3D Point Cloud Registration for Localization using a Deep Neural Network Auto-Encoder

Gil Elbaz Tamar Avraham Anath Fischer
Technion - Israel Institute of Technology
gil.elbaz@alumni.technion.ac.il tammya@cs.technion.ac.il meranath@technion.ac.il

Abstract

We present an algorithm for registration between a large-scale point cloud and a close-proximity scanned point cloud, providing a localization solution that is fully independent of prior information about the initial positions of the two point cloud coordinate systems. The algorithm, denoted LORAX, selects super-points—local subsets of points—and describes the geometric structure of each with a low-dimensional descriptor. These descriptors are then used to infer potential matching regions for an efficient coarse registration process, followed by a fine-tuning stage. The set of super-points is selected by covering the point clouds with overlapping spheres, and then filtering out those of low-quality or nonsalient regions. The descriptors are computed using state-of-the-art unsupervised machine learning, utilizing the technology of deep neural network based auto-encoders.

This novel framework provides a strong alternative to the common practice of using manually designed key-point descriptors for coarse point cloud registration. Utilizing super-points instead of key-points allows the available geometrical data to be better exploited to find the correct transformation. Encoding local 3D geometric structures using a deep neural network auto-encoder instead of traditional descriptors continues the trend seen in other computer vision applications and indeed leads to superior results. The algorithm is tested on challenging point cloud registration datasets, and its advantages over previous approaches as well as its robustness to density changes, noise and missing data are shown.

1. Introduction

1.1. Overview

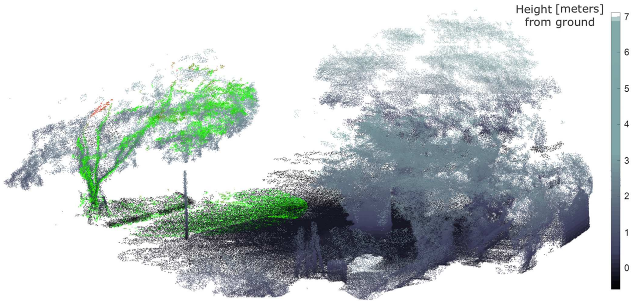


Figure 1: Registration between a close-proximity point cloud (colored) and a large-scale point cloud (grayscale)

Point clouds, similarly to images, capture semantic information describing the objects in the world around us. In contrast to image data, which holds a two-dimensional projection of the scene in a fixed grid, a point cloud is a set of unorganized three-dimensional points in a unified coordinate system, capturing 3D spatial information. Methods for point cloud data analysis have been developed throughout the past few decades [1], and the significance of research in this field is trending upwards due to advances in affordable high-quality 3D scanning technology [2], machine learning breakthroughs, and new interesting applications.

Point cloud registration is defined as finding the transformation between two separate point cloud coordinate systems. It is key for Simultaneous Localization and Mapping (SLAM) [3, 4], 3D reconstruction of scenes [5], and it has become central in vision-based autonomous driving [6]. Much progress has been achieved, yet there are still major challenges, such as registration of large-scale point clouds, with low scene overlap and without prior positional information.

Outdoor localization today relies heavily on GPS technology, accompanied by ground based augmentation systems to improve accuracy. This localization requires receiving signals from multiple satellites simultaneously with an accuracy that varies greatly with the number of satellites available, the weather, and physical obstructions blocking or altering the signal path. The technology is inaccurate and unreliable in areas around the world with little to no ground infrastructure [7].

In this paper we focus on a localization technique that relies on registering a large-scale point cloud and a small-scale point cloud scanned within a scene at different times. The registration is independent of proximity information between the clouds' initial coordinate systems. We define the two point clouds: a "global point cloud", made up of a large scanned outdoor scene with a coordinate system fixed to a real-world geographic coordinate system, and a "local point cloud", made up of a substantially smaller point cloud, captured online at an unknown location and orientation within the global point cloud scene. The transformation between the local and global point cloud is calculated using a machine learning analysis of their geometry. This serves as a high quality localization method that is completely independent of GPS for outdoor environments. See Fig. 1.

1.2. Related Work

Registration algorithms are divided into those dealing with coarse registration and those dealing with fine registration. Coarse registration algorithms make no prior proximity assumptions of the point cloud positions and aim for a coarse alignment, meaning a lenient loss function policy. Fine registration algorithms assume that the input point clouds are approximately aligned; they thus utilize the initial proximity between the points to fine-tune the alignment between the point cloud coordinate systems. Fine registration can be used when the two clouds are acquired consecutively with large overlaps between their scenes, or as a follow-up to a coarse registration procedure.

Numerous point cloud coarse registration methods have been developed [8], yet coarse registration remains an open challenge with much room for improvement. In the Fast Point Feature Histogram [9, 10] (FPFH) algorithm, a histogram based descriptor is calculated for each point within the point cloud, over multiple scales. Salient persistent histograms over multiple scale calculations are labeled as key-points, which are then matched to find the registration between the point clouds. Other descriptors for locating and describing key-points were suggested as well. See [11] for a survey. Some examples are 3D-SIFT [12] , NARF [13], and SHOT [11]. Many complex hand-coded features were proposed, with the same goal of being invariant to rotation and translation, and robust to noise.

In the field of 2D computer vision, a similar development period of hand-coded features has come to an abrupt end due to the breakthrough research in the field of deep learning [14]. Using the deep learning methods, much more advanced features (with complexity beyond human design) are calculated from within the data, advancing major fields in 2D computer vision such as detection, classification, segmentation, localization and registration [15]. These methods focus almost exclusively on 2D data. The unstructured, continuous and large point cloud datasets pose extreme problems not enabling a straightforward dimension adaption into 3D space. In order to utilize the 3D data in our method, the point cloud is densely sampled, and the 2.5 dimensional data of each local surface is captured and combined. The advanced tools of deep learning are applied to this data, in the form of unsupervised machine learning for high quality dimension reduction [16], as a critical stage for the coarse point cloud registration.

A different approach based on linear plane matching was developed for the coarse registration of airborne LIDAR point clouds [17]. By relying on the presence of linear structures, this approach is limited to specific dataset classes.

The problem of fine registration between point clouds has been intensively studied, and high quality solutions now exist for online applications such as SLAM [3, 4]. The solutions revolve around the Iterative Closest Point (ICP) [18] algorithm and its improvements [19]. A noteworthy fine registration method that is based on the correlation of Extended Gaussian Images in the Fourier domain [20] was proposed as an alternative to ICP, although its final stage again relied on iterations of ICP for fine-tuning. Fine-registration is not the focus of this research, although to achieve end-to-end registration the standard ICP algorithm is utilized in its final stages.

All of the above registration methods were designed for input point cloud pairs that are similar in order of magnitude and low in quantity (under 1 million) of points.

1.3. Contribution

This work proposes and tests two original approaches, applied for the first time as a base to point cloud registration:

1. Using super-points (selected by a Random Sphere Cover Set) as the basic units for matching, instead of the commonly used key-points or local linear structures. This utilizes a much wider variety of geometric structures, and better exploits the available data for finding the correct transformation. In addition, it transforms the complexity of the rest of the algorithm to be correlated to the surface area covered in the point cloud scene instead of on the number of points in the scene. It is simple, fast, and allows for scalability.

2. Encoding local 3D geometric structures using a deep neural network auto-encoder. This method provides state-of-the-art encoding in image analysis applications. By adapting the data and applying this method within the algorithmic pipeline developed, it creates features from within the data that are proven to outperform manually designed local geometry features.

We show here that combining these ideas produces promising registration results on multiple challenging datasets. The method is generic in the sense that it can work with any data regardless of the type of sensor or scene.

While most registration algorithms deal with similar-sized point clouds, we take on the unique problem setup of point clouds that differ significantly in size, and we designed our algorithm to be effective on large-scale scanned data. Although the algorithm requires an initial offline stage, the online stages can be implemented efficiently in parallel, making it suitable for real-time applications.

2. The LORAX Registration Algorithm

We focus on the registration of two point clouds: a global point cloud depicting a large outdoor area, and a small local point cloud captured from within the global point cloud scene. The global point cloud can contain as many as ~100 million 3D points, while the local point cloud is two-to-three orders of magnitude smaller.

In this section we present the LOcalization by Registration using a deep Auto-encoder reduced Cover Set (LORAX) algorithm.

2.1. Algorithm Overview

The algorithm includes the following steps:

1. Division of the point clouds to super-points using the new Random Sphere Cover Set algorithm.
2. Selection of a normalized local coordinate system for each super-point.
3. Projection of super-point data onto 2D depth maps.
4. Saliency detection and filtration of super-points.
5. Dimension reduction using a Deep Neural Network Auto-Encoder.
6. Finding candidate matches between correlating descriptors.
7. Coarse registration using a localized search.
8. Iterative Closest Point fine-tuning.

Next, each stage of the algorithm will be explained in detail and analyzed.

2.2. Random Sphere Cover Set (RSCS)

First, the set of super-points (SP) that will be used as the basic units for the matching process is defined. Each super-point is a subset of points describing a local surface. Overlapping is allowed (i.e., one point can be included in several super-points). To obtain coverage of almost all points in a cloud (~95%) with a manageable representation, we suggest the following iterative procedure: (1) randomly select a point P that does not yet belong to any SP; (2) define a new SP as the set of points located inside the sphere of a fixed radius R_{sphere} with P as its center.

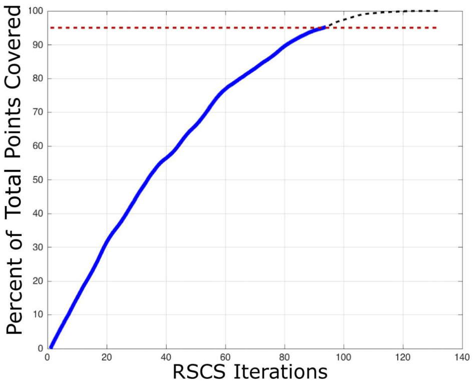


Figure 2: Coverage of points vs. RSCS iterations

This simple procedure, which we refer to as RSCS, has interesting properties that allow the R_{sphere} parameter to be estimated. In random sphere packing, non-overlapping spheres were shown to fill approximately 64% of an enclosed 3D region [21]. Let V_{local} be the volume of a sphere encompassing the local point cloud and let m be the number of matches used in the final stage of the algorithm. To ensure that the minimum, m, of SP pairs will be matched, we select R_{sphere} so that it will be possible to randomly pack $2m$ spheres inside the volume V_{local}.

$$R_{sphere} \approx \left(\frac{3}{4 \cdot \pi} \cdot \frac{0.64}{2m} \cdot V_{local} \right)^{\frac{1}{3}} \tag{1}$$

An estimation of the number of SPs created by the RSCS algorithm given the intrinsic parameters of the local and global point clouds is analyzed in the Supplementary Material (in project Github). It is shown that the method covers points of the point cloud at an exponentially decaying rate. Fig. 2 shows the percentage of points covered as a function of RSCS iterations. The RSCS algorithm is applied once on the global point cloud and multiple times on the local point cloud, for robustness in the later stages. The N_{SP}^{local} SPs found from the multiple applications of RSCS on the local point cloud are combined into a single set representing the local point cloud in the next stages of the algorithm, as shown in Fig. 7(a) and (b).

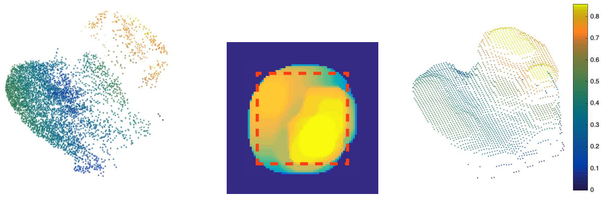
2.3. Selection of a Normalized Local Coordinate System for each Super-point

The local coordination system for a SP is defined as follows: the origin is set to be the centroid of the SP, then the coordinate system of the SP is set using Singular Value Decomposition (SVD) on the estimated covariance matrix of the points within the SP.

The assumption that each SP describes a surface of the scene is utilized in this stage. The eigenvalues of surfaces hold two large eigenvectors of similar size and one smaller eigenvector. This signifies that the points are mostly scattered in two dimensions while the third dimension has significantly lower variance. The z-axis is set to be the third eigenvector. In order to define the x-axis, the mean height of discrete radial slices of the SP are calculated and inserted into a polar histogram. Then the x-axis is set to the direction corresponding to the largest bin. This local coordinate system creates invariance to the location and orientation of the SP, while preserving its geometry.

2.4. Depth Map Projection

After bringing each SP into a local coordinate system, they can be directly compared. However the results will be completely unreliable, as they will have been affected by the variation in point density and by random noise. A dimension reduction is crucial to mitigate these effects. To this end, the continuous point location data is converted into a discrete image format (of size $[d_{im1}, d_{im1}]$). The SP is scaled to the dimensions of the image d_{im1} (we used $d_{im1} = 64$), after which the z-axis height of each point is projected onto the depth map for each corresponding pixel. Finally, the image is cropped to $[d_{im2}, d_{im2}]$ (we used $d_{im2} = 32$), to remove the circular edges of the SP in the depth map. See Fig. 3 (a) and (b).



(a) Example SP (b) The depth map (c) The reconstruction

Figure 3: Super-point depth map projection

To reduce the effects of noise and varying densities, a max filter and then a mean filter are applied to the image. This modification of the SP information can be visualized by reconstructing the SP from the depth map. As shown in Fig. 3(c), the reconstruction reliably holds the same geometric shapes and qualities as the original SP point cloud, while creating a complete coverage over the unknown sparse regions.

2.5. Saliency Detection and Filtration

For the fastest and best quality registration, the number of irrelevant SPs that pass through this pipeline should be reduced. Irrelevant SPs are filtered out by three criteria: density, geometric properties, and saliency levels.

Density Test: The density is measured both in absolute terms and in comparison to other SPs. This means that SPs containing fewer than N_d points are filtered out. In addition, SPs with relatively few points in comparison to their K nearest neighbor SP (Euclidean distance is measured between SP centroids) are also filtered out.

Geometric Quality Test: The height of each SP within its individual local coordinate system is measured. Low height SPs, which signal flat surfaces, are filtered out.

Saliency Test: SP depth maps from the global point cloud are reshaped into a column "depth vector" of length d_{im2}^2. A Principal Component Analysis (PCA) is performed for the set of depth vectors. The SPs (from the local and global point clouds) that are accurately reconstructed using only the first three eigenvectors have commonly found geometric characteristics within the dataset, and are thus filtered out. This reduces the chance of matches between similar SPs located in different areas of the point clouds.

2.6. Auto-Encoder Dimension Reduction

A key stage in this algorithm is the comparison between SP geometries from within global and local point clouds. The comparison of high-dimensional objects is prone to large noise and variance, even with identical semantic meaning. To compare the semantics of the SP geometry, the dimension of the depth map images must be reduced while retaining maximum geometrical information. We constructed and tested two separate dimension reduction methods in this research. The first is based on PCA and the second on a Deep Auto-Encoder (DAE).

2.6.1 Linear Dimension Reduction

The PCA method lowers the dimension of the data by projecting it onto a lower dimension linear hyperplane. A base of k eigenvectors are calculated from the depth vectors of the global point cloud SP (similarly to the saliency detection, but here $k > 3$). The k eigenvalues corresponding to the eigenvectors define the super-position required to recreate each SP. This compact representation feature holding geometric information of the SP is denoted as the Principal Component Analysis Super-point Feature (PCAF). It is important to note that PCA creates a linear projection of the data, which results in high data loss and a relatively large reduced form. This method serves as a comparable benchmark for the DAE method.

2.6.2 Deep Auto-Encoder Dimension Reduction

It was shown that DAE neural networks [16] yield state-of-the-art image compression. Here we use this technique to obtain compact representations of the 2.5D super-point geometry, captured in the depth maps.

DAE neural network architectures are made up of encoder and decoder stages. The encoder stage starts with the input layer and then is connected to hidden layers,

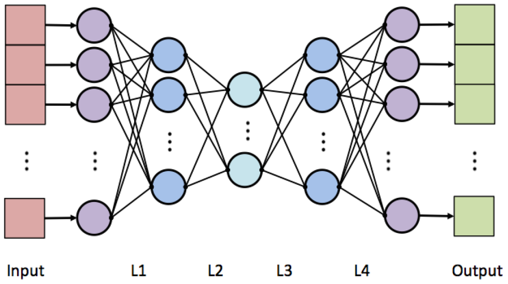


Figure 4: Deep Auto-encoder architecture

which gradually decrease in dimension until reaching the requested compact dimension. The decoder stage starts with the compact representation of the data; each succeeding hidden layer is of a higher dimension, until the output layer dimension, equal to the input dimension, is reached. The loss function is defined as the pixel-wise error between the input and the output layer, optimizing the network to achieve the best compact representation of the image.

To design a DAE that would fit our application, we performed extensive empirical testing and optimization. We concluded that a network with 4 fully connected hidden layers, using the sigmoid non-linear activation function between each of the layers and dropout (DO) on the input layer, returns the satisfactory results. To further reduce the number of parameters in the network, the 4th and 1st hidden layers as well as the 3rd and 2nd layers mirror each other with identical weights, while retaining individually learned bias values. This weight sharing means that the back propagation learning algorithm is constrained to optimize the same weights for the encoding and decoding process [22]. The compact dimension of the reduction is defined by the encoder output dimension. The architecture uses the following dimensions: 1032(Input), [1032,128](L1), [128,10](L2), [10,128](L3), [128,1032](L4), 1032(output). See Fig. 4.

A combination of data driven and synthetic depth maps are utilized to initially train the deep neural network. 100,000 super-point depth maps were used to train the proposed DAE. The training stage is unsupervised, i.e., no manual annotation of data is required and the network is initialized with random weights. This training process is offline, and the network can be improved by updating it periodically with additional point cloud data acquired from scanned local clouds. The offline training process can be lengthy, but the encoding portion can be activated online inexpensively and quickly.

This compact low-dimension representation can be seen as a feature capturing the geometric information of the entire SP. It represents the SP at a fixed lower-dimension vector, not correlated with the number of points in the SP. This is a substantial improvement in the reduction of complexity, in comparison to the competing local descriptors at each point. The SP auto-encoder based Feature (SAF) can be used for many tasks, such as detection or classification of 3D objects, while here we optimize it for registration.

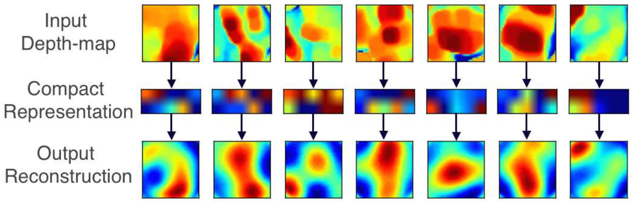


Figure 5: Visualization of deep auto-encoder input, reduction and reconstruction

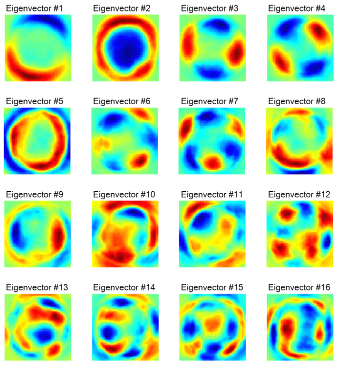
Fig. 5 shows examples of depth maps input into the DAE and reduced to a 10-dimensional SAF (5x2 matrix enlarged for better visualization) and then reconstructed to the original dimensions through the decoder. The height of the depth map is translated into color: blue corresponds to zero height and dark red corresponds to maximal height. The reconstruction is not identical to the input, but it does capture the general geometry of the SP. This is optimal for robustness to noise and small changes—crucial for our application—while capturing the significant SP geometric properties.

To further show the effectiveness of the DAE, the features learned from within the data are analyzed and compared to the eigenvectors calculated from the PCA method. To do this, an independent activation of each dimension in the SAF vector is input into the decoder to visualize what the DAE has learned. See Fig. 6.

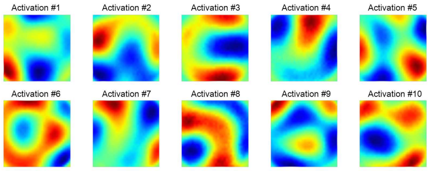
The eigenvectors of the PCA and the independent decoder activations of the DAE are the “building-blocks” of the compact representations created in both methods. Fig. 6(a) shows the growing complexity of each eigenvector, which is sorted by eigenvalue. This is in contrast to the unstructured DAE activation images. See Fig. 6(b). The PCA method can be approximated as a single hidden layer neural network with only linear functions [23]. This means that in order to represent complex geometry, complex eigenvectors are needed. Due to multi-layered super-position of the values in the DAE, complex geometry is represented using relatively simple “building blocks”.

2.7. Selecting Candidates for Matching

After describing each SP by a SAF vector we select a set of similarly described SP pairs to act as candidates for matching. By measuring the Euclidian distance between SAF features, each SP in the local point cloud is paired with its K-nearest-neighbors from the global point cloud (we set K to 3). When the distance associated with the i+1 nearest neighbor is significantly larger than that associated with the i nearest neighbor, we filter out candidates i+1 to K. Note that the number of candidates, $P_{candidates}$ is in the order of O(N_{SP}^{local}). To get a feeling for the number of candidates selected, consider a problem set with a global point cloud of 10 million points, and a local point cloud of 500 thousand points. For this set we get approximately 200 SPs from the local point cloud, meaning that about 550 candidates are selected. See Fig. 7(c).

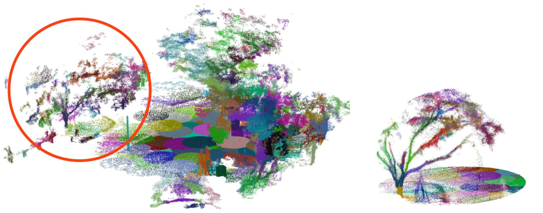


(a) PCA eigenvectors

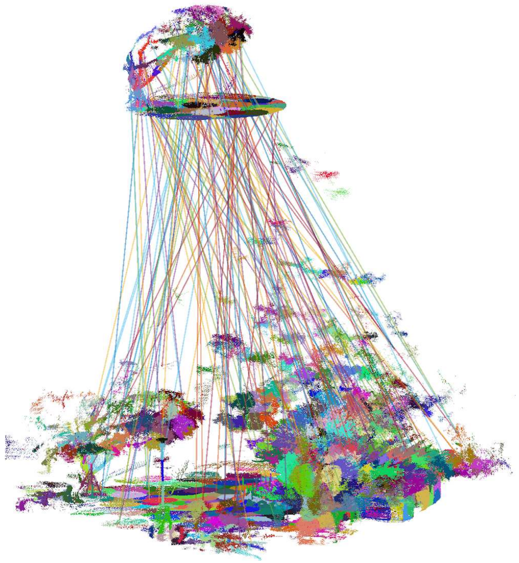


(b) DAE independent decoder activations

Figure 6: Compact Representation Vectors



(a) Global point cloud with RSCS (b) Local point cloud with RSCS



(c) Matching candidate connections

Figure 7: RSCS and matching candidates (each point is colored according to the last RSCS iteration to cover it)

2.8. Coarse Registration by Localized Search

To find the 6DoF (6 degrees of freedom) transformation between the point clouds, at least 3 matches are required (we used $m = 6$ for robustness). Dealing with the search space size of at least $\binom{P_{candidates}}{m}$ is impractical (over 10^{13} for the example above). We therefore consider for each iteration only m candidate-pairs for which all global cloud points can be encompassed in a sphere with a volume not exceeding V_{local}. This reduces the search space of transformation options by 8 to 9 orders of magnitude (reducing the options in the example above to about 40,000). We use a RANSAC [24] procedure, iteratively selecting 6 candidate-pairs, computing a transformation, and checking the consensus by measuring the average (physical) distance between transformed points in the local point cloud and their nearest neighbors in the global point cloud. (To save runtime, we transform only a diluted version of the local point cloud.) We tested 10,000 random selections (about 1/4 of the search space). Instead of selecting only the best scoring transformation as the result of the coarse registration step, we record the 5 best transformations $(T_1, \ldots, T_5)$ in which the local point clouds are non-overlapping. Then the fine-tuning step (described in the next section) is applied to each, and the one that yielded the best scoring fine registration is finally selected.

2.9. Iterative Closest Point Fine-tuning

Simple Iterative Closest Point (ICP) fine tuning is performed, initialized by each of $T_1, ..., T_5$ transformations. The registration with the lowest ICP loss is chosen, defining the LORAX output transformation. This step stems from the realization that the “closest” coarse registration result doesn’t always correlate with the correct registration result, as there are many local minima in the optimization function. The best fine registration is shown empirically to correspond to the best coarse registration in about 75% of the cases, the second-best in about 18% of the cases, and the third-best in about 4%. This stage can be replaced by any fine-tuning approach.

2.10. Efficiency Discussion

Our current implementation was not optimized for real-time performance. However, this algorithm does have the potential to be incorporated in field equipment and perform real-time localization, given that the global point cloud is captured ahead of time via aerial LIDAR or stereographic reconstruction. The neural network training process and the calculations on the global point cloud up to the dimension reduction stage are designed to run offline. The compact descriptors from the global cloud may be saved into the online equipment along with a downsampled version of the global point cloud. Once a local point cloud is captured online from an unknown position within the global scene, the SP division, normalization, saliency detection, and DAE dimension reduction stages can be carried out in parallel for each SP independently. Then KNN candidate selection based on KD-tree [25], RANSAC based localized candidate search [26], and ICP [27] can be accomplished efficiently in parallel as well (using multiple CPUs and/or a GPU). The code for the RSCS method and SAF descriptor are available at: https://github.com/gilbaz/LORAX.

3. Experiments and Results

The advantage of the RSCS SP creation over the FPFH persistent key-point detection and the descriptive quality of SAF in comparison to the FPFH descriptor are shown throughout the experiments. Each stage of the registration was extensively tested using the “Challenging Datasets for Point Cloud Registration Algorithms” [28], matching a close-proximity point cloud to a global large-scale point cloud of the same scene captured in two different seasons. In addition, controlled experiments were carried out using large-scale aerial point clouds, in order to better understand the effects of different types of noise on the registration.

3.1. Challenging Dataset Registration

The datasets utilized contain many point clouds of outdoor scenes, captured by a ground LIDAR scanner, over multiple seasons. By stitching together the point cloud laser scans captured in each scene for each season, an authentic global point cloud is created. This point cloud data is ideal for testing the LORAX algorithm. We test it by registering local point clouds to a global point cloud of the same scene, captured in a different season. This setup is indeed challenging; the scenes contain some rigid stationary objects such as a gazebo structure, lamp poles and benches, but also inconsistent objects like people, bushes and tree branches. This challenge is elevated by missing information due to scanning angles and occlusions.

See Fig. 1 for an example of LORAX’s results. The colors indicate the relative distance between each point in the local point cloud (after being registered) and its nearest neighbor in the global point cloud, where green is closer and red is further.

We test and compare the registration performance with RSCS super-points vs. with key-points, and with FPFH descriptors vs. PCAF and vs. SAF. The results are summarized in Table 1.

For the comparison to key-point based methods we followed [9]. The ‘RSCS+FPFH’ method computed the FPFH descriptor corresponding with the center points of the RSCS superpoints. Each result reported is the average performance for 12 local point clouds, 9 from the ’Gazebo’ dataset and 3 from the ’Wood’ dataset. For each registration result we measure the relative translational error (RTE) and the relative rotation error (RRE) that are used and defined in [29]. When the RTE is below a predefined threshold (we used 1 meter), the registration result is defined as a binary ’success’. For each test we report the binary success rate and the average RTE and RRE scores of the successful tests. All methods use the same fine-tuning procedure and therefore achieve similar RTEs, yet they have different resulting RREs and binary success rates, indicating the quality and the robustness of the coarse registration.

Table 1 shows that using a combination of RSCS for the point cloud sub-division and the DAE to create SAF yields the most robust and highest-quality registration results. The robustness gained from RSCS is evident in the comparison of KP+FPFH to RSCS+FPFH. The comparison RSCS+FPFH to RSCS+PCAF and RSCS+SAF shows the advantage of using machine-learning based features over manually designed features, as well as the advantage of SAF over PCAF.

3.2. Noise, Occlusion, and Density Sensitivity Tests

To further test the quality and limitations of LORAX, we used a few urban outdoor scene point clouds provided by [30]. These point clouds, of approximately 1.5 million points each, depict large areas of 250 square meters. See Fig. 8 for an example.

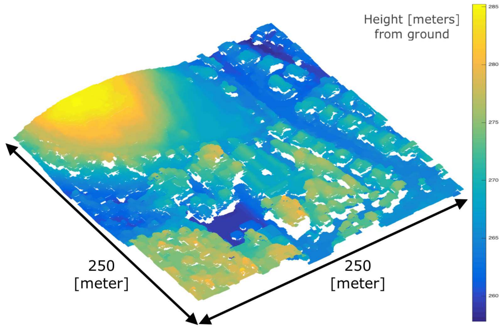


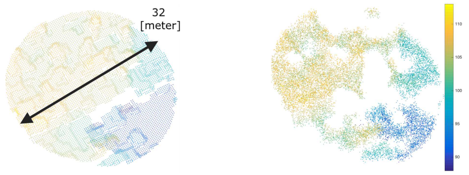
Figure 8: Aerial scanned point cloud depicting hill with surrounding houses and roads. An example from [30] dataset.

	RRE [degrees]	RTE [meters]	Binary Success Rate
KP+FPFH	12.2±4.8	0.44±0.2	8/12 (67%)
RSCS+FPFH	9.1±2.6	0.43±0.24	8/12 (67%)
RSCS+PCAF	7.2±2.3	0.40±0.32	8/12 (67%)
*RSCS+SAF	2.5±1.2	0.42±0.27	10/12 (83%)

Table 1: Registration results

In order to be able to control different parameters of noise, density and occlusions, we performed semi-synthetic experiments in which small point clouds with radii of 15-50 meters were cropped from the large original point clouds. The LORAX and the KP+FPFH registration algorithms were tested on altered versions of the local point clouds, matching them to the original global point cloud.

Noise modification included: (1) adding random noise by randomly moving 10%, 20%, 50% of the points a uniformly distributed distance of up to 3 meters, (2) randomly removing 10%, 20%, 50% of the points to test sensitivity to the density of the cloud, and (3) simulating occlusions by removing 10%, 20%, 50% of points within local random spheres. See Fig. 9.



(a) Original cropped local point cloud

(b) Same cloud with simulated noise and occlusions

Figure 9: Simulating noise, density change, and occlusions

50 randomly cropped point clouds from 3 full scenes were tested to analyze the effects of downsampling (density change) (DS), random relocation noise (RN), and occlusions (OC) on each. Fig. 10 summarizes the results. To clarify, each point on the graph represents the average binary success rate of 50 registration tests given the noise specifications defined. In this experiment the binary success rate is defined by an RTE threshold of 2.5 meters (due to the large scale of the global point cloud).

These results show high robustness to point density, due to the depth map projection stage. Random noise has little effect on our algorithm, due to the SAF representation, which captures only the major geometry characteristics. The occlusion is the hardest defect to deal with. The algorithm overcomes occlusions at a low level, but is greatly hindered otherwise. Overall we see that LORAX is not sensitive to substantial levels of random noise, density change and occlusions, and that its robustness deteriorates only at extreme levels. The KP+FPFH algorithm (dashed line) returned a low binary success rate when tested on clean local point clouds due to the lack of "key-point" inducing scene features, in many sections of the global point cloud. These results add confidence in the direction of this research.

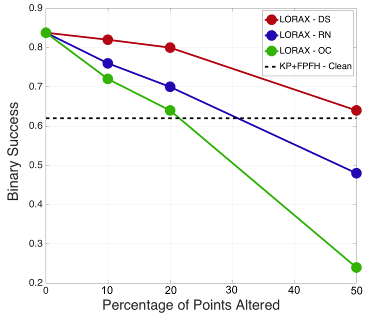


Figure 10: Noise, Occlusion, and Density Sensitivity Tests

4. Conclusion

This paper presented LORAX, an innovative point cloud registration algorithm. With the goal of outdoor localization, this algorithm deals with the challenges of a multiple magnitude difference in the number of points between the two registered point clouds and with a large total number of points involved. Two original approaches were presented: 1) using super-points (selected by a random sphere cover set) as the basic units for matching, instead of key-points and 2) encoding local 3D geometric structures using a deep neural network auto-encoder. We have shown that the combination of these ideas yields promising registration results on challenging datasets. The method is generic is the sense that it can work with any data regardless of the type of sensor or scene. Moreover, while it includes an offline training stage, the online stages can be implemented efficiently in parallel, making it suitable for serving real-time applications.

In future work; we intend to adapt this approach for similar sized point clouds with small scene overlaps. Another interesting direction is designing a multi-scale super-point version of this algorithm. Finally, using a convolutional auto-encoder with an input of height and color information could produce excellent super-point features useful for a wide range of point cloud analysis tasks.

Acknowledgments

This research was supported partially by the Technion and the Magnet Omek Consortium, Ministry of Industry and Trade, Israel. The authors would like to acknowledge Elbit Systems Ltd for providing data for this research.

References

[1] Gary KL Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank C Langbein, Yonghuai Liu, David Marshall, Ralph R Martin, Xian-Fang Sun, and Paul L Rosin. Registration of 3d point clouds and meshes: a survey from rigid to nonrigid. *IEEE transactions on visualization and computer graphics*, 19(7):1199–1217, 2013.

[2] Brent Schwarz. Mapping the world in 3d. *Nat. Photonics*, 4(7):429–430, 2010.

[3] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics &amp; automation magazine*, 13(2):99–110, 2006.

[4] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European Conference on Computer Vision*, pages 834–849. Springer, 2014.

[5] Norbert Haala and Martin Kada. An update on automatic 3d building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(6):570–580, 2010.

[6] Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Soeren Kammel, J Zico Kolter, Dirk Langer, Oliver Pink, Vaughan Pratt, et al. Towards fully autonomous driving: Systems and algorithms. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 163–168. IEEE, 2011.

[7] US DoD. Global positioning system standard positioning service performance standard. *Assistant secretary of defense for command, control, communications, and intelligence*, 2001.

[8] Ben Bellekens, Vincent Spruyt, Rafael Berkvens, Rudi Penne, and Maarten Weyn. A benchmark survey of rigid 3d point cloud registration algorithms.

[9] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3212–3217. IEEE, 2009.

[10] Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, and Michael Beetz. Aligning point cloud views using persistent feature histograms. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3384–3391. IEEE, 2008.

[11] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *European conference on computer vision*, pages 356–369. Springer, 2010.

[12] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 357–360. ACM, 2007.

[13] Bastian Steder, Radu Bogdan Rusu, Kurt Konolige, and Wolfram Burgard. Narf: 3d range image features for object recognition. In *Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, volume 44, 2010.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[16] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[17] Hangbin Wu and Hongchao Fan. Registration of airborne lidar point clouds by matching the linear plane features of building roof facets. *Remote Sensing*, 8(6):447, 2016.

[18] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992.

[19] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152. IEEE, 2001.

[20] Ameesh Makadia, Alexander Patterson, and Kostas Daniilidis. Fully automatic registration of 3d point clouds. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 1297–1304. IEEE, 2006.

[21] GD Scott and DM Kilgour. The density of random close packing of spheres. *Journal of Physics D: Applied Physics*, 2(6):863, 1969.

[22] Yann LeCun et al. Generalization and network design strategies. *Connectionism in perspective*, pages 143–155, 1989.

[23] Terence D Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural networks*, 2(6):459–473, 1989.

[24] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[25] Kun Zhou, Qiming Hou, Rui Wang, and Baining Guo. Real-time kd-tree construction on graphics hardware. *ACM Transactions on Graphics (TOG)*, 27(5):126, 2008.

[26] Donghwa Lee, Hyongjin Kim, and Hyun Myung. Gpu-based real-time rgb-d 3d slam. In *Ubiquitous Robots and Ambient Intelligence (URAI), 2012 9th International Conference on*, pages 46–48. IEEE, 2012.

[27] Deyuan Qiu, Stefan May, and Andreas Nüchter. Gpu-accelerated nearest neighbor search for 3d registration. In *International Conference on Computer Vision Systems*, pages 194–203. Springer, 2009.

[28] François Pomerleau, Ming Liu, Francis Colas, and Roland Siegwart. Challenging data sets for point cloud registration algorithms. *The International Journal of Robotics Research*, 31(14):1705–1711, 2012.

[29] Yanxin Ma, Yulan Guo, Jian Zhao, Min Lu, Jun Zhang, and Jianwei Wan. Fast and accurate registration of structured point clouds with small overlaps.

[30] *Elbit Systems Ltd.*