marshall (1996). Data reduction for Reverse Engineering. RECCAD, Deliverable Document 1 COPERUNICUS project, No. 1068, Computer and Auto-mation Institute of Hungarian Academy of Science, Jan 1996

[16] K.H., Lee, H., Woo, & T., Suk (2001). Data Reduction Methods for Reverse Engineering. Int J Adv Manuf Technol, 735-743.Stanford University Computer Graphics Laboratory. (1994).

[17] Song, W., Cai, S., Yang, B., Cui, W., & Wang, Y. (2009). A Reduction Method of Three-Dimensional Point Cloud. 2nd International Conference on Biomedical Engineering and Informatics, 2009. (pp. 1-4). Tianjin: IEEE.