Generalized Relative Neighborhood Graph (GRNG) for Similarity Search*

Cole Foster, Berk Sevilmis, and Benjamin Kimia

Brown University, Providence RI 02912, USA
{cole_foster,benjamin_kimia}@brown.edu

Abstract. Similarity search is a fundamental building block for information retrieval on a variety of datasets. The notion of a neighbor is often based on binary considerations, such as the k nearest neighbors. However, considering that data is often organized as a manifold with low intrinsic dimension, the notion of a neighbor must recognize higher-order relationship, to capture neighbors in all directions. Proximity graphs such as the Relative Neighbor Graphs (RNG), use trinary relationships which capture the notion of direction and have been successfully used in a number of applications. However, the current algorithms for computing the RNG, despite widespread use, are approximate and not scalable. This paper proposes a novel type of graph, the Generalized Relative Neighborhood Graph (GRNG) for use in a pivot layer that then guides the efficient and exact construction of the RNG of a set of exemplars. It also shows how to extend this to a multi-layer hierarchy which significantly improves over the state-of-the-art methods which can only construct an approximate RNG.

Keywords: Generalized Realtive Neighborhood Graph · Incremental Index Construction · Scalable Search

1 Introduction

The vast majority of generated data in our society is now in digital form. As the data representation has evolved beyond numbers and strings, whose organization and retrieval are based on cosine similarity in vector spaces through inverted files (Google, Yahoo, Microsoft, etc.), the notion of organization and retrieval has likewise evolved to the use of metric similarity. The task of similarity search, namely, finding the "neighbors" of a given query, is a fundamental building block in such application domains as information retrieval (web search engines, e-commerce, museum collections, medical image processing), pattern recognition, data mining, machine learning, and recommendation systems.

The notion of a "neighbor" has often been captured simply in terms of distances to a query. In this sense, given N objects and a query Q, the most common method for search is finding the k nearest neighbors. A more nuanced notion of "neighbor" considers that dataset object are generally samples of a manifold of low intrinsic dimension. In this sense, a notion of a neighbor is no longer binary, but rather involves higher order interactions, at least triplets.

The structure of the manifold can be captured by embedding it in a low-dimensional Euclidean world. This would then allow the Euclidean structure for

* The support of NSF award 1910530 is gratefully acknowledged.

information retrieval. Many previous approaches, such as hashing and quantization, fall in this category. Another approach is to capture the structure of the manifold with graphs, such as using the kNN graph [2], where every point is connected to the k nearest neighbors. Alternatively, there is a class of *proximity graphs* (also known as empty neighborhood graphs) which rely on trinary relationships between points: Two points define a local neighborhood and if no third point falls in this neighborhood, the two points are neighbors. The two most popular examples are the Relative Neighborhood Graph (RNG) [9, 15] and the Gabriel Graph (GG) [5].

There are a large number of applications that use the RNG. The RNG is used in graph-based visualization of large image datasets for browsing and interactive exploration and is viewed as the smallest proximity graph that captures the local structure of the manifold [11–13]. In urban planning theory, RNGs have been used to model topographical arrangements of cities and the road networks. In internet networks, Escalante *et al.* [3] found that broadcasting over the RNG network is superior to blind flooding. De Vries *et al.* [16] propose to use the RNG to reveal related dynamics of page-level social media metrics. Han *et al.* [8] aims to improve the efficiency of a Support Vector Machine (SVM) classifier by using the RNG to extract probable support vectors from all the training samples. Goto *et al.* [6] use the RNG to reduce a training dataset consisting of handwritten digits to 10% of its original size. A related and more recent area is the selection of training data for Convolutional Neural Networks (CNNs) where the RNG is used to reduce the underlying redundancy of the dataset.

Despite such widespread use of RNG, there is not a large literature on efficient construction of the RNG. Hacid *et al.* [7] propose an approximate incremental RNG construction algorithm for data mining and visualization purposes. The incremental construction algorithm selects two random query dataset items and establishes a link between them. Then for each query dataset item, a hypersphere centered at the query dataset item with a radius proportional to the sum of the distance to the exact nearest neighbor of this query dataset item plus the distance from the nearest neighbor to its farthest neighbor is considered. All the dataset items that fall inside this hypersphere and their previously established links are reconsidered for RNG link validation/invalidation which provides the local index update. The approximate, incremental RNG construction algorithm proposed by Rayar *et al.* [11] first selects two random dataset items and establishes a link between them. Then for each query dataset item, a hypersphere centered at the query dataset item with a radius proportional to the sum of the distance to the exact nearest neighbor of this query dataset item plus the distance from the nearest neighbor to its farthest neighbor is considered. All the dataset items that fall inside this hypersphere are assumed to be candidate RNG neighbors of the query dataset item and hence first the RNG neighbors of the query dataset item among this set is found and the links between the query dataset item and its approximate RNG neighbors are established. Then, the L^{th} edge neighbors of the query dataset item are retrieved and their previously established RNG links are reconsidered for RNG link validation/invalidation due to the introduction of the query dataset item which provides the local index update.

The premise of this paper is that it is possible to efficiently build an *exact* RNG that is scalable to large datasets. The idea is hierarchical, incremental construction with each layer guiding and facilitating the construction of the layer below. We show that the construction of RNG in one layer requires the construction of a novel type of graph, the Generalized Relative Neighborhood Graph (GRNG). The GRNG of a set of pivots allows for exact, efficient, and scalable construction of a large set of exemplars. In turn, the construction of this GRNG of pivots can be based on the GRNG of a coarser set of pivots. This allows for a hierarchical, multi-layer design that efficiently constructs the exact RNG of a large set of exemplars. We show that our construction is more efficient than the competing methods of Hacid et al and Rayar et al. In addition, it is exact compared to brute force construction, in contrast to these methods which have missing or extra links.

2 Formulation and Notation

Consider the set of all objects of interest $\mathcal{X}$ and let $\mathcal{S} \subset \mathcal{X}$ be a dataset containing N such objects. Let $d(x, y)$ denote the distance (dissimilarity metric) between $x, y \in \mathcal{X}$. Two popular graphs used to model $\mathcal{S}$ are the kNN Graph, where each element is connected to its k nearest neighbors, and the Minimum Spanning Tree (MST) which is the spanning tree (connected tree involving all nodes) that has the least cumulative sum of distances over all links.

A *proximity graph* of $\mathcal{S}$ is a graph $G(S, E)$ with nodes $x \in \mathcal{S}$ and links between x and y if certain proximity rules are met. For example, a *Gabriel Graph* (GG) [5] connects two points $x_1, x_2 \in \mathcal{S}$ iff $d^2(x_3, x_1) + d^2(x_3, x_2) \geq d^2(x_1, x_2), \forall x_3 \in \mathcal{S}$. Geometrically, if $\mathcal{X}$ is the Euclidean space, this means that the sphere with x_1x_2 as diameter is empty. Another important example is the *Relative Neighborhood Graph* (RNG) [9, 15] which connects x_1 and $x_2 \in S$ iff

$$\max(d(x_3, x_1), d(x_3, x_2)) \geq d(x_1, x_2), \forall x_3 \in \mathcal{S}. \tag{1}$$

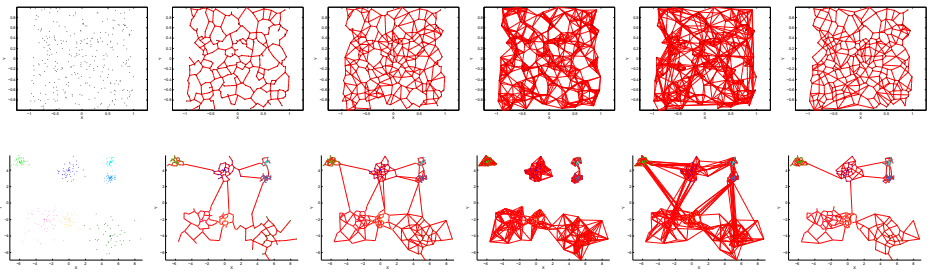
Geometrically, if $\mathcal{X}$ is the Euclidean space, this means that the *lune*(x_1, x_2), namely, the intersection of the two spheres of radius x_1x_2 through centers x_1 and x_2, is empty. Other proximity graphs of interest include the *Delaunay Triangulation* (DT) graph [1] and the β-skeleton graph [10]. Proximity graphs generally require consideration of all members x_3 of S for each pair (x_1, x_2) of S, and as such require $O(N^3)$ for naive construction. Note that 1NN $\subset$ MST $\subset$ RNG $\subset$ GG $\subset$ DG. See Figure 1.

The notion of a *pivot* arises as a way to capture a group of exemplars. Define the *pivot domain*, Figure **??**, $\mathcal{D}$ of pivot p_i and domain radius r_i as,

$$\mathcal{D}(p_i, r_i) = \{x \in \mathcal{S} \mid d(x, p_i) \leq r_i\}. \tag{2}$$

While pivots do not need to be members of S, in our nested approach, the set $\mathcal{P}$ of M pivots, $\mathcal{P} = \{p_1, p_2, ..., p_M\} \subset \mathcal{S}$. The aim is to have a sufficient number of pivots to cover $\mathcal{S}$, *i.e.*, $\mathcal{S} = \bigcup_{i=1}^{M} \mathcal{D}(p_i, r_i)$.

Observe that the knowledge of $d(x, p_i)$ bounds $d(x, y)$ for $y \in S_i$ as $d(x, p_i) - r_i \leq d(y, p_i) \leq d(x, p_i) + r_i$ using the triangle inequality. In the absence of an embed-



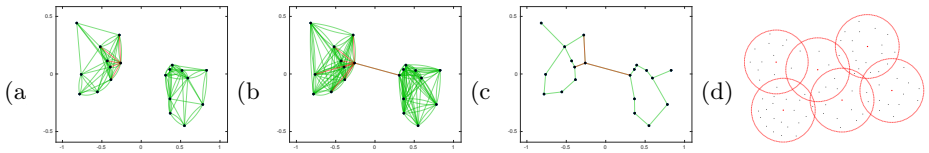
(a) (b) (c) (d) (e) (f)

Fig. 1: A comparison of graphs for representing both uniformly distrbuted (top) and clustered data (bottom). (a) Points in 2D, (b) RNG, (c) GG, (d) kNN, k=8, (e) Tellez [14] b=4, t=4, and (f) NSG [4], R=8.

ding Euclidean structure the triangle inequality is the only constraint available for relative ranking of distances between triplets of points.

Why the Relative Neighborhood Graph? A graph represents the topology of a manifold: nodes are samples of a manifold, while links define the topology of the "tangent plane", *i.e.*, the immediate neighbors. In this sense, the RNG represents local connectivity better than kNN, especially in asymmetric distributed cases, Figure 1. Consider in Figure 2(b-d) how for kNN the connectivity of x is predominantly with the elements on one side and it is not until k is increased to $k = 8$ that neighbors on the other side begin to connect with x. This is of course because the connection between two elements is solely based on the distance relative to others. In contrast, the RNG connectivity involves the local distribution of points so that the local geometry is captured in the graph connectivity from "all sides". This is also why kNN graphs are often disconnected with real data whose distribution is often asymmetric and clustered, bottom row of Figure 1 (b,d), unless k is sufficiently increased, in which case the graph is unnecessarily dense. In contrast, the RNG is proven to be a connected graph. Along the same line, kNN can be sensitive to small perturbations of data as the relative ranking of distances can change due to perturbation, in contrast to the RNG which is stable in generic configurations.

Another benefit of RNG is that it is parameter free, in contrast to kNN where "k" has to be specified, Tellez [14] where "b" and "t" have to be defined, and NSG [4] where "R" has to be defined. It will also be shown later that the out degree of RNG is dependent on the intrinsic dimension of the manifold and it is generally



(a (b (c (d)

Fig. 2: The kNN connectivity is only based on distance between two elements and not on geometric distribution, (a) k=5 and (b) k=8. In contrast, the RNG (c) captures local geometry without regard to distance and requires no parameters. (d) Pivots (red dots) and associated radii define a pivot domain (red discs).

significantly smaller than the out degree of the kNN graph, namely, k, where k has to be sufficiently high to capture the local geometry.

3 Incremental Construction of the RNG

The incremental approach to constructing RNG assumes that RNG($\mathcal{S}$) is available and computes RNG($\mathcal{S} \cup Q$) from it. The query Q is the newest element: *(i) Localize Q within $\mathcal{S}$:* finding the RNG Neighbors of Q. The naive approach would consider for all $x_i \in \mathcal{S}$ whether $\exists x_j \in lune(Q, x_i)$; all x_i with empty lunes are RNG neighbors of Q. Note that this involves $O(N^2)$ operations where $N = |\mathcal{S}|$, and this is clearly not scalable, and *(ii) Adding Q to the dataset:* When the task is search, the first step finds the RNG neighbors. If Q needs to be added, additionally all pairs of existing links between x_i and x_j need to be validated, whether $Q \in \text{lune}(x_i, x_j)$ in which case x_i and x_j are no longer RNG neighbors. This operation is on the order of $O(\alpha N)$ where α is the average out degree of the RNG, typically a small number. Thus, the localization step is significantly more computationally intensive than the validation step.

The remedy to indexing complexity is organization. Specifically, when exemplar groups are represented by pivots, many inferences can take place at the level of pivot domains without computing distances between Q and exemplars. The basic idea in this paper is to construct conditions on pivots that have implications for efficient incremental construction of RNG of exemplars. This is organized in seven stages: *i)* In Stages I,II, and III entire pivot domains $\mathcal{D}(p_i, r_i)$ are discarded from considering RNG neighbor relations with Q by just measuring $d(Q, p_i)$; *ii* Stages IV,V, and VI: pivots are used in invalidating potential RNG links with the remaining exemplars; *iii* Stage VII: pivots are used to exclude entire domains during the RNG validation process of existing links. What relationship between p_i and p_j can prevent the formation of a RNG link between x_i and x_j?

Theorem 1. *Consider exemplars $x_i \in \mathcal{D}(p_i, r_i)$ and $x_j \in \mathcal{D}(p_j, r_j)$. Then*

$$\begin{cases} d\left(p_k, p_i\right) < d\left(p_i, p_j\right) - (2r_i + r_j) \\ d\left(p_k, p_j\right) < d\left(p_i, p_j\right) - (r_i + 2r_j) \end{cases} \Rightarrow \max(d(p_k, x_i), d(p_k, x_j)) < d(x_i, x_j) \tag{3}$$

This theorem, whose proof is in the supplementary appendix, states that a pivot p_k that falls in a lune defined by the intersection of the sphere at p_i with radius $d\left(p_i, p_j\right) - (2r_i + r_j)$ and the sphere at p_j with radius $d\left(p_i, p_j\right) - (r_i + 2r_j)$ also falls in the RNG lune of x_i and x_j, thereby invalidating the potential RNG link between x_i and x_j, *without computing $d(p_k, x_i)$ and $d(p_k, x_j)$*! This is a proximity relationship between p_i, p_j, and p_k, which effectively defines a novel type of graph.

Definition 1. *(Generalized Relative Neighborhood Graph (GRNG)): Two pivots $p_i, p_j \in \mathcal{P}$ have a GRNG link iff no pivots $p_k \in \mathcal{P}$ can be found inside the generalized lune defined by,*

$$\begin{cases} d\left(p_k, p_i\right) \geq d\left(p_i, p_j\right) - (2r_i + r_j) & \text{(4a)} \\ d\left(p_k, p_j\right) \geq d\left(p_i, p_j\right) - (r_i + 2r_j)\,. & \text{(4b)} \end{cases}$$

Observe that GRNG($\mathcal{P}$) is just the RNG when $r_i = 0, \forall i$, thus it is a generalization of it, Figure 3. Also, note that GRNG($\mathcal{P}$) is a superset of RNG($\mathcal{P}$) since lune(p_i, p_j) is larger than the generalized-lune(p_i, p_j), abbreviated as G-lune(p_i, p_j). This implies that the larger r_i and r_j are, the denser the graph is, until it is effectively the complete graph. This places a constraint on how large r_i and r_j can be. Furthermore, it is easy to show that GRNG($\mathcal{P}$) is a connected graph. In practice, all pivots share the same uniform radius, *i.e.*, $r_i = r, \forall i$. The single parameter r is the minimum for which the union of all pivot domains cover $\mathcal{S}$. Thus, the number of pivots M and r are inversely related. In what follows $d(Q, p_i), i = 1, 2, \ldots, M$ is computed.

Stage I: Pivot-Pivot Interaction: The most important implication of the GRNG($\mathcal{P}$) via Theorem 1, is that a lack of a GRNG link between p_i and p_j invalidates all potential links between their constituents. Stage I therefore begins by locating the pivot parents of Q in $\mathcal{P}$, Equation 2. If Q has no parents, Q is added to the set of pivots $\mathcal{P}$ and GRNG($\mathcal{P}$) is updated. Otherwise, Q can only have RNG links with the common GRNG neighbors of *all* of Q's parents. See Figure 4.

Stage II: Query-Pivot Interaction: Stage I removes entire pivot domains from interacting with Q, namely, those exemplars in the domain of pivots that do not have GRNG links to *all* parents of Q. Note, however, that the GRNG lune is significantly reduced in size due to the increased radii, in comparison with RNG, *i.e.*, by $2r_i + r_j, r_i + 2r_j$ on each side. This stage enlarges the G-lune by considering Q itself as a virtual parent pivot with $r_Q = 0$.

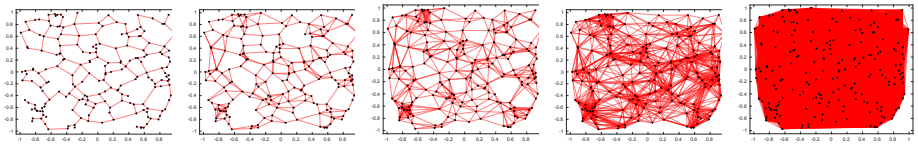
Proposition 1. *If p_k is in the G-lune of (p_i, r_i) and $(Q, r_Q = 0)$, i.e.,*

$$\begin{cases} d(Q, p_k) < d(Q, p_i) - r_i & \text{(5a)} \\ d(p_i, p_k) < d(Q, p_i) - 2r_i. & \text{(5b)} \end{cases}$$

Then, p_k is also in the RNG lune(Q, x_i) $\forall x_i \in \mathcal{D}(p_i, r_i)$, thereby invalidating it, i.e., $\max(d(p_k, Q), d(p_k, x_i)) < d(Q, x_i)$.

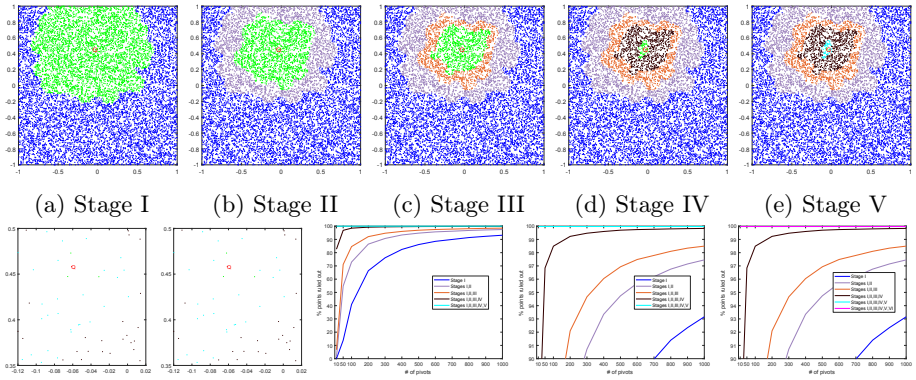
The proof is in the supplementary. Note that since Q is not really a pivot, we cannot simply lookup *GRNG* neighbors of it. Rather, Equations 5 must be explicitly checked for all pivots p_i that survive the elimination round of Stage I. Thus, additional entire pivot domains are eliminated, Figure 4.

Stage III: Pivot-Exemplar Interaction: This stage is symmetric with Stage II by enlarging the G-lune, but instead of using Q as a virtual pivot, an exemplar is



(a) (b) (c) (d) (e)

Fig. 3: GRNG of a set of 200 points in $[-1, 1]^2$ where all $r_i = r$ and for different selection of r: (a) $r = 0$, (b) $r = 0.01$, (c) $r = 0.02$, (d) $r = 0.04$, and (e) $r = 0.419$. When r exceeds $\frac{1}{6}$ the maximum distance between points it is the complete graph (e).



(a) Stage I (b) Stage II (c) Stage III (d) Stage IV (e) Stage V

(f) Stage V (g) Stage VI (h) Stage V (i) Stage V (j) Stage VI

Fig. 4: The savings achieved from Stages I-VI for different number of pivots for a dataset of 10,000 uniformly distributed points in $[-1,1]^2$ where the green area shows remaining exemplars after each stage. (f),(g),(i), and (j) are zoomed in.

used a a virtual, zero-radius pivot. These exemplar are constituents x_j of surviving pivots p_j.

Proposition 2. *If a pivot p_k falls in the G-lune of a parent (p_i, r_i) of Q and $(x_j, r_j = 0)$, i.e.,*

$$\begin{cases} d(p_k, p_i) < d(p_i, x_j) - 2r_i & \text{(6a)} \\ d(p_k, x_j) < d(p_i, x_j) - r_i, & \text{(6b)} \end{cases}$$

then $\max(d(p_k, Q), d(p_k, x_j)) < d(Q, x_j)$ and Q cannot have a RNG link with x_j.

The proof is in the supplementary appendix. In Stage III, then, for all parents of Q, (p_i, r_i), and each exemplar x_j of the remaining pivots p_j, Equations 6 are checked which if valid rule out the exemplar x_j. Note that once a p_k is found that eliminates x_j, the process stops, so it is judicious to pick p_k in order of distance to p_i as closer pivots are more likely to fall in the G-lune of p_i and x_j, Figure 4.

Stage IV: Pivot-Mediated Exemplar-Exemplar Interactions: The aim of the next three stages is to prevent brute-force examination of all exemplars x_k potentially invalidating RNG link(Q,x_i) by falling in lune(Q, x_i). In Stage IV only pivots are checked, *i.e.*, whether pivot p_k satisfies

$$\max(d(p_k, Q), d(p_k, x_i)) < d(Q, x_i), \quad k = 1, 2, ..., M. \tag{7}$$

Observe that only p_k for which $d(p_k, Q) < d(Q, x_i)$ need to be considered, and for those $d(p_k, x_i) < d(Q, x_i)$ is checked. Note that if one p_k satisfies this, link(Q, x_j) is invalidated and the process is stopped, Figure 4.

Stage V: Exemplar-Mediated Exemplar-Exemplar Interactions: In this stage, all the exemplars x_k which may invalidate the potential RNG link between Q and x_i are explored by checking

$$\max\left(d\left(Q,x_k\right),d\left(x_i,x_k\right)\right) < d\left(Q,x_i\right). \tag{8}$$

Observe that since the process stops if one x_k falls in the lune, so it is judicious to begin with a select group of x_k that would more likely fall in the lune(Q,x_i). First, the closest neighbors of x_i can be found by consulting the RNG neighbors of x_i and neighbors of neighbors, and so on until $d(x_i,x_k)$ exceeds $d(Q,x_i)$. Second, since some distances $d(Q,x_k)$ have been computed and cached for other purposes, these can be rank-ordered and these x_k can be explored until $d(Q,x_k)$ exceeds $d(Q,x_j)$, Figure 4.

Stage VI: RNG Link Verification: If the potential RNG link(Q,x_i) is not invalidated by the select group of exemplars x_k, the entire remaining set of x_k must exhaustively be considered to complete the verification. Note, however, that exemplars x_k in pivot domain p_k can be excluded from this consideration and without the costly computation of $d(Q,x_k)$ if the entire pivot domain is fully outside the lune(Q,x_i):

Proposition 3. *No exemplar x_k of pivot domain p_k can fall in lune(Q,x_i) if*

$$\max(d\left(Q,p_k\right)-\delta_{\max}(p_k),d\left(x_i,p_k\right)-\delta_{\max}(p_k)) \geq d(Q,x_i), \tag{9}$$

where $\delta_{\max}(p_k)=\max_{\forall x_k,d(x_k,p_k)\leq r_k} d\left(p_k,x_k\right)$ is the maximum distance of exemplar $x_k \in \mathcal{D}(p_k,r_k)$ from p_k.

The proof can be found in the supplementary appendix. For the remaining pivot domains, the computation of $d(Q,x_k)$ can still be avoided for some exemplar x_k:

Proposition 4. *Any exemplar x_k in the pivot domain of p_k for which*

$$\max(d\left(Q,p_k\right)-d\left(x_k,p_k\right),d\left(x_i,p_k\right)-d\left(x_k,p_k\right)) \geq d(Q,x_i) \tag{10}$$

falls outside lune(Q,x_i). Proof is in the supplementary appendix.

Any exemplar x_k which is not ruled out by Proposition 3 and 4 must now be explicitly considered. If none are in the lune(Q,x_i), then link(Q,x_i) is validated.

Stage VII: Existing RNG Link Validation: The above six stages locate Q in the RNG and identify its RNG neighbors. This is sufficient for a RNG search query. However, if the dataset $\mathcal{S}$ is to be augmented with Q, a final check must be made as to which existing RNG links would be removed by the presence of Q. While this is a brute force $O(\alpha N)$ operation, it is important to avoid computing $d(Q,x_i)$ for all $x_i \in \mathcal{S}$. Observe that Q does not threaten links that are "too far" from it. This notion can be implemented if two parameters are maintained, one for exemplars and one for pivots:

$$\bar{\mu}_{\max}\left(x_i\right)=\max_{x_j\in \mathrm{RNG}\left(x_i\right)} d\left(x_i,x_j\right), \mu_{\max}\left(p_i\right)=\max_{d(x_i,p_i)\leq r_i}\left[\bar{\mu}_{\max}\left(x_i\right)+d(x_i,p_i)\right]. \tag{11}$$

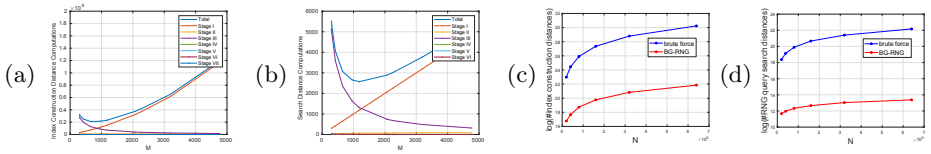


Fig. 5: Stage-by-stage distance computations for construction (a) and search (b) with N=102,400 exemplars uniformly distributed in 2D.

Proposition 5. *A query Q does not invalidate RNG links at x_i if $d(Q, x_i) \geq \bar{\mu}_{\max}(x_i)$. A query Q does not invalidate any RNG link of any exemplars $x_i \in \mathcal{D}(p_i, r_i)$, if $d(Q, p_i) > \mu_{\max}(p_i)$.*

The proof is in the supplementary. This proposition suggests a three-step procedure: (i) remove entire pivot domains if $d(Q, p_i) \geq \mu_{\max}(p_i)$; (ii) remove all exemplars in the remaining pivot domains for which $d(Q, x_i) \geq \bar{\mu}_{\max}(x_i)$; (iii) check the RNG condition explicitly for the remaining x_i and any x_j it links to. This completes the incremental update of $\mathcal{S}$ to $\mathcal{S} \cup \{Q\}$.

Experimental Results The improvements due to this two-layer GRNG-RNG configuration are examined in experiments by varying dimensions and number of exemplars. Figure 5 examines the number of distance computations required for construction and search per stage as a function of the number of pivots. Observe that the first stage cost increases exponentially while the remaining stages experience an exponential drop. This is also observed for search distances per query. The total cost thus has an optimum for each. Since construction is offline while search is online, the number of pivots is optimized for the latter. Figure 5(c) examines the search costs for different dimensions. It is clear that search time rises exponentially with increasing dimension. Observe from Figure 5(b) that additional pivots would have enjoyed the exponential drop in all stages except for Stage I which involves GRNG Construction. If the cost of this stage as a function of M can be lowered, the overall cost will be decreased dramatically. The next section proposes a two-layer scheme for constructing GRNG using a coarser GRNG in the same way the RNG construction was guided by a GRNG.

4 Incremental Construction of the GRNG

The question naturally arises whether the construction of the GRNG of the pivot layer itself can benefit from a two-layer pivot-based indexing approach similar to the construction of the same for the RNG of the exemplars. Formally, let $\bar{\mathcal{P}} = \left\{(\bar{p}_{\bar{i}}, \bar{r}_{\bar{i}}) | i = 1, 2, \ldots, \bar{M}\right\}$ denote pivots obtained from the previous section; refer to these as *fine-scale pivots* to distinguish them from the *coarse-scale pivots* $\mathcal{P} = \{(P_i, r_i) | i = 1, 2, \ldots, M\}$. The idea is for each coarse-scale pivot p_i to represent a number of fine-scale pivots $\bar{p}_{\bar{i}}$. Define the *Relative Pivot Domain* $\mathcal{D}(p_i, r_i)$ as the set of all fine-scale domain pivots $(\bar{p}_{\bar{i}}, \bar{r}_{\bar{i}})$ whose entire exemplar domain is within a radius of r_i, *i.e.*, $d(p_i, \bar{p}_{\bar{i}}) \leq r_i - \bar{r}_{\bar{i}}$. In this scenario, a query Q is either a fine-scale pivot for now with r_Q matching that of other fine-scale pivots, or it can be considered

a fine-scale pivot with zero radius. The query computes $d(Q, p_i), i = 1, 2, \ldots, M$ and if $d(Q, p_i) < r_i - r_Q$, p_i is a parent of Q. The question then arises as to what kind of graph structure for the coarse-scale pivots can efficiently locate a query in the GRNG of the fine-scale pivots. The following shows that the GRNG of coarse-scale pivots can accomplish this:

Stage I: "Coarse-Scale Pivot" - "Coarse-Scale Pivot" Interactions:

Theorem 2. *Consider two fine-scale pivots* $(\bar{p}_{\bar{i}}, \bar{r}_{\bar{i}}) \in \mathcal{D}(p_i, r_i)$ *and* $(\bar{p}_{\bar{j}}, \bar{r}_{\bar{j}}) \in \mathcal{D}(p_j, r_j)$. *Then, if* (p_i, r_i) *and* (p_j, r_j) *do not share a GRNG link,* $(\bar{p}_{\bar{i}}, \bar{r}_{\bar{i}})$ *and* $(\bar{p}_{\bar{j}}, \bar{r}_{\bar{j}})$ *cannot have a GRNG link either. The proof is in the supplementary appendix.*

This theorem, in analogy to Theorem 1 of the previous section, allows for the efficient localization of a query Q for search in stating that the fine-scale GRNG neighbors of Q are only among children of coarse-scale GRNG neighbors of Q's parents, thus, removing entire pivot domains of non-neighbors, see Figure 6.

Stage II: Query - "Coarse-Scale Pivot" Interactions:

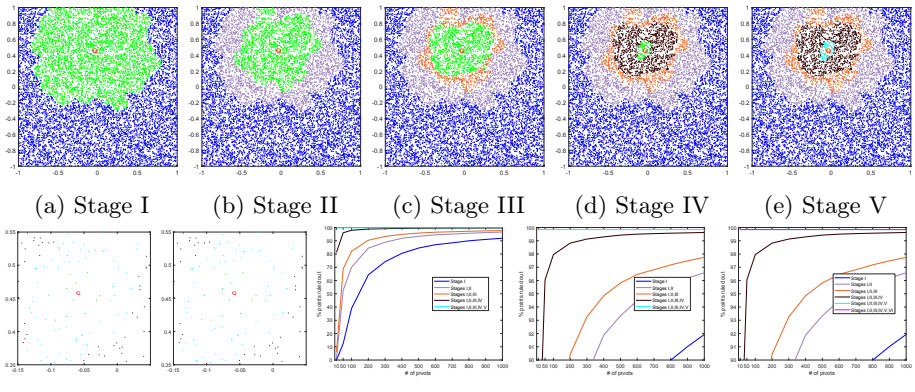
In this stage, (Q, r_Q) is considered as a virtual pivot.

Proposition 6. *The query* Q *does not form GRNG links with any children* $(\bar{p}_{\bar{i}}, \bar{r}_{\bar{i}})$ *of those coarse-scale pivots* (p_i, r_i) *that do not form a GRNG link with* Q *when considered as a virtual pivot with* $r_Q = 0$. *The proof is in the supplementary appendix.*

Stage III: "Coarse-Scale Pivot" – "Fine-Scale Pivot" Interactions:

This stage is mirror symmetric to Stage II, except that instead of treating Q as a virtual coarse-scale pivot, a specific fine-scale pivot $(\bar{p}_{\bar{j}}, \bar{r}_{\bar{j}})$ is considered a virtual pivot.

Proposition 7. *If* $(\bar{p}_{\bar{j}}, \bar{r}_{\bar{j}})$ *does not form a coarse-scale GRNG link with a parent* (p_i, r_i) *of* Q, *then* $(\bar{p}_{\bar{j}}, \bar{r}_{\bar{j}})$ *does not form a fine-scale GRNG link with* (Q, r_Q).



(a) Stage I (b) Stage II (c) Stage III (d) Stage IV (e) Stage V

(f) Stage V (g) Stage VI (h) Stage V (i) Stage V (j) Stage VI

Fig. 6: The savings achieved from Stages I-VI for a dataset of 10,000 uniformly distributed points in $[-1, 1]^2$ where the green area shows remaining exemplars after each stage. (f),(g),(i), and (j) are zoomed in.

The proof is simply an application of Theorem (2) with $(\bar{p}_{\bar{j}}, \bar{r}_{\bar{j}})$ considered as both a fine-scale and a coarse-scale pivot. This third stage rules out all the remaining fine-scale pivots which are not a GRNG neighbor of **all** Q's parents, Figure 6.

Stage IV: "Coarse-Scale Pivot"–Mediated "Fine-Scale Pivot" Interactions: All the GRNG links between the remaining fine-scale pivots and Q must now be investigated. In Stage IV only coarse-scale pivots are considered as potential occupiers of the G-lune by probing

$$\begin{cases} d\left(p_k, Q\right) < d\left(Q, \bar{p}_{\bar{j}}\right) - \left(2\bar{r}_Q + \bar{r}_{\bar{j}}\right) & \text{(12a)} \\ d\left(p_k, \bar{p}_{\bar{j}}\right) < d\left(Q, \bar{p}_{\bar{j}}\right) - \left(\bar{r}_Q + 2\bar{r}_{\bar{j}}\right). & \text{(12b)} \end{cases}$$

Since $d(Q, \bar{p}_{\bar{j}}) - (2r_Q + \bar{r}_{\bar{j}})$ is a known value, only pivots p_k closer to Q than this value need to be considered. Similarly, for $d(\bar{p}_{\bar{j}}, \bar{r}_{\bar{j}}) \in \mathcal{D}(p_j, r_j)$, observe that $d(p_k, \bar{p}_{\bar{j}}) \geq d(p_k, p_j) - (r_j - \bar{r}_{\bar{j}})$, so that if $d(p_k, p_j) \geq d(Q, \bar{p}_{\bar{j}}) - (r_Q + 2\bar{r}_{\bar{j}}) + (r_j - \bar{r}_{\bar{j}})$, then Equation (12b) does not hold and there is no need to consider such p_k. Thus, very few p_k are actually considered, Figure 6.

Stage V: "Fine-Scale Pivot" – Mediated "Fine-Scale Pivot" Interactions: Those links between Q and $\bar{p}_{\bar{j}}$ that survive the pivot test must now test against occupancy of G-lune$(Q, \bar{p}_{\bar{j}})$ by exemplars $\bar{p}_{\bar{k}}$. In this stage, a select group of $\bar{p}_{\bar{k}}$, namely those close to Q and $\bar{p}_{\bar{j}}$ which are more likely to be in G-lune$(Q, \bar{p}_{\bar{j}})$ are considered, leaving the rest to Stage VI. Specifically, these are the $k = 25$ nearest neighbors of Q and $\bar{p}_{\bar{j}}$, Figure 6.

Stage VI: "Fine-Scale Pivot" "Fine-Scale Pivot" Interactions: Very few fine-scale pivots $\bar{p}_{\bar{j}}$ remain at this stage. These need to be validated with all other fine-scale pivots $\bar{p}_{\bar{k}}$. However, the following proposition prevents consideration of a majority of them. Define

$$\delta_{\max}\left(p_k\right) = \max_{\forall \bar{p}_{\bar{k}},\, d(p_k, \bar{p}_{\bar{k}}) \leq (r_k - \bar{r}_{\bar{k}})} d(p_k, \bar{p}_{\bar{k}}). \tag{13}$$

Proposition 8. *All fine-scale pivots* $(\bar{p}_{\bar{k}}, \bar{r}_{\bar{k}}) \in \mathcal{D}(p_k, r_k)$ *satisfying*

$$\begin{cases} d\left(Q, p_k\right) - \delta_{\max}(p_k) \geq d\left(Q, \bar{p}_{\bar{j}}\right) - \left(2\bar{r}_Q + \bar{r}_{\bar{j}}\right) & \text{(14a)} \\ d\left(\bar{p}_{\bar{j}}, p_k\right) - \delta_{\max}(p_k) \geq d\left(Q, \bar{p}_{\bar{j}}\right) - \left(2\bar{r}_{\bar{j}} + \bar{r}_Q\right) & \text{(14b)} \end{cases}$$

fall outside the G-lune$(Q, \bar{p}_{\bar{j}})$, for a query $(Q, \bar{r}_Q)$ *and a fine-scale pivot* $(\bar{p}_{\bar{j}}, \bar{r}_{\bar{j}})$.

Proof is in the supplementary appendix. This proposition excludes entire pivot domains from the validation process. The following proposition further restricts the remaining sets.

Proposition 9. *All fine-scale pivots* $(\bar{p}_{\bar{k}}, \bar{r}_{\bar{k}}) \in \mathcal{D}(p_k, r_k)$ *satisfying*

$$\begin{cases} d\left(Q, p_k\right) - d\left(p_k, \bar{p}_{\bar{k}}\right) \geq d\left(Q, \bar{p}_{\bar{j}}\right) - \left(2\bar{r}_Q + \bar{r}_{\bar{j}}\right) & \text{(15a)} \\ d\left(\bar{p}_{\bar{j}}, p_k\right) - d\left(p_k, \bar{p}_{\bar{k}}\right) \geq d\left(Q, \bar{p}_{\bar{j}}\right) - \left(\bar{r}_Q + 2\bar{r}_{\bar{j}}\right), & \text{(15b)} \end{cases}$$

falls outside the GRNG-lune$(Q, \bar{p}_{\bar{j}})$ for a query $(Q, \bar{r}_Q)$ *and a fine-scale pivot* $(\bar{p}_{\bar{j}}, \bar{r}_{\bar{j}})$.

Proof is in the supplementary appendix. After the majority of fine-scale pivots $(\bar{p}_{\bar{k}}, \bar{r}_{\bar{k}})$ have been eliminated, the remaining ones must test the two GRNG conditions. For efficiency, if first condition $d(Q, \bar{p}_{\bar{k}}) < d(Q, \bar{p}_{\bar{j}} - (2\bar{r}_Q + \bar{r}_{\bar{j}})$ does not hold, the second condition $d(\bar{p}_{\bar{j}}, \bar{p}_{\bar{k}}) < d(Q, \bar{p}_{\bar{j}} - (\bar{r}_Q + 2\bar{r}_{\bar{j}})$ need not be tested, Figure 6. **Stage VII: "Coarse-Scale Pivot" – "Fine-Scale Pivot" Validations:** The incremental construction requires checking which existing GRNG links may be invalidated by the addition of Q. Define first,

$$\begin{cases} \bar{\mu}_{\max}\left(\bar{p}_{\bar{i}}\right) = \max\limits_{\bar{p}_{\bar{j}},\, \mathrm{GRNG}(\bar{p}_{\bar{i}})} \left[d\left(\bar{p}_{\bar{i}}, \bar{p}_{\bar{j}}\right) - \left(2\bar{r}_{\bar{i}} + \bar{r}_{\bar{j}}\right)\right] & \text{(16a)} \\ \mu_{\max}\left(p_i\right) = \max\limits_{\forall(\bar{p}_{\bar{i}}, \bar{r}_{\bar{i}}) \in \mathcal{D}(p_i, r_i)} \left[\bar{\mu}_{\max}\left(\bar{p}_{\bar{i}}\right) + d(p_i, \bar{p}_{\bar{i}})\right]. & \text{(16b)} \end{cases}$$

Proposition 10. *The insertion of Q does not invalidate any GRNG links involving fine-scale pivot $\bar{p}_{\bar{i}}$ for which*

$$d(Q, \bar{p}_{\bar{i}}) \geq \bar{\mu}_{\max}(\bar{p}_{\bar{i}}). \tag{17}$$

Furthermore, the insertion of Q does not interfere with the GRNG link involving fine-scale pivots $(\bar{p}_{\bar{i}}, \bar{r}_{\bar{i}}) \in \mathcal{D}(p_i, r_i)$ if

$$d(Q, p_i) \geq \mu_{\max}(p_i). \tag{18}$$

Proof is in the supplementary appendix. The proposition suggests a three-step approach to examining existing links: (i) Remove all coarse-scale pivot domains p_i

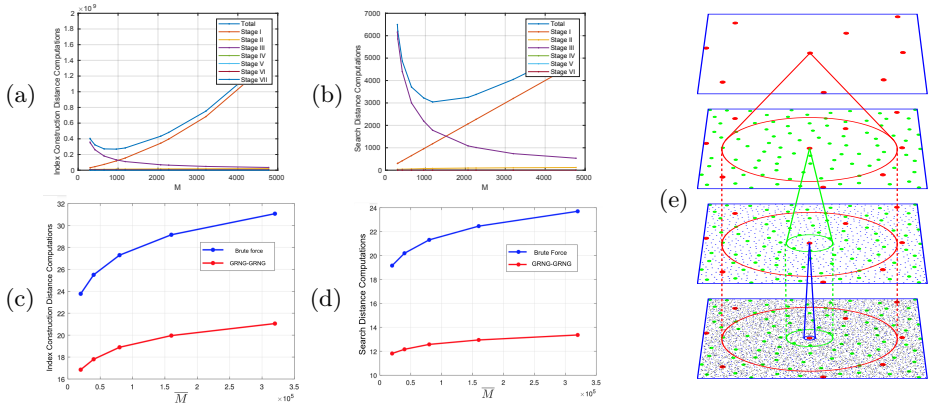


Fig. 7: Stage by stage analysis for GRNG-GRNG hierarchy for $\bar{M} = 102,400$ uniformly distributed fine-scale pivots in 2D as a function of M, the number of coarse-scale pivots. The number of distance computations for construction (a) and search (b) show Stage I is increasing with M while the other stages exponentially decaying with an optimum for each in total. The improvements of GRNG-GRNG with respect to brute-force as a function of M for construction (c) and search (d) distances is significant (e). The monotonically increasing Stage I in (a-b) suggest using a multi-layer hierarchy.

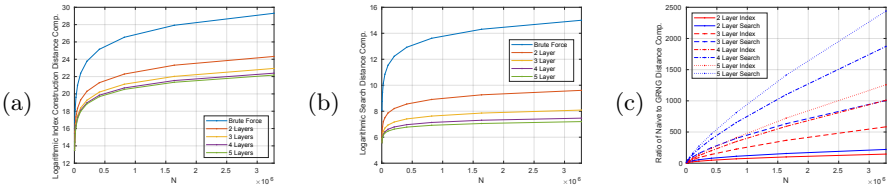
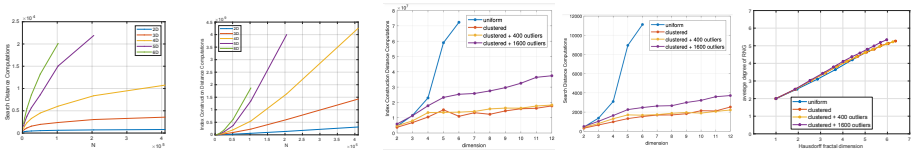


Fig. 8: The index construction and search distance computations of a standard, brute force algorithm, and multilayer hierarchical GRNG networks compared as a function of N.

satisfying Equation 18; (*ii*) Remove all fine-scale pivot domains $(\bar{p}_{\bar{i}}, \bar{r}_{\bar{i}})$ satisfying Equation 17; (*iii*) For any remaining fine-scale pivot $(\bar{p}_{\bar{i}}, \bar{r}_{\bar{i}})$ connecting with $(\bar{p}_{\bar{j}}, \bar{r}_{\bar{j}})$, if Q is in the G-lune$(\bar{p}_{\bar{i}}, \bar{p}_{\bar{j}})$, then the link needs to be removed.



(a) (b) (c) (d) (e)

Fig. 9: The construction distance increases as a function of number of exemplars and dimensions (sublinear) for uniformly distributed data. However, with clustered data, even with outliers, both construction costs and search distances increase much less rapidly.

5 Experiments

We first explore the effectiveness of the GRNG Hierarchy on synthetic data. Figure 9d shows the results for optimal multi-layer hierarchies on uniform and clustered data. We also evaluate on several real-world datasets, namely, COREL, MNIST, and LA. For MNIST, a neural network trained using triplet loss was used to reduce the 784D Euclidean representation into 64D. The results are shown in Table 1. These results show that our method is significantly more efficient while also producing the exact RNG.

References

1. Delaunay, B.: Sur la sphère vide. A la mémoire de Georges Voronoï. Bulletin de l'Académie des Sciences de l'URSS (6), 793–800 (1934)
2. Dong, W., Moses, C., Li, K.: Efficient k-nearest neighbor graph construction for generic similarity measures. In: Proceedings of the 20th international conference on World wide web. pp. 577–586 (2011)
3. Escalante, O., Pérez, T., Solano, J., Stojmenovic, I.: RNG-based searching and broadcasting algorithms over internet graphs and peer-to-peer computing systems. In: The 3rd ACS/IEEE Int. Conf. on Comp. Systems & App. p. 17. IEEE (2005)

Table 1: Results for real world datasets. (top) Corel, 68,040 instances in 57D (middle) MNIST, 60,000 instances with 64D embeddings obtained through a neural network, and (bottom) LA, 1,073,727 instances in 2D. Both the brute-force method and the algorithm by Hacid et al are impractical to run on a dataset of such size. The accuracy of Rayar et al is found by comparing to our method.

Algorithm	Total Links	Extra (+) and Missing (-) Links	Average Degree	Search Distances	Index Construction Distances
Hacid et. al	212,211	21,802/-4	6.2378	177,972.36	9,823,840,198,726
Rayar et. al	190,908	535/-40	5.6116	169,575.08	6,432,673,175
Ours	190,413	0/-0	5.5971	43,729.20	1,611,369,217

Algorithm	Total Links	Extra (+) and Missing (-) Links	Average Degree	Search Distances	Index Construction Distances
Hacid et. al	118,248	+3,778/-3	3.9416	87,713.10	1,430,022,984,523
Rayar et. al	114,893	+865/-445	3.8298	88,172.04	2,639,416,420
Ours	114,473	+0/-0	3.8158	10,058.90	407,689,553

Algorithm	Total Links	Extra (+) and Missing (-) Links	Average Degree	Search Distances	Index Construction Distances
Hacid et. al	Impractical				
Rayar et. al	1,277,369	+3254/-33706*	2.3793	2,147,500	1,153,035,099,784
Ours	1,307,821	-	2.4360	1,020.71	1,042,175,220

4. Fu, C., Xiang, C., Wang, C., Cai, D.: Fast approximate nearest neighbor search with the navigating spreading-out graph. Proceedings of the VLDB Endowment **12**(5), 461–474 (2019)
5. Gabriel, K.R., Sokal, R.R.: A new statistical approach to geographic variation analysis. Systematic Zoology **18**(3), 259–278 (1969)
6. Goto, M., Ishida, R., Uchida, S.: Preselection of support vector candidates by relative neighborhood graph for large-scale character recognition. In: 2015 13th Int. Conf. on Document Analysis & Recog. (ICDAR). pp. 306–310 (2015)
7. Hacid, H., Yoshida, T.: Incremental neighborhood graphs construction for multidimensional databases indexing. In: Conference of the Canadian Society for Computational Studies of Intelligence. pp. 405–416. Springer (2007)
8. Han, D., Han, C., Yang, Y., Liu, Y., Mao, W.: Pre-extracting method for svm classification based on the non-parametric k-nn rule. In: 2008 19th International Conference on Pattern Recognition. pp. 1–4. IEEE (2008)
9. Jaromczyk, J.W., Toussaint, G.T.: Relative neighborhood graphs and their relatives. Proceedings of the IEEE **80**(9), 1502–1517 (1992)
10. Kirkpatrick, D.G., Radke, J.D.: A framework for computational morphology. In: TOUSSAINT, G.T. (ed.) Computational Geometry, Machine Intelligence and Pattern Recognition, vol. 2, pp. 217 – 248 (1985)
11. Rayar, F., Barrat, S., Bouali, F., Venturini, G.: An approximate proximity graph incremental construction for large image collections indexing. In: International Symposium on Methodologies for Intelligent Systems. pp. 59–68. Springer (2015)
12. Rayar, F., Barrat, S., Bouali, F., Venturini, G.: Incremental hierarchical indexing and visualisation of large image collections. In: 24th European Symposium on Artificial Neural Networks, ESANN 2016, Bruges, Belgium, April 27-29, 2016 (2016)
13. Rayar, F., Barrat, S., Bouali, F., Venturini, G.: A viewable indexing structure for the interactive exploration of dynamic and large image collections. ACM Transactions on Knowledge Discovery from Data (TKDD) **12**(1), 1–26 (2018)
14. Tellez, E.S., Ruiz, G., Chavez, E., Graff, M.: Local search methods for fast near neighbor search. arXiv preprint arXiv:1705.10351 (2017)
15. Toussaint, G.T.: The relative neighbourhood graph of a finite planar set. Pattern Recognition **12**(4), 261 – 268 (1980)
16. de Vries, N.J., Arefin, A.S., Mathieson, L., Lucas, B., Moscato, P.: Relative neighborhood graphs uncover the dynamics of social media engagement. In: Int. Conf. on Advanced Data Mining and App. pp. 283–297 (2016)

6 Supplementary Material

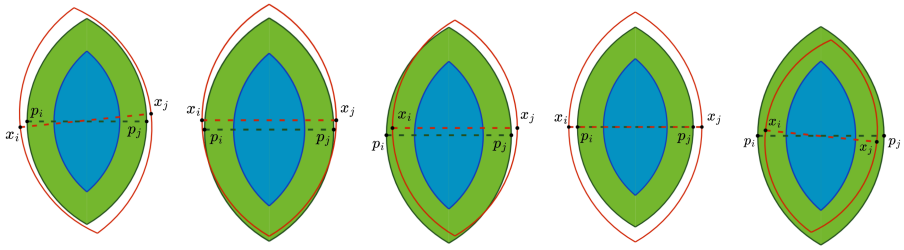


Fig. 10: A few examples illustrating the similarity between the lune (x_i, x_j) and the lune (p_i, p_j), where x_i and x_j are both fairly close to p_i and p_j, respectively, relative to $d(x_i, x_j)$ but otherwise unconstrained. Lune(x_i, x_j) is shown in red, lune(p_i, p_j) is shown in green, and generalized-lune(p_i, p_j) is shown in blue. Note how much smaller the generalized lune is compared to the RNG lunes.

Proposition 11. *Let Q, p_j, and p_k satisfy*

$$\begin{cases} d(Q, p_k) < d(Q, p_j) - r_j & \text{(19a)} \\ d(p_j, p_k) < d(Q, p_j) - 2r_j. & \text{(19b)} \end{cases}$$

Then, for all points x_j in the pivot domain of p_j of radius r_j, i.e., $d(x_j, p_j) \leq r_j$, we have

$$\max(d(p_k, Q), d(p_k, x_j)) < d(Q, x_j), \tag{20}$$

i.e., the pivot p_k prevents the formation of an RNG link between Q and all x_j, constituents of pivot p_j.

Proof. Apply Theorem 1 with $p_i = Q$, $p_j = p_j$ and $p_k = p_k$ with radii $r_i = r_Q = 0$. The conditions of the theorem is then

$$\begin{cases} d(p_k, Q) < d(Q, p_j) - (2r_Q + r_j) & \text{(21a)} \\ d(p_k, p_j) < d(Q, p_j) - (r_Q + 2r_j), & \text{(21b)} \end{cases}$$

which are Equations 19 and thus holds by assumption. The consequence of the theorem is then $\max(d(p_k, x_i), d(p_k, x_j)) < d(x_i, x_j)$, for any x_i and x_j in the pivot domains of Q and p_j, respectively. Since the only member of the pivot Q is Q, then $\max(d(p_k, Q), d(p_k, x_j)) < d(Q, x_j)$.

Proposition 12. *Consider an exemplar x_j and a pivot (p_i, r_i). Then if there is a pivot p_k satisfying*

$$\begin{cases} d(p_k, p_i) < d(p_i, x_j) - 2r_i & \text{(22a)} \\ d(p_k, x_j) < d(p_i, x_j) - r_i, & \text{(22b)} \end{cases}$$

then all Q in the pivot domain of p_i are prevented from forming an RNG link with x_j, i.e.,

$$\max\left(d\left(p_k, Q\right), d\left(p_k, x_j\right)\right) < d\left(Q, x_j\right). \tag{23}$$

Proof. Apply Theorem 1 with x_j as p_j with radii $r_j = 0$. Then, the condition of Theorem 1 is

$$\begin{cases} d\left(p_k, p_i\right) < d\left(p_i, x_j\right) - (2r_i + 0) & \text{(24a)} \\ d\left(p_k, x_j\right) < d\left(p_i, x_j\right) - (r_i + 0), & \text{(24b)} \end{cases}$$

which is the premise of the proposition. Then by Theorem 1 using Q as an exemplar in the pivot domain of p_i, and x_j as the sole exemplar in the pivot domain of x_j gives Equation 23.

Proposition 13. *In considering a potential link between the query Q and exemplar x_j, if a pivot p_k satisfy either one of the following*

$$\begin{cases} d\left(Q, p_k\right) - \delta_{\max}(p_k) \geq d\left(Q, x_j\right) & \text{(25a)} \\ d\left(x_j, p_k\right) - \delta_{\max}(p_k) \geq d\left(Q, x_j\right), & \text{(25b)} \end{cases}$$

where

$$\delta_{\max}(p_k) = \max_{\forall x_k, d(x_k, p_k) \leq r_k} d\left(p_k, x_k\right), \tag{26}$$

then none of the exemplars in the pivot domain of p_k can interfere with the formation of the RNG link (Q, x_j).

Proof. Let x_k be in the pivot domain of p_k. Then

$$\begin{cases} d\left(Q, x_k\right) \geq d\left(Q, p_k\right) - d(p_k, x_k) \geq d\left(Q, p_k\right) - \max\limits_{\forall x_k, d(x_k, p_k) \leq r_k} d\left(p_k, x_k\right) \geq d(Q, x_j) & \text{(27a)} \\ d\left(x_j, x_k\right) \geq d\left(x_j, p_k\right) - d(p_k, x_k) \geq d(x_j, p_k) - \max\limits_{\forall x_k, d(x_k, p_k) \leq r_k} d\left(p_k, x_k\right) \geq d(Q, x_j), & \text{(27b)} \end{cases}$$

or $\max\left(d\left(Q, x_k\right), d\left(x_j, x_k\right)\right) \geq d\left(Q, x_j\right)$, which puts x_k outside the lune(Q, x_j).

Proposition 14. *In considering the RNG link between a query Q and exemplar x_j, an exemplar x_k in the pivot domain of p_k satisfying either one of the following inequalities*

$$\begin{cases} d\left(Q, p_k\right) - d\left(x_k, p_k\right) \geq d\left(Q, x_j\right) & \text{(28a)} \\ d\left(x_j, p_k\right) - d\left(x_k, p_k\right) \geq d\left(Q, x_j\right), & \text{(28b)} \end{cases}$$

does not interfere with the formation of the RNG link.

Proof.

$$\begin{cases} d\left(Q, x_k\right) \geq d\left(Q, p_k\right) - d\left(x_k, p_k\right) \geq d(Q, x_j) & \text{(29a)} \\ d\left(x_j, x_k\right) \geq d\left(x_j, p_k\right) - d\left(x_k, p_k\right) \geq d(Q, x_j), & \text{(29b)} \end{cases}$$

or $\max\left(d\left(Q, x_k\right), d\left(x_j, x_k\right)\right) \geq d\left(Q, x_j\right)$, which puts x_k outside the lune(Q, x_j).

Proposition 15. *A query Q will not invalidate RNG links ending at x_i if $d(Q,x_i) \geq \bar{\mu}_{\max}(x_i)$,*

$$\bar{\mu}_{\max}(x_i) = \max_{x_j \in RNG \text{ } neighbors \text{ } of x_i} d(x_i, x_j). \tag{30}$$

Proof. The query Q lies outside lune(x_i, x_j) because

$$\max(d(Q,x_i), d(Q,x_j)) \geq d(Q,x_i) \geq \bar{\mu}_{\max}(x_i) \geq d(x_i,x_j). \tag{31}$$

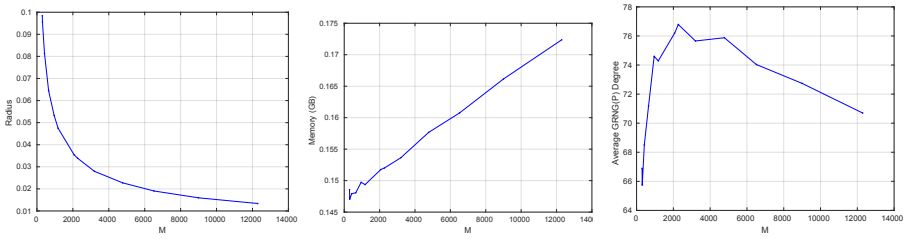
Proof. The query Q lies outside lune(x_i, x_j) for any RNG neighbors of x_i because

$$\max(d(Q,x_i), d(Q,x_j)) \geq d(Q,x_i) \geq \bar{\mu}_{\max}(x_i) \geq d(x_i,x_j). \tag{32}$$

Similarly,

$$d(Q,x_i) \geq d(Q,p_i) - d(p_i,x_i) \geq \mu_{\max}(p_i) - d(p_i,x_i) \geq \bar{\mu}_{\max}(x_i) + d(x_i,p_i) - d(p_i,x_i) \geq \bar{\mu}_{\max}(x_i). \tag{33}$$

Now by the first proposition Q does not invalidate RNG links at x_i.



(a) (b) (c)

Fig. 11: More stats from Figure 12 (a)Radius Giving number of pivots. (b) memory use, (c) average degree of GRNG.

.

Proposition 16. *The query Q does not form GRNG links with any children $(\bar{p}_i, \bar{r}_i)$ of coarse-scale pivots (p_i, r_i) that it does not form a GRNG link with.*

Proof. The condition that (Q, r_Q) is not a GRNG neighbor of (p_i, r_i) means that $\exists p_k$ coarse-scale pivot such that

$$\begin{cases} d(Q,p_k) < d(Q,p_i) - (2\bar{r}_Q + r_i) & \text{(34a)} \\ d(p_i,p_k) < d(Q,p_i) - (\bar{r}_Q + 2r_i). & \text{(34b)} \end{cases}$$

Since the GRNG of coarse-scale pivots does not include Q, this condition must be explicitly tested, *i.e.*, Q should be added (at least virtually), to the GRNG$(\mathcal{P})$. When this is completed, by Theorem 2, since (Q, r_Q) does not form a GRNG link with (p_i, r_i) then its children, *i.e.*, (Q, r_Q) itself cannot form GRNG links with $(\bar{p}_{\bar{i}}, \bar{r}_{\bar{i}}) \in \mathcal{D}(p_i, r_i)$.

Table 2: Average number of distances computations for finding the RNG Neighbors of 100 queries in an optimal 2-Layer GRNG Hierarchy.

N	2D	3D	4D	5D	6D
3,200	438.62	962.96	1,280.44	1,574.04	2,060.78
6,400	635.36	1,603.13	2,181.42	2,593.77	3,404.45
12,800	899.11	2,111.74	4,046.77	4,423.54	5,609.44
25,600	1,335.19	3,204.82	6,601.08	7,154.21	9,382.17
51,200	1,812.09	4,166.68	10,826.40	11,937.50	15,737.00
102,400	2,602.51	5,956.77	16,239.10	20,612.60	27,001.40
204,800	3,672.50	8,139.86	21,761.10	34,464.20	47,338.70
409,600	5,183.30	11,352.70	31,196.90	56,880.80	
819,200	7,379.43	15,616.50	42,211.30		
1,638,400	10,543.70	21,802.80			
3,276,800	14,864.20	30,049.30			

Table 3: Ratio between dataset size N and average number of search distances. Represents the savings.

N	2D	3D	4D	5D	6D
3,200	7.30	3.32	2.50	2.03	1.55
6,400	10.07	3.99	2.93	2.47	1.88
12,800	14.24	6.06	3.16	2.89	2.28
25,600	19.17	7.99	3.88	3.58	2.73
51,200	28.25	12.29	4.73	4.29	3.25
102,400	39.35	17.19	6.31	4.97	3.79
204,800	55.77	25.16	9.41	5.94	4.33
409,600	79.02	36.08	13.13	7.20	
819,200	111.01	52.46	19.41		
1,638,400	155.39	75.15			
3,276,800	220.45	109.05			

Table 4: Average time (ms) for finding the RNG Neighbors of 100 queries in an optimal 2-Layer GRNG Hierarchy.

N	2D	3D	4D	5D	6D
3,200	0.412	0.861	2.068	2.852	6.066
6,400	0.638	1.245	3.408	6.562	36.385
12,800	0.915	2.195	10.252	9.988	35.918
25,600	1.796	4.131	10.521	39.948	55.921
51,200	2.780	5.084	18.185	61.121	110.866
102,400	5.214	9.062	31.183	104.685	203.889
204,800	7.490	14.185	49.818	188.756	419.997
409,600	13.933	20.437	77.360	293.748	
819,200	22.334	33.326	137.406		
1,638,400	39.275	42.544			
3,276,800	96.146	78.418			

Table 5: Time (hr) to incrementally construct the 2-Layer GRNG Index for uniformly distributed data.

N	2D	3D	4D	5D	6D
3,200	2.631E-04	8.808E-04	2.626E-03	1.028E-02	1.816E-02
6,400	7.207E-04	2.346E-03	8.770E-03	1.952E-02	0.134
12,800	1.884E-03	7.390E-03	4.213E-02	4.720E-02	0.214
25,600	6.581E-03	2.496E-02	7.253E-02	0.218	0.558
51,200	1.928E-02	6.775E-02	0.246	0.583	1.538
102,400	7.37E-02	0.180	0.718	2.195	4.434
204,800	0.195	0.555	2.198	6.155	16.497
409,600	0.678	1.424	6.254	18.583	
819,200	2.101	4.338	18.445		
1,638,400	7.381	11.439			
3,276,800	34.513	41.779			

Table 6: Total distance computations to incrementally construct the 2-Layer GRNG Index for uniformly distributed data.

N	2D	3D	4D	5D	6D
3,200	1,930,920	10,067,708	10,473,370	4,991,140	5,620,782
6,400	5,331,485	30,859,620	42,420,322	15,924,752	21,869,207
12,800	13,406,458	93,836,003	175,916,012	47,096,101	69,837,894
25,600	34,086,261	247,336,100	575,258,091	206,262,745	228,199,077
51,200	91,931,558	653,653,274	1,757,804,446	752,666,633	727,327,667
102,400	243,241,773	1,648,937,181	4,901,713,631	2,723,494,366	2,311,395,377
204,800	653,721,994	4,003,268,824	13,049,444,992	8,920,116,049	7,446,547,400
409,600	1,773,737,263	9,644,102,006	33,181,379,037	27,996,266,271	
819,200	4,796,775,610	23,309,124,834	81,589,547,397		
1,638,400	13,341,696,766	57,073,630,261			
3,276,800	36,741,205,495	139,223,206,018			

Table 7: Memory use (GB) during the incremental construction of the 2-Layer GRNG Hierarchy.

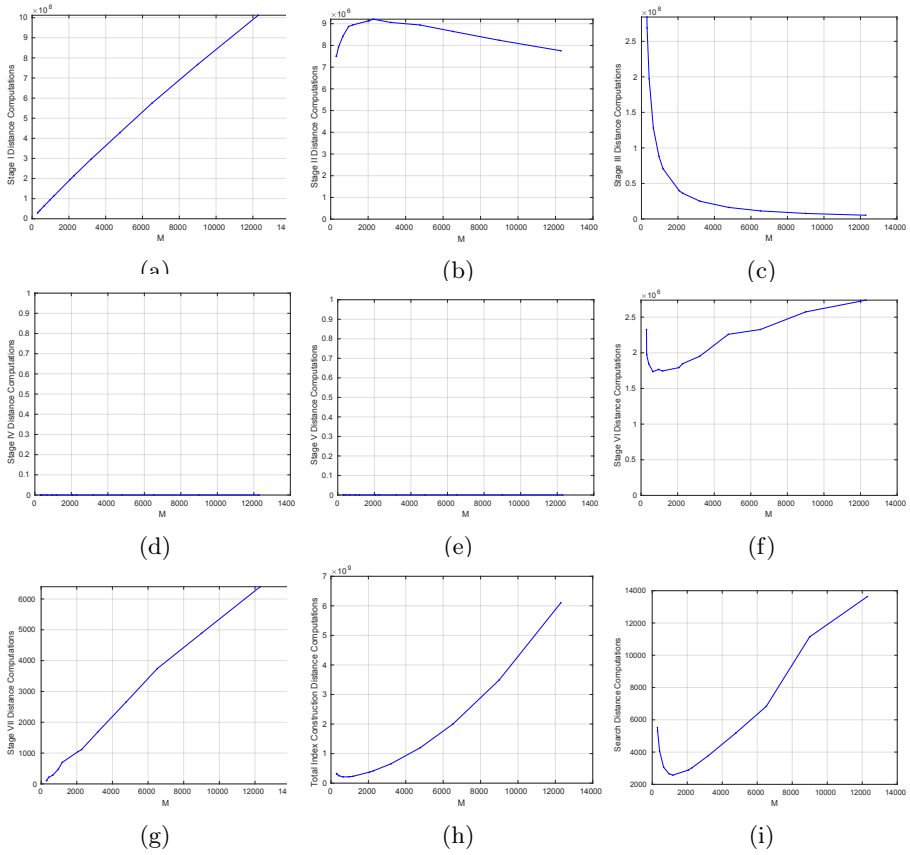
N	2D	3D	4D	5D	6D
3,200	7.931E-03	1.199E-02	2.223E-02	3.510E-02	6.544E-02
6,400	1.330E-02	2.141E-02	4.180E-02	7.115E-02	0.152
12,800	2.345E-02	3.904E-02	8.70E-02	0.147	0.306
25,600	4.389E-02	7.50E-02	1.76E-01	0.318	0.684
51,200	7.663E-02	1.46E-01	0.347	0.690	1.487
102,400	1.50E-01	0.291	0.712	1.532	3.297
204,800	0.296	0.576	1.473	3.273	7.175
409,600	0.591	1.153	3.010	7.083	
819,200	1.174	2.326	6.180		
1,638,400	2.362	4.738			
3,276,800	4.709	10.172			

Table 8: Average out degree of the GRNG for optimal 2-Layer GRNG Hierarchy on data of uniform distribution.

N	2D	3D	4D	5D	6D
3,200	57.95	225.70	537.90	755.87	1,215.23
6,400	64.19	258.39	683.46	955.18	1,799.90
12,800	66.55	303.87	884.89	1,122.68	2,300.27
25,600	67.53	338.29	1,060.78	1,633.40	2,954.72
51,200	72.16	378.26	1,263.93	2,099.24	3,623.84
102,400	75.10	412.30	1,466.11	2,705.52	4,367.13
204,800	77.57	442.66	1,681.82	3,305.78	5,209.58
409,600	80.12	473.60	1,905.11	4,008.62	
819,200	80.85	500.25	3,305.78		
1,638,400	83.44	528.95			
3,276,800	83.88	551.76			

Table 9: Optimal number of pivots for optimal 2-Layer GRNG Hierarchies in data of uniform distribution.

N	2D	3D	4D	5D	6D
3,200	236	466	611	763	1,217
6,400	342	680	861	967	1,804
12,800	468	1,012	1,282	1,139	2,307
25,600	642	1,440	1,795	1,699	2,970
51,200	917	2,079	2,552	2,251	3,653
102,400	1,281	2,997	3,573	3,032	4,417
204,800	1,784	4,255	5,013	3,922	5,293
409,600	2,499	6,010	6,998	5,085	
819,200	3,416	8,541	9,705		
1,638,400	4,823	12,137			
3,276,800	6,610	17,091			



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 12: Analyzing the cost, in distances computations, of each stage as a function of number of pivots, $M = |\mathcal{P}|$, for 2D uniformly distributed exemplar of N=102,400. (a-g) Stages I-VII, (h) Total Index Construction, (i) Average Search.

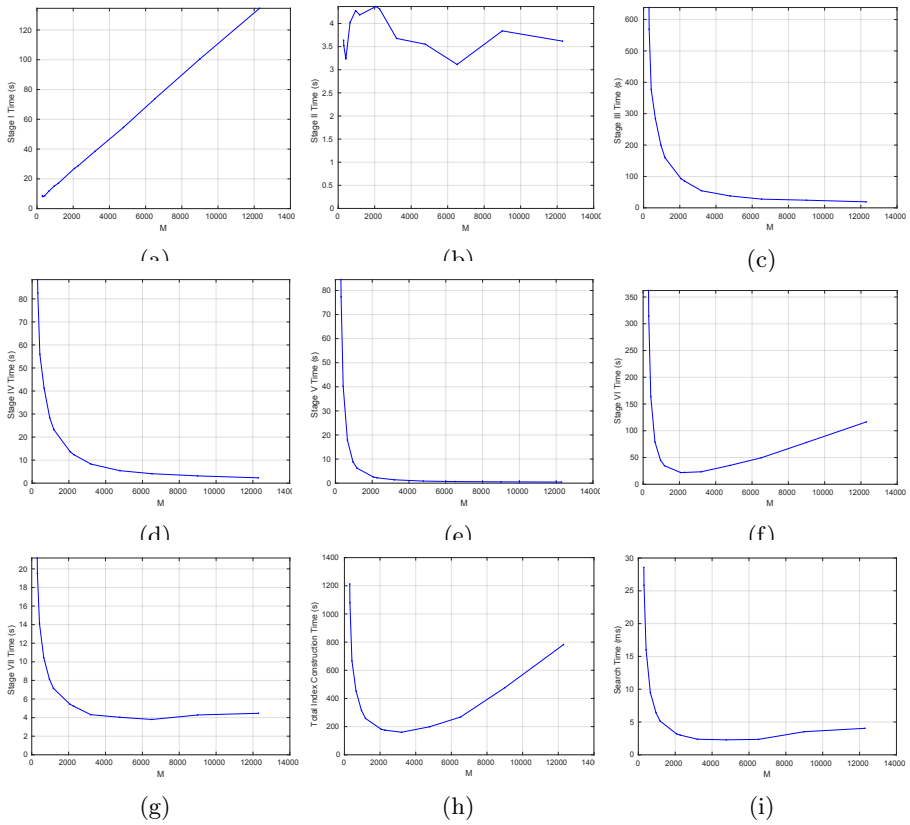


Fig. 13: Analyzing the cost, in time, of each stage as a function of number of pivots, $M = |\mathcal{P}|$, for 2D uniformly distributed exemplar of N=102,400. (a-g) Stages I-VII, (h) Total Index Construction, (i) Average Search.