

Succinct Landmark Database

Tomomi NAGASAKA* and Kanji TANAKA*

(Received January 23, 2012)

Recently developed robotic mapping techniques enable the acquisition of large scale landmark databases. This paper explores an approach for succinct landmark database, which memorizes a large collection of point landmarks while allowing to random access the location of i-th landmark. Our approach combines and extends three independent compression techniques: space coding (space filling curve), succinct representation (directly addressable variable-length codes), and exemplar-based compression (Manhattan world exemplars). Experiments using real datasets evaluate effectiveness of the presented techniques in terms of compactness, access speed, and accuracy of landmark database.

Key Words **:** landmark database, space coding, succinct representation, exemplar-based compression.

1. Introduction

Landmark database plays an important role in robotic mapping and localization applications[1]-.[3] A landmark database aims to memorize a large collection of locations of point landmarks w.r.t. an environment map, while allowing to random access the location of i-th landmark. Classical data structures such as array of landmark locations allow a modern PC to memorize and random access tens of millions of landmarks. However, it is impossible to scale up to larger collection of landmarks, which is becoming available with recently developed robotic mapping techniques.[4] This paper explores *succict representation* of landmark database, i.e. compression of the data structure while preserving the random accessibility, in order to enhance compactness, access speed, and accuracy of landmark database.

The problem explored in this paper might be partially similar to the classical problem of point cloud compression in the field of point-based geometry. Many efforts have focused on compact 3D representation of a given point cloud while preserving details, by employing various types of model-based representations (e.g. meshes,[5] depth maps,[6] primitives,[7] grammars,[8] patches,[9] hybrids[10]) as well as point-based representations (e.g. spatial subdivision,[11] predictive coding[12]), although most of existing techniques do not support the random accessibility. Model-based representations focus on simplification and approximation of surfaces while preserving details, and do not aim to recover the original point cloud. Point-based representations do not support random access to the decompressed points. Predictive coding,[12] where points are predicted from previously coded neighbors, always relies on previous points in the encoded sequence. Spatial subdivision,[11] where connectivity information is eliminated for compression, also needs to decode the entire point set.

*Dept. of Human & Artificial Intelligent Systems

Our approach combines and extends three independent compression techniques: **space coding, succinct representation** and **exemplar-based compression**. The following subsections briefly review individual techniques, and describe our approach.

1.1 Space Coding

Space coding methods aim at linearization of higher dimensional (e.g. 3D) spaces using **space-filling curves** that have several desirable properties,[13] (1) they pass through each point in the space once and only once, (2) two points that are neighbors in space are neighbors along the curve and vice versa, and (3) easy to retrieve neighbor of a point. In particular, we are interested in the **Morton order** method, which uses bit interleaving of binary representation of the individual i-th coordinates of the point. Unlike other order methods such as Peano-Hilbert order, Canto order as well as Spiral order methods, the Morton

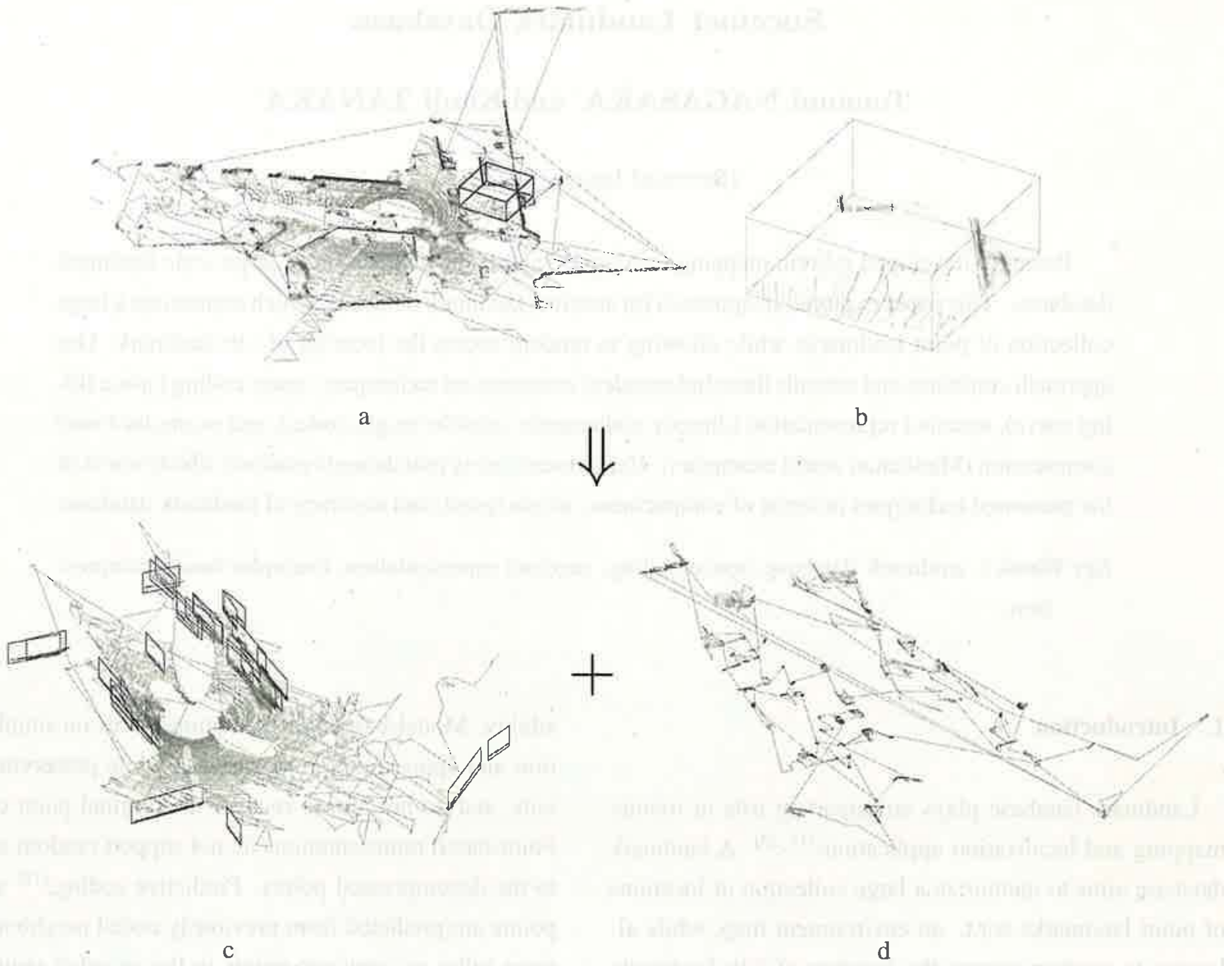


Figure 1: Overview of our approach. Top: Succinct representation, where the input pointset is directly encoded and reordered using the Morton-difference-Gamma coding scheme, and directly addressable variable length encoded. Bottom: Exemplar-based compression, where the input pointset is firstly split into those which belong to exemplar patterns and the rest, and then the resulted pointsets are individually encoded, reordered, and succinctly represented. (a) Input pointset. (b) A close-up. (c) Points belonging to the floor exemplar as well as wall exemplars. (d) Points belonging to non-exemplar patterns.

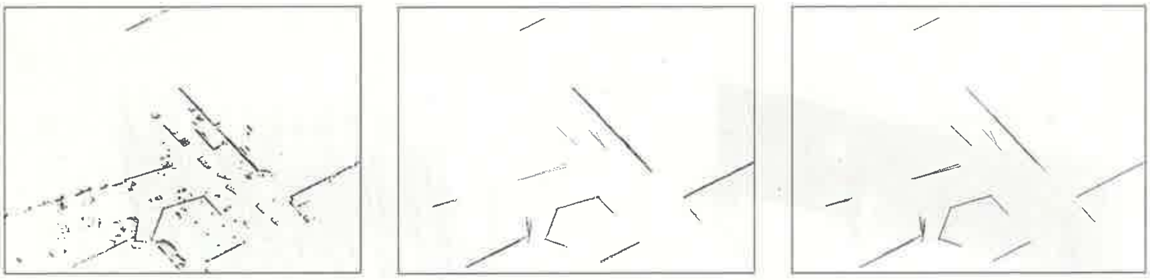
order method has all of the following desirable properties, (4) it supports **easy converting** between the 3D point locations and the curves, (5) it supports **stable** space ordering i.e. the relative order of the individual 3D locations is maintained when the resolution is doubled or halved, (6) it guarantees **admissible** curves, i.e. at every point on the curve, at least one horizontal and one vertical neighbor of each point have already been encountered.

It is reasonable to assume that difference-codes of Morton code are small integers and compress well by **Gamma codes**. Sucn an encoding strategy (**Morton+difference+Gamma**) has been already explored and verified by[14] in the context of constructing nearest neighbor graphs.

1.2 Succinct Representation

We now address succinct representation of the Gamma codes. A well-known difficulty in variable length coding (e.g. Gamma code) is that it is not possible to access directly the i-th encoded element, since its position in the encoded sequence depends on the sum of the lengths of the previous codewords. A naive solution to random access a variable-length encoded sequence is to regularly **sample** it and store the positions of the samples in the encoded sequence, which introduces space and time penalty relying on the sequence's length. In contrast, succinct representation aims to save space cost close to information-theoretic lower bound while allowing random access.

In particular, we adapt **directly addressable variable length codes**, a succinct representation, recently devel-

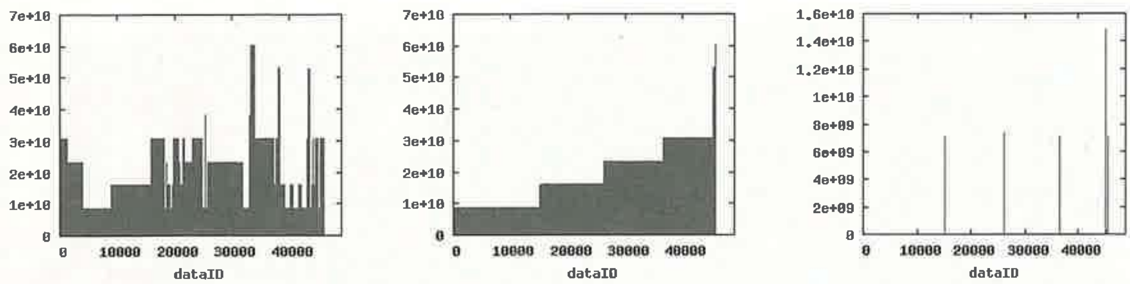


Input (25517 points).

Extraction result (192 lines).

Refinement result (42 lines).

Figure 2: Exemplar extraction.



Morton code.

Sorted Morton code.

Difference Morton code.

Figure 3: Morton code.

oped in,[15] which consumes only exactly the same space as the original encoded sequence, and also supports fast random access as well as partial sums whose time costs are independent from length of the sequence.

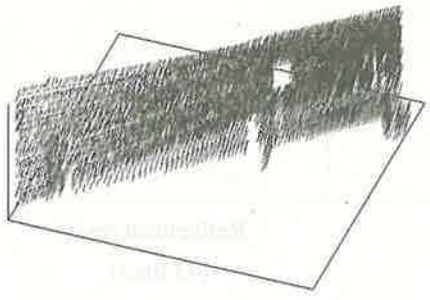
1.3 Exemplar-based Compression

We further consider exemplar-based compression of the pointset's representaton. Its basic idea is to exploit exemplar patterns of local point configuration, such as "walls" and "trees", which should repetitively appear in typical natural scenes as well as in man-made environments. The technique analyzes and detects repetitive patterns in the input pointset, and replaces them with reference to a pattern dictionary. The pattern dictionary is a set of exemplar patterns, and could be learned on-the-fly in the input point cloud, or given in advance in the case of general purpose compressor.

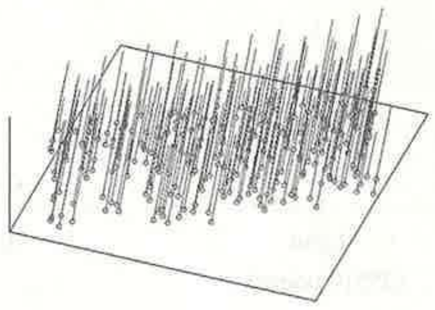
In our approach, we extend the exemplar-based compression scheme so as to allow random access. First, points are reordered in the acsending order of exemplar IDs they belong, in order to eliminate the information of examplar IDs. Second, points belonging to each exemplar pattern are further reordered and encoded into a space filling code (denoted as **local pointID**), so as to allow random access. In the above approach, overheads introduced by exemplar patterns are composed of the dictionary as well as parameters (e.g. size, location, shape) of each exemplar pattern. Points belonging to exemplar patterns compress well, since total length of their space filling curves should be smaller than that of the input pointset. Other points that do not belong to any examplar pattern are directly Morton-difference-Gamma encoded. Best compression ratio is achieved when all the input points belong to exemplar patterns, and the overheads are practically small.

1.4 Contributions

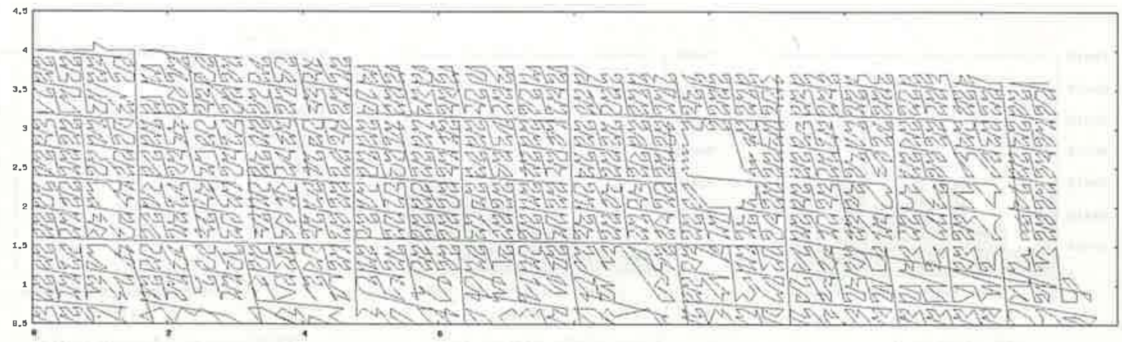
This paper proposes a succinct representation of landmark database. We describe two different schemes in order to facilitate various applications. The first scheme aims to succinctly represent the landmark database in a lossless manner employing the techniques for space coding and succinct representation. The second scheme aims at further compaction of the landmark database in a lossy manner combining the above two techniques with the exemplar-based compression technique. In experiments, we invetigate compactness of either scheme as well as ac-



Projection.



2D points.



Morton curve.

Figure 4: Encoding 2D exemplar.

cess speed and accuracy of the presented techniques using real datasets.

2. Approach

2.1 Succinct Representation

The compression problem addressed in this paper takes as input a sequence of $N+1$ point locations to be compressed $p_0, \cdots, p_N$, and outputs a succinct representation $S(i)$ of the landmark locations and a sequence of pointIDs (denoted as **global pointIDs**) $i_0, \cdots, i_N$ of the original point locations, so that $S(i)$ returns location of the landmark with global pointID i.

As aforementioned, our approach employs the Morton-difference-Gamma encoding scheme. In detail, the input points $p_0, \cdots, p_N$ are encoded into Morton codes (Fig.3left), and reordered in Morton order $\alpha_0, \cdots, \alpha_N$ (Fig.3middle). Then, the Morton code sequence is encoded by difference coding $\Delta\alpha_i = \alpha_i - \alpha_{i-1}$ into a length N sequence $\Delta\alpha_1, \cdots, \Delta\alpha_N$ (Fig.3right). Then, the difference code sequence is encoded into a sequence of Gamma codes $\gamma_1, \cdots, \gamma_N$.

Our scheme for succinct representation is an adaptation of the reordering technique presented in[15] for the Gamma code case. It aims to reorder codes such that it is possible to random access j-th codeword x_j without need of any sampling method. First, it separates the original Gamma code sequence into a pair of sequences of chunks $U_r, \cdots, U_1$ and $B_r, \cdots, B_1$. Each stream U_i (or B_i) contains the i-th significant chunks of the unary part (or the binary part) of a Gamma code. The decoding algorithm recovers at each i-th step, i-th bit of each codeword at once. Since i-th bit of the unary part of a Gamma code indicates whether the codeword terminates at that bit or not, it can identify the number of codewords that have not been terminated yet as well as the concatenation of such non-terminated codewords in an incremental manner. The resulted data structure supports random access to the original Gamma code as well as partial sum $X[j] = x_1 + \cdots + x_j$ for any given i with a constant cost $O(1)$ independent from the length N of original sequence.

Figure 5: Input pointsets.

2.2 Exemplar-based Compression

The basic idea of exemplar-based compression is to replace a part of input pointset with compact reference to a pattern dictionary. The pattern dictionary is a set of exemplar patterns of local point configuration, each of which is a parametric or non-parametric representation of a pointset, and each point is assigned with a unique local pointID.

The encoding procedure aims to obtain a succinct representation of a given input pointset (Fig.1a), and proceeds in the following steps:

1. Split points into those which belong to one of exemplar patterns $P_1, \cdots, P_M$ (Fig.1b) or those which do not belong to any exemplar pattern P_0 (Fig.1c).
2. Encode each point $\{p\}$ belonging to each exemplar pattern $P_i (i \in [1, M])$ with local pointID $j (\in [1, |P_i|])$ into a pairing $k = (i, j)$.
3. Encode each point belonging to the non-exemplar pattern P_0 into a pairing $k = (0, j)$, where j is the 3D Morton code of the point's location.

Points in non-exemplar patterns are directly Morton-difference-Gamma encoded and converted to a succinct representation, denoted as **DB0**.

In experiments, we will focus on typical **Manhattan world -like** environments, mainly composed of planes perpendicular to the floor plane, and employ such plane as an exemplar point configuration. To obtain a set of plane exemplars, the input points are projected onto the floor plane, a set of 2D line segments are extracted from the 2D points by a **Hough transform** algorithm on the 2D projected points, and then those line segments whose score exceeds a preset threshold $C_{score} = 10$ as well as the floor plane are regarded as (2D projections of) exemplar planes (Fig.2), denoted as **DB1**. Given such exemplar planes $P_1, \cdots, P_M$, each point in the input pointset is classified into an exemplar $P_i (i \in [1, M])$ whose distance from the point is smaller than a preset threshold $C_{dist} = 1.0$m if there exists such an exemplar P_i, or the non-exemplar P_0 otherwise. We approximate each point belonging to an exemplar $P_i (i \in [1, M])$ by the perpendicular foot q from the point to the plane, and assign the 2D Morton code j of the 2D location q as the local pointID (Fig.4). Note that 2D Morton code given such a support plane approximates the 3D location with a spatially more efficient manner than the original 3D Morton code.

The exemplars are also encoded with the space coding as well as the succinct representation. First, points belonging to each cluster are reordered in the ascending order of local pointID, and then difference-Gamma encoded and converted to the succinct representation. Second, all the succinct representations are concatenated in the ascending order of clusterID, denoted as **DB2**. Note that length of the resulted sequence exactly equals size of the original pointset. We assign global pointID k to k-th point in the concatenated sequence. The information of clusterID is eliminated from each point, and instead, size of each cluster is memorized using the succinct representation, denoted as **DB3**.

The decoding procedure aims to obtain the location of a point with a given global pointID k, time-space efficiently. It proceeds in the following steps. First, clusterID j of the point is binary searched using the global pointID k as well as the information of cluster size **DB3**. Second, the cluster's 1st point's global pointID k_j is computed using the partial sum operation $X[j]$ over **DB3**. Third, the point's localID k' is computed using the partial sum operations $k' = X[k] - X[k_j]$ over **DB2**, decoded as 2D Morton (if $j > 0$) or 3D Morton (if $j = 0$) code, and then converted to the 3D location using the information of exemplar planes **DB1**, as well as non-exemplar patterns **DB0**.

Obviously, the compression schemes trade time (and accuracy) for space, when compared with the original uncompressed representation. In experiments, we will investigate the tradeoffs using real datasets.

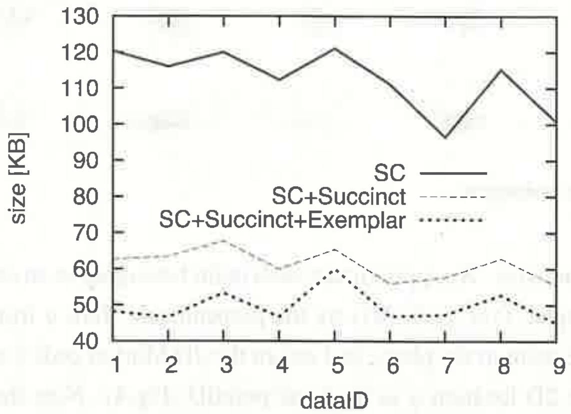


Figure 6: Datasize.

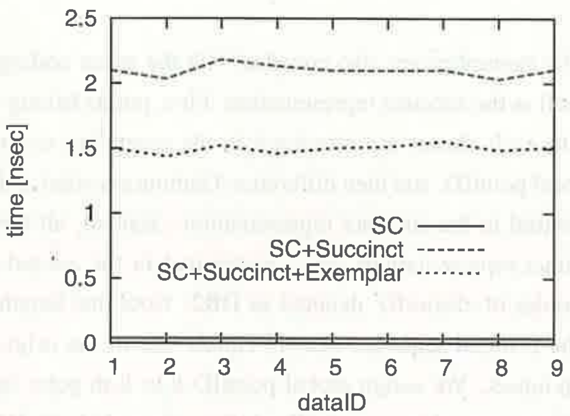


Figure 7: Access speed.

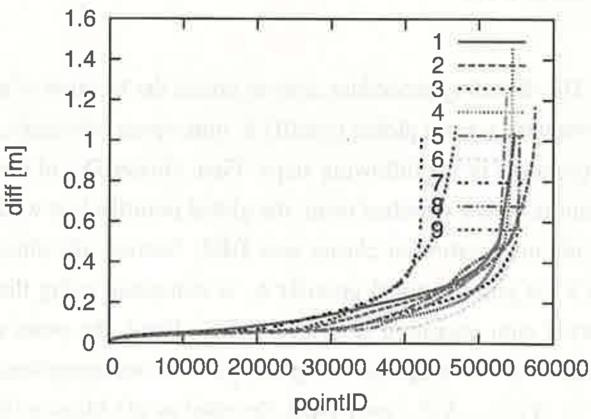


Figure 8: Accuracy.

3. Experiments

We evaluate effectiveness of the presented techniques, in terms of compression ratio, access speed, as well as accuracy. Nine pointsets shown in Fig.5 are employed as benchmark dataset. Each pointset is an unorganized pointcloud with around 60K points acquired by a 3D laser scanner, although we never use the knowledge of sensor's viewpoint for the compression purpose. All experiments are performed on 2.4GHz Intel CPU and 2GB of RAM. The sensor resolution is assumed to be 0.1m. 21bit Morton code per point (+ an offset Morton code) suffice for that resolution. Fig.6 reports size of the original pointset ("SC"), size of the representations presented in 2.1 and in 2.2 ("SC+succinct" and "SC+succinct+Exemplar"), respectively. For fair comparison, the original pointset is fixed length encoded (i.e. 21bits) and stored in an array. One can see that, the variable length coding techniques ("SC+succinct", "SC+succinct+Exemplar") clearly outperform the naive fixed length coding scheme. "SC+succinct+Exemplar" saves around 10KB-20KB when compared with "SC+succinct". Fig.7 shows average time cost for random access per point. One can see that "SC" beats the other two, and "SC+succinct+Exemplar" is slightly better than that of "SC+succinct". Fig.8 reports approximation error of the "SC+succinct+Exemplar". One can see that around half of points have non-negligible approximation error. This is due to that the exemplar-based compression scheme approximates 3D points with their 2D projection onto exemplar planes. The examinations of compression ratio, access speed, as well as accuray for two other resolutions, 0.2m and 0.05m, are also conducted and similar tendency is observed, as reported in Fig.9.

4. Conclusions

The primary contribution of the paper is proposal of succinct representation for landmark database. Our approach combines and extends three independent compression techniques: space coding, succinct representation, and Exemplar-based compression. The succinct representation reorders the sequence in a lossless manner so as to allow random access, which is an essential requirement for landmark database. The exemplar-based compression further compresses the landmark database while preserving the random accessibility. Experiments using real dataset have shown effectiveness of the presented techniques in terms of compactness, access speed, and accuracy.

The tradeoff between accuracy and space of our approach depends on how well the exemplar patterns fit input points. From the viewpoint of accuracy, better fitting between points and exemplars naturally leads to lower approximation error. From the viewpoint of space, better compression ratio is expected when larger portion of the input pointset is compactly explained out by the exemplar patterns. To improve the exemplar-based compression performance is a main direction of our future research.

In this paper, we employed a simple, plane-type exemplar, which is represented by a 2D straight line segment on the floor plane. However, our approach is not restricted to 2D straight line segments, but also applicable to arbitrary shape 2D curved segments, and even general 2D point sequences. Furthermore, a main limitation of segment-based representations is that they essentially require $O(N)$ space for size N input pointset. In our previous studies, we have presented several schemes for compactly representing **sparse feature maps** while preserving random accessibility, including **repetitiveness-based** compression,[16] **grammar-based** compression,[17] **incremental** compression,[18] as well as compression using **geometric priors**.[19] In such "map compression" frameworks, the coding scheme presented in this paper would serve as an essential building block.

References

[1] Adrien Angeli, David Filliat, Stephane Doncieux, and Jean arcady Meyer. A fast and incremental method for loop-closure detection using bags of visual words, " conditionally accpeted for publication in. *IEEE Transactions On Robotics, Special Issue on Visual SLAM*, 2008.

[2] Mark Cummins and Paul Newman. Accelerated appearance-only SLAM. *Proc. IEEE International Conference on Robotics and Automation*, 2008.

[3] B. Williams, G. Klein, and I. Reid. Real-time slam relocalisation. *Proc. IEEE 11th Int. Conf. Computer Vision*, pages 1–8, 2007.

[4] Viorela Ila Kai Ni Frank Dellaert, Justin Carlson and Charles E. Thorpe. Subgraph-preconditioned conjugate gradients for large scale slam. *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2010.

[5] Qian-Yi Zhou and Ulrich Neumann. 2.5d building modeling with topology control. pages 2489–2496, 2011.

[6] Myron Z. Brown, Darius Burschka, and Gregory D. Hager. Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:993–1008, 2003.

[7] Qian-Yi Zhou and Ulrich Neumann. A streaming framework for seamless building reconstruction from large-scale aerial lidar data. pages 2759–2766, 2009.

[8] Peter Wonka, Michael Wimmer, François Sillion, and William Ribarsky. Instant architecture. *ACM Transaction on Graphics*, 22(3):669–677, July 2003.

[9] Hanspeter Pfister, Matthias Zwicker, Jeroen Van Baar, and Markus Gross. Surfels: Surface elements as rendering primitives. *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 335–342, 2000.

[10] Florent Lafarge, Renaud Keriven, Mathieu Brédif, and Hiep Vu. Hybrid multi-view reconstruction by Jump-Diffusion. *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 350–357, 2010.

[11] Ruwen Schnabel and Reinhard Klein. Octree-based point-cloud compression. 2006.

[12] Stefan Gumhold, Zachi Karni, Martin Isenburg, and Hans-Peter Seidel. Predictive point-cloud compression. 2005.

[13] Hans Sagan. Space-filling curves. *Springer-Verlag*, 1994.

[14] Benoît Hudson. Succinct representation of well-spaced point clouds. *CoRR*, abs/0909.3137, 2009.

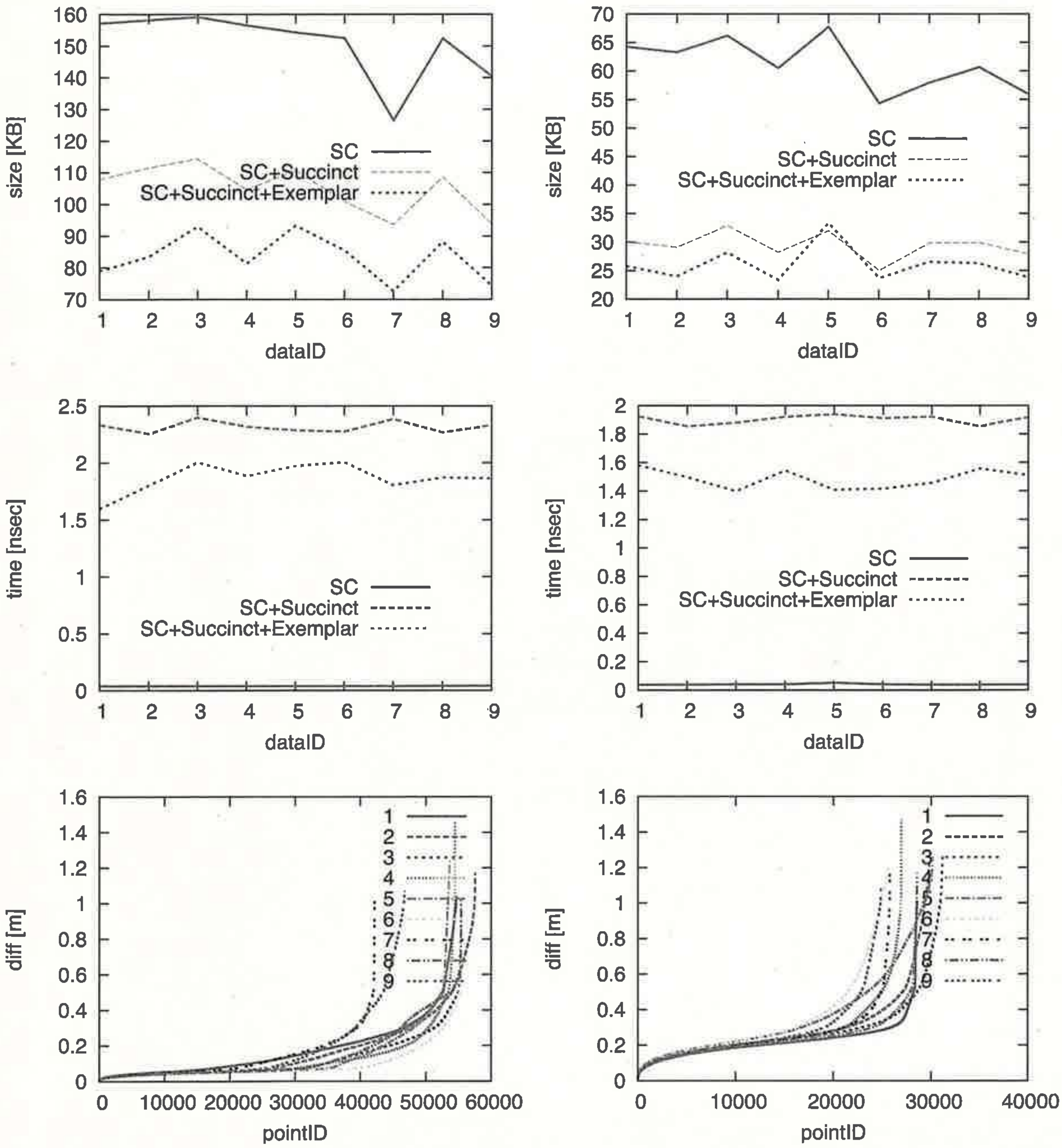
[15] Nieves R. Brisaboa, Susana Ladra, and Gonzalo Navarro. Directly addressable variable-length codes. pages 122–130, 2009.

[16] Nagasaka Tomoni and Tanaka Kanji. Dictionary-based map compression for sparse feature maps. *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2011.

[17] Kondo Kensuke, Tanaka Kanji, and Nagasaka Tomomi. Grammar-based map compression using manhattan-world priors. *Proc. IEEE Int. Conf. Robotics and Biomimetics (ROBIO)*, 2011.

[18] Nagasaka Tomomi and Tanaka Kanji. An incremental scheme for dictionary-based compressive slam. *Proc. IEEE Int. Conf. Intelligent Robots and Systems (IROS)*, 2011.

[19] Nagasaka Tomomi and Tanaka Kanji. Dictionary-based map compression using geometric priors. *Proc. IEEE Int. Conf. Robotics and Biomimetics (ROBIO)*, 2011.



res 0.05

res 0.2

Figure 9: Other resolutions.