Translation Synchronization via Truncated Least Squares

Xiangru Huang
The University of Texas at Austin
2317 Speedway, Austin, 78712
xrhuang@cs.utexas.edu

Zhenxiao Liang
Tsinghua University
Beijing, China, 100084
liangzx14@mails.tsinghua.edu.cn

Chandrajit Bajaj
The University of Texas at Austin
2317 Speedway, Austin, 78712
bajaj@cs.utexas.edu

Qixing Huang
The University of Texas at Austin
2317 Speedway, Austin, 78712
huangqx@cs.utexas.edu

Abstract

In this paper, we introduce a robust algorithm, *TranSync*, for the 1D translation synchronization problem, in which the aim is to recover the global coordinates of a set of nodes from noisy measurements of relative coordinates along an observation graph. The basic idea of TranSync is to apply truncated least squares, where the solution at each step is used to gradually prune out noisy measurements. We analyze TranSync under both deterministic and randomized noisy models, demonstrating its robustness and stability. Experimental results on synthetic and real datasets show that TranSync is superior to state-of-the-art convex formulations in terms of both efficiency and accuracy.

1 Introduction

In this paper, we are interested in solving the 1D translation synchronization problem, where the input is encoded as an observation graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with n nodes (i.e. $\mathcal{V} = \{1, \cdots, n\}$). Each node is associated with a latent coordinate $x_i^\star \in \mathbb{R}, 1 \leq i \leq n$, and each edge $(i, j) \in \mathcal{E}$ is associated with a noisy measurement $t_{ij} = x_i^\star - x_j^\star + \mathcal{N}(\epsilon_{ij})$ of the coordinate difference $x_i - x_j$ under some noise model $\mathcal{N}(\epsilon_{ij})$. The goal of translation synchronization is to recover the latent coordinates (up to a global shift) from these noisy pairwise measurements. Translation synchronization is a fundamental problem that arises in many application domains, including joint alignment of point clouds [7] and ranking from relative comparisons [8, 16].

A standard approach to translation synchronization is to solve the following linear program:

$$\text{minimize} \sum_{(i,j) \in \mathcal{E}} |t_{ij} - (x_i - x_j)|, \quad \text{subject to} \quad \sum_{i=1}^n x_i = 0, \tag{1}$$

Where the constraint ensures that the solution is unique. The major drawback of the linear programming formulation is that it can only tolerate up to 50% of measurements coming from biased noise models (e.g., uniform samples with non-zero mean). Moreover, it is challenging to solve (1) efficiently at scale. Solving (1) using interior point method becomes impractical for large-scale datasets, while more scalable methods such as coordinate descent usually exhibit slow convergence.

In this paper, we introduce a robust and scalable algorithm, TranSync, for translation synchronization. The algorithm is rather simple, we solve a truncated least squares at each iteration k:

$$\{x_i^{(k)}\} = \underset{\{x_i\}}{\operatorname{argmin}} \sum_{(i,j)\in\mathcal{E}} w_{ij}|t_{ij} - (x_i - x_j)|^2, \quad \text{subject to} \quad \sum_{i=1}^{n} \sqrt{d_i}x_i = 0, \quad d_i := \sum_{j\in\mathcal{N}(i)} w_{ij}. \tag{2}$$

where the weights $w_{ij} = Id(|t_{ij} - (x_i^{(k-1)} - x_j^{(k-1)})| < \delta_k)$ are obtained from the solution at the previous iteration using a geometrically decaying truncation parameter δ_k. Although TranSync requires solving a linear system at each step, these linear systems are fairly similar to each other, meaning that the solution at the previous iteration provides an excellent warm-start for solving the linear system at the current iteration. As a result, the computational efficiency of TranSync is superior to state-of-the-art methods for solving the linear programming formulation. We analyze TranSync under both deterministic and randomized noise models, demonstrating its robustness and stability. In particular, we show that TranSync is able to handle biased noisy measurements.

We have evaluated TranSync on both synthetic datasets and real datasets used in the applications of joint alignment of point clouds and ranking from pair-wise measurements. Experimental results show that TranSync is superior to state-of-the-art solvers for the linear programming formulation in terms of both computational efficiency and accuracy.

1.1 Related Work

Translation synchronization falls into the general problem of map synchronization, which takes maps computed between pairs of objects as input, and outputs consistent maps across all the objects. Map synchronization appears as a crucial step in many scientific problems, including fusing partially overlapping range scans [15], assembling fractured surfaces [14], solving jigsaw puzzles [5, 11], multi-view structure from motion [25], data-driven shape analysis and processing [17], and structure from motion [27].

Early methods for map synchronization focused on applying greedy algorithms [14, 15, 18] or combinatorial optimization [20, 23, 27]. Although these methods exhibit certain empirical success, they lack theoretical understanding (e.g. we do not know under what conditions can the underlying ground-truth maps be exactly recovered).

Recent methods for map synchronization apply modern optimization techniques such as convex optimization and non-convex optimization. In [13], Huang and Guibas introduce a semidefinite programming formulation for permutation synchronization and its variants. Chen et al. [4] generalize the method to partial maps. In [26], Wang and Singer introduce a method for rotation synchronization. Although these methods provide tight, exact recovery conditions, the computational cost of the convex optimizations provide an obstruction for applying these methods to large-scale data sets.

In contrast to convex optimization, very recent map synchronization methods leverage non-convex optimization approaches such as spectral techniques and gradient-based optimization. In [21, 22], Pachauri et al. study map synchronization from the perspective of spectral decomposition. Recently, Shen et al. [24] provide an analysis of spectral techniques for permutation synchronization. Beyond spectral techniques, Zhou et al. [28] apply alternating minimization for permutation synchronization. Finally, Chen and Candes [3] introduce a method for the generalized permutation synchronization problem using the projected power method. To the best of our knowledge, we are the first to develop and analyze continuous map synchronizations (e.g., translations or rotations) beyond convex optimization.

Our approach can be considered as a special case of reweighted least squares (or RLS) [9, 12], which is a powerful method for solving convex and non-convex optimizations. The general RLS framework has been applied for map synchronization (e.g. see [1, 2]). Despite the empirical success of these approaches, the theoretical understanding of RLS remains rather limited. The analysis in this paper provides a first step towards the understanding of RLS for map synchronization.

1.2 Notation

Before proceeding to the technical part of this paper, we introduce some notation that will be used later. The unnormalized graph Laplacian of a graph G is denoted as L_G. If it is obvious from the

---

Algorithm 1 TranSync($c, k_{\max}$)

---

1. $\mathbf{x}^{(-1)} \leftarrow \mathbf{0}$. $\delta_{-1} \leftarrow \infty$.
for $k = 0, 1, 2, k_{\max}$ **do**
   2. Obtain the truncated graph $\mathcal{G}^{(k)}$ using $\mathbf{x}^{(k-1)}$ and δ_{k-1}.
   3. **Break if** $\mathcal{G}^{(k)}$ is disconnected
  if $\mathcal{G}$ is non-bipartite **then**
   4. Solve (2) using (3) to obtain $\mathbf{x}^{(k)}$.
  else
   5. Solve (2) using conjugate gradient to obtain $\mathbf{x}^{(k)}$.
  end if.
   6. $\delta_k = \min \left(\max\limits_{(i,j) \in \mathcal{E}} |t_{ij} - (x_i^{(0)} - x_j^{(0)})|, c\delta_{k-1} \right)$.
end for
Output: $\mathbf{x}^{(k)}$.

---

context, we will always shorten L_G as L to make the notation uncluttered. Similarly, we will use $D = \mathrm{diag}(d_1, \cdots, d_n)$ to collect the vertex degrees and denote the vertex adjacency and vertex-edge adjacency matrices as A and B respectively. The peusdo-inverse of a matrix X is given by X^+. In addition, we always sort the eigenvalues of a symmetric matrix $X \in \mathbb{R}^{n \times n}$ in increasing order (i.e. $\lambda_1(X) \le \lambda_2(X) \le \cdots \le \lambda_n(X)$). Moreover, we will consider several matrix norms $\| \cdot \|, \| \cdot \|_{1,\infty}$ and $\| \cdot \|_{\mathcal{F}}$, which are defined as follows:

$$\|X\| = \sigma_{\max}(X), \quad \|X\|_{1,\infty} = \max_{1 \le i \le n} \sum_{j=1}^{n} |x_{ij}|, \quad \|X\|_{\mathcal{F}} = \Big(\sum_{i,j} x_{ij}^2 \Big)^{\frac{1}{2}}.$$

Note that $\|X\|_{1,\infty}$ is consistent with the L^∞-norm of vectors.

2 Algorithm

In this section, we provide the algorithmic details of TranSync. The iterative scheme (1) requires an initial solution $\mathbf{x}^{(0)}$, an initial truncation parameter δ_0, and a stopping condition. The initial solution can be determined by solving for $\mathbf{x}^{(0)}$ from (2) w.r.t. $w_{ij} = 1$. We set the initial truncation parameter $\delta_0 = \max\limits_{(i,j) \in \mathcal{E}} |t_{ij} - (x_i^{(0)} - x_j^{(0)})|$, so that the edge with the biggest residual is removed. We stop TranSync either after the maximum number of iterations is reached or the truncated graph becomes disconnected. Algorithm 1 provides the pseudo code of TranSync.

Clearly, the performance of TranSync is driven by the efficiency of solving (2) at each iteration. TranSync takes an iterative approach, in which we utilize a warm-start $\mathbf{x}^{(k-1)}$ provided by the solution obtained at the previous iteration. When the truncated graph is non-bipartite, we find a simple weighted average scheme delivers satisfactory computational efficiency. Specifically, it generates a series of vectors $\mathbf{x}^{k,0} = \mathbf{x}^{(k-1)}, \mathbf{x}^{k,1}, \cdots, \mathbf{x}^{k,n_{\max}}$ via the following recursion:

$$\overline{x}_i^{k,l+1} = \sum_{j \in \mathcal{N}(i)} w_{ij}(x_j^{k,l} + t_{ij}) / \sum_{j \in \mathcal{N}(i)} w_{ij}, \qquad x_i^{k,l+1} = \overline{x}_i^{k,l+1} - \frac{1}{\sum_{i'=1}^{n} \sqrt{d_i}} \sum_{i'=1}^{n} \sqrt{d_{i'}} \overline{x}_{i'}^{k,l+1}, \tag{3}$$

which may be written in the following matrix form:

$$\mathbf{x}^{k,l+1} = \Big(I_n - \frac{1}{n} D^{-\frac{1}{2}} \mathbf{1} \mathbf{1}^T D^{\frac{1}{2}}\Big) D^{-1} \big(A\mathbf{x}^{k,l} + B\mathbf{t}^{(k)}\big), \tag{4}$$

Remark 2.1 *The corresponding normalization constraint in (3), i.e., $\sum\limits_{i} \sqrt{d_i} x_i = 0$, only changes the solution to (2) by a constant factor. We utilize this modification for the purpose of obtaining a concise convergence property of the iterative scheme detailed below.*

The following proposition states that (3) admits a geometric convergence rate:

3

Proposition 2.1 $\boldsymbol{x}^{k,l}$ *geometrically converges to* $\boldsymbol{x}^{(k+1)}$. *Specifically,* $\forall l \geq 0$,

$$\|D^{\frac{1}{2}}\big(\boldsymbol{x}^{k,l} - \boldsymbol{x}^{(k)}_{\mathrm{shift}}\big)\| \leq (1 - \rho)^l \|D^{\frac{1}{2}}\big(\boldsymbol{x}^{k,0} - \boldsymbol{x}^{(k)}_{\mathrm{shift}}\big)\|, \quad \boldsymbol{x}^{(k)}_{\mathrm{shift}} = \boldsymbol{x}^{(k)} - \frac{\sum_i \sqrt{d_i} x_i^{(k)}}{\sum_i \sqrt{d_i}} \boldsymbol{1}.$$

where $\rho < 1$ *is the spectral gap of the normalized Graph Laplacian of the truncated graph.*

Proof. The proof is straightforward from (4) and is omitted for brevity.

Since the intermediate solutions are mainly used to prune outlier observations, it is clear that $O(\log(n))$ iterations of (4), which induce a $O(1/n)$ error for solving (2), are sufficient. The complexity of checking if the graph is non-bapriatite is $O(|\mathcal{E}|)$. The total running time for solving (2) is thus $O\big(|\mathcal{E}| \log(n)\big)$. This means the total running time of TranSync is $O(|\mathcal{E}| \log(n) k_{\max})$, making it scalable to large-scale datasets.

3 Analysis of TranSync

In this section, we provide exact recovery conditions of TranSync. We begin with describing an exact recovery condition under a deterministic noise model in Section 3.1. We then study an exact recovery condition to demonstrate that TranSync can handle biased noisy samples in Section 3.2.

3.1 Deterministic Exact Recovery Condition

We consider the following deterministic noisy model: We are given the ground-truth location $\mathbf{x}^{gt}$. Then, for each correct measurement $t_{ij}, (i, j) \in \mathcal{G}$, $|t_{ij} - (x_i^{gt} - x_j^{gt})| \leq \sigma$ for a threshold σ. In contrast, each incorrect measurement $t_{ij}, (i, j) \in \mathcal{G}$ could take any real number. The following theorem provides an exact recovery condition under this noisy model.

Theorem 3.1 *Let* d_{bad} *be the maximum number of incorrect measurements per node. Define*

$$\alpha = \max_k L^\dagger_{G,kk} + \max_{i \neq j} L^\dagger_{G,ij} + \frac{n}{2} \max_{\substack{i,j,k \\ \text{pairwisely different}}} |L^\dagger_{G,ki} - L^\dagger_{G,kj}|,$$

and

$$h = \alpha d_{bad}, \quad p = \frac{d_{bad}\alpha}{1 - 2h}, \quad q = \frac{(n - d_{bad})\alpha}{1 - 2h}.$$

Suppose $h < \frac{1}{6}$ *(or* $p < \frac{1}{4}$*), then starting from any initial solution* $\boldsymbol{x}^{(0)}$, *and for any large enough initial truncation threshold* $\epsilon \geq 2\|\boldsymbol{x}^{(0)}\|_\infty + \sigma$ *and iterative step size* c *satisfying* $4p < c < 1$, *we have*

$$\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{gt}\|_\infty \leq q\sigma + 2p\epsilon c^{k-1},$$

where

$$k \leq -\log\left(\frac{\epsilon(c - 4p)}{(1 + 2q)\sigma}\right) / \log c + 1.$$

Moreover, we can eventually reach an $x^{(k)}$ *such that*

$$\|\boldsymbol{x}^{(k)}\|_\infty \leq \frac{2p + cq}{c - 4p}\sigma$$

which is independent of the initial solution $\boldsymbol{x}^{(0)}$, *initial truncation threshold* ϵ, *and values of all wrong measurements* $\boldsymbol{t}_{G\backslash G_{good}}$.

Proof: See Appendix A. □

Theorem 3.1 essentially says that TransSync can tolerate a constant fraction of arbitrary noise. To understand how strong this condition is, we consider the case where $G = \mathcal{K}_n$ is given by a clique. Moreover, we assume the nodes are divided into two clusters of equal size, where all the measurements within each cluster are correct. For measurements between different clusters, half of them are correct and the other half are wrong. In this case, 25% of all measurements are wrong. However, we cannot recover the original $\mathbf{x}^{gt}$ in this case. In fact, we can set the wrong measurements in a consistent

manner, i.e $t_{ij} = x_i^{gt} - x_j^{gt} + b$ for a constant $b \neq 0$, leading to two competing clusters (one correct and the other one incorrect) with equal strength. Hence, in the worst case, any algorithm can only tolerate at most 25% of measurements being wrong.

We now try to use Theorem 3.1 to analyze the case where the observation graph is a clique. In this case, it is clear that $\alpha = \frac{1}{n}$, and $p = \frac{d_{bad}}{n}$, i.e the fraction of wrong measurements out of all measurements. Hence, in the clique case, we have shown that TranSync converges to a neighborhood of the ground truth from any initial solution if the fraction of wrong measurements is less that $\frac{1}{6}$ (i.e., 2/3 of the upper bound).

3.2 Biased Random Noisy Model

We proceed to provide an exact recovery condition of TranSync under a biased random noisy model. To simplify the discussion, we assume the observation graph $\mathcal{G} = \mathcal{K}_n$ is a clique. However, our analysis framework can be extended to handle arbitrary graphs.

Assume $\sigma << a + b$. We consider the following noise model, where the noisy measurements are independent, and they follow

$$t_{ij} = \begin{cases} x_i^{gt} - x_j^{gt} + U[-\sigma, \sigma] & \text{with probability } p \\ x_i^{gt} - x_j^{gt} + U[-a, b] & \text{with probability } 1 - p \end{cases} \quad (5)$$

It is easy to check that the linear programming formulation is unable to recover the ground-truth solution if $\frac{b}{a+b}(1 - p) > \frac{1}{2}$. The following theorem shows that TranSync achieves a sub-constant recovery rate instead.

Theorem 3.2 *There exists a constant c so that if $p > c/\sqrt{\log(n)}$, then w.h.p,*

$$\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{gt}\|_\infty \leq (1 - p/2)^k (b - a), \qquad \forall\, k = 0, \cdots, [-\log(\frac{b+a}{2\sigma})/log(1 - p/2)].$$

The major difficulty of proving Theorem 3.2 is that $\mathbf{x}^{(k)}$ is dependent on $\mathbf{t}_k$, making it hard to control $\mathbf{x}^{(k)}$ using existing concentration bounds. We address this issue by showing that the solutions $\mathbf{x}^{(k)}, k = 0, \cdots$, stay close to the segment between x^{gt} and $x^{gt} + (1 - p)\frac{a+b}{2}\mathbf{1}$. Specifically, for points on this segment, we can leverage the independence of t_{ij} to derive the following concentration bound for one step of TranSync:

Lemma 3.1 *Consider a fixed observation graph $\mathcal{G}$. Let $\overline{r} = \frac{(a+b)p}{(a+b)p+2(1-p)\delta}$ and $d_{\min}$ be the minimum degree of $\mathcal{G}$. Suppose $d_{\min} = \Omega(log^2(n))$, and $p + \overline{r}(1 - p) = \Omega(\log^2(n)/d_{\min})$. Consider an initial point $\boldsymbol{x}^{(0)}$ (independent from t_{ij}) and a threshold parameter δ such that $-a + \delta \leq \min_i x_i^{(0)} \leq \max_i x_i^{(0)} \leq b - \delta$. Then w.h.p., one step of TranSync outputs $\boldsymbol{x}^{(1)}$ which satisfies*

$$\|\boldsymbol{x}^{(1)} - (1-\overline{r})\boldsymbol{x}^{(0)} + \overline{r}\boldsymbol{x}^{gt}\|_\infty = O\left(\sqrt{\frac{\log(n)}{(p + \overline{r}(1-p))d_{\min}\lambda_2(\overline{L}_\mathcal{G})}}\right) \cdot \sqrt{\max(\|\boldsymbol{x}^{(0)}\|_{d,\infty}^2, \overline{r}^2)} + O\left(\frac{p}{\overline{r}}\sigma^2\right),$$

where $\|\boldsymbol{x}^{(0)}\|_{d,\infty} = \max\limits_{1 \leq i,j \leq n} |x_i^{(0)} - x_j^{(0)}|$, and $\overline{L}_\mathcal{G}$ is the normalized graph Laplacian of $\mathcal{G}$.

Proof: See Appendix B.1. □

Remark 3.1 *Note that when $\mathcal{G}$ is a clique or a graph sampled from the standard Erdős-Rényi model $G(n, q)$, then $O(\sqrt{\frac{\rho \log(n)}{(p+\overline{r}(1-p))\lambda_2(L_\mathcal{G})}}) = O(\sqrt{\frac{\log(n)}{(p+\overline{r}(1-p))n}})$.*

To prove Theorem 3.2, we show that when $\overline{k} = O(\log^{\frac{3}{4}}(n))$, the L^∞ distance between $\mathbf{x}^{(k)}$ to the line segment between x^{gt} and $x^{gt} + (1 - p)\frac{a+b}{2}\mathbf{1}$ only grows geometrically, and this distance is in the order of $o(p)$. On the other hand, $(1 - p/2)^{\overline{k}} = o(p)$. So when $k \geq \overline{k}$, that distance decays with a geometrical rate that is small than c. The details are deferred to Appendix B.2.

Improving recovery rate via sample splitting. Note that Lemma 3.1 enables us to apply standard sampling tricks to improve the recovery rate. To simplify the discussion, we will assume σ is sufficiently small. First of all, it is clear that if re-sampling is allowed at each iteration, then TranSync admits a recovery rate of $O(\frac{\log(n)}{\sqrt{d_{min}}})$. When re-sampling is not allowed, we can improve the recovery rate by dividing the observations into $O(\frac{\log(n)}{\sqrt{n}})$ independent sets, and apply one set of observations at each iteration. In this case, the recovery rate is $O(\frac{\log^2(n)}{\sqrt{n}})$. These recovery rates suggest that the recovery rate in Theorem 3.2 could potentially be improved. Nevertheless, Theorem 3.2 still shows that TranSync can tolerate a sub-constant recovery rate, which is superior to the linear programming formulation.

4   Experimental Results

In this section, we provide a detailed experimental evaluation of the proposed translation synchronization (TranSync) method. We begin with describing the experimental setup in Section 4.1. We then perform evaluations on synthetic and real datasets in Section 4.2 and Section 4.3 respectively.

4.1   Experimental Setup

Datasets. We employ both synthetic datasets and real datasets for evaluation. The synthetic data is generated following the noisy model described in (5). In the following, we encode the noisy model as $\mathcal{M}(\mathcal{G}, p, \sigma)$, where $\mathcal{G}$ is the observation graph, p is the fraction of correct measurements, and σ describes the interval of correct measurements. Besides the synthetic data, we also consider two real datasets coming from the applications of joint alignment of point clouds and global ranking from relative rankings.

Baseline comparison. We choose coordinate descent for solving (1) as the baseline algorithm. Specifically, denote the solution of $x_i, 1 \leq i \leq n$ at iteration k as $x_i^{(k)}$. Then $\{x_i^{(k)}\}$ are given by the following recursion:

$$x_i^{(k)} = \arg\min_{x_i} \sum_{j \in \mathcal{N}(i)} |x_i - (x_j^{(k-1)} - t_{ij})|$$
$$= \underset{j \in \mathcal{N}(i)}{\text{median}}\{x_j^{(k-1)} - t_{ij}\}, \qquad 1 \leq i \leq n, \qquad k = 1, 2, \cdots, \tag{6}$$

We use the same initial starting point as TranSync. We also tested interior point methods, and all the datasets used in our experiments are beyond their reach.

Evaluation protocol. We report the min, median, and max of the coordinate-wise difference between the solution of each algorithm and the underlying ground-truth. We also report the total running time of each algorithm on each dataset (See Table 1).

4.2   Experimental Evaluation on Synthetic Datasets

We generate the synthetic datasets by sampling from four kinds of observation graphs and two values of σ, i.e. $\sigma \in \{0.01, 0.04\}$. The graphs are generated according to two modes: 1) *dense* graphs versus *sparse* graphs, and 2) *regular* graphs versus *irregular* graphs. To illustrate the strength of TranSync, we choose $p \in \{0.4, 0.8\}$ for dense graphs and $p \in \{0.8, 1.0\}$ for sparse graphs. Below is a detailed descriptions for all kinds of observation graphs generated.

- G_{dr} (dense, regular): The first graph contains $n = 2000$ nodes. Independently, we connect an edge between a pair of vertices v_i, v_j with a fixed probability $p = 0.1$. The expected degree of each vertex is 200.

- G_{di} (dense, irregular): The second graph contains $n = 2000$ nodes. Independently, we connect an edge between a pair of vertices v_i, v_j with probability $p = 0.4s_is_j$, where $s_i = 0.2 + 0.6\frac{i-1}{n-1}, 1 \leq i \leq n$ are scalar values associated the vertices. The expected degree of each vertex is about 200.

			Coordinate Descent				TranSync			
Graph	p	σ	min	median	max	time	min	median	max	time
G_{dr}	0.4	0.01	0.95e-2	1.28e-2	11.40e-2	0.939s	0.30e-2	0.37e-2	0.60e-2	0.178s
G_{dr}	0.4	0.04	3.87e-2	4.73e-2	18.59e-2	1.325s	1.04e-2	1.22e-2	1.59e-2	0.155s
G_{dr}	0.8	0.01	0.30e-2	0.34e-2	0.41e-2	0.781s	0.16e-2	0.18e-2	0.28e-2	0.149s
G_{dr}	0.8	0.04	1.19e-2	1.35e-2	1.78e-2	1.006s	0.57e-2	0.70e-2	0.87e-2	0.133s
G_{di}	0.4	0.01	2.17e-2	17.59e-2	50.51e-2	0.865s	0.39e-2	0.52e-2	0.93e-2	0.179s
G_{di}	0.4	0.04	5.46e-2	19.40e-2	53.88e-2	1.043s	1.25e-2	1.55e-2	2.42e-2	0.169s
G_{di}	0.8	0.01	0.34e-2	0.42e-2	0.58e-2	0.766s	0.17e-2	0.24e-2	0.33e-2	0.159s
G_{di}	0.8	0.04	1.39e-2	1.66e-2	2.30e-2	0.972s	0.68e-2	0.86e-2	1.16e-2	0.141s
G_{sr}	0.8	0.01	0.58e-2	0.65e-2	0.79e-2	10.062s	0.38e-2	0.45e-2	0.61e-2	1.852s
G_{sr}	0.8	0.04	2.35e-2	2.62e-2	3.54e-2	12.375s	1.35e-2	1.55e-2	2.05e-2	1.577s
G_{sr}	1.0	0.01	0.45e-2	0.50e-2	0.58e-2	9.798s	0.28e-2	0.32e-2	0.39e-2	0.188s
G_{sr}	1.0	0.04	1.84e-2	1.99e-2	2.36e-2	11.626s	1.14e-2	1.29e-2	1.60e-2	0.179s
G_{si}	0.8	0.01	0.72e-2	0.85e-2	75.85e-2	10.236s	0.52e-2	0.64e-2	1.10e-2	1.835s
G_{si}	0.8	0.04	2.88e-2	3.38e-2	11.48e-2	12.350s	1.79e-2	2.16e-2	3.59e-2	1.610s
G_{si}	1.0	0.01	0.53e-2	0.62e-2	0.77e-2	9.388s	0.37e-2	0.43e-2	0.57e-2	0.180s
G_{si}	1.0	0.04	2.24e-2	2.52e-2	3.12e-2	12.200s	1.44e-2	1.72e-2	2.47e-2	0.187s

Table 1: Experimental results comparing TranSync and Coordinate Descent (CD) under different settings. All statistics (min, median, max) and mean running time are computed among 100 independent experiments with the same setting. As observed, TranSync outperforms Coordinate Descent in all experiments.

- G_{sr} (sparse, regular): The third graph is generated in a similar fashion as the first graph, except that the number of nodes $n = 20K$, and the connecting probability is set to $p = 0.003$. The expected degree of each vertex is 60.
- G_{si} (sparse, irregular): The fourth graph is generated in a similar fashion as the second graph, except that the number of nodes $n = 20K$, and the connecting probability between a pair of vertices is $p = 0.1s_i s_j$, where $s_i = 0.07 + 0.21\frac{i-1}{n-1}, 1 \le i \le n$ are scalar values associated the vertices. The expected degree of each vertex is about 60.

For this experiment, instead of using $k_{\max}$ as stopping condition as in Algorithm 1, we stop when we observe $\delta_k < \delta_{\min}$. Here $\delta_{\min}$ does not need to be close to σ. In fact, we choose $\delta_{\min} = 0.05, 0.1$ for $\sigma = 0.01, 0.04$, respectively. We also claim that if a small validation set (with size significantly less than n) of correct observations is available, our performance could be further improved.

As illustrated in Table 1, TranSync dominates coordinate descent in terms of both accuracy and prediction. In particular, TranSync is significantly better than coordinate descent on dense graphs in terms of accuracy. In particular, on dense but irregular graphs, coordinate descent did not converge at all when $p = 0.8$. The main advantage of TranSync on sparse graphs is the computational cost, although the accuracy is still considerably better than coordinate descent.

4.3 Experimental Evaluation on Real Datasets

Translation synchronization for joint alignment of point clouds. In the first application, we consider the problem of joint alignment of point clouds from pair-wise alignment [10]. To this end, we utilize the Patriot Circle Lidar dataset[1]. We uniformly subsampled the dataset to 6K scans. We applied Super4PCS [19] to match each scan to 300 randomly selected scans, where each match returns a pair-wise rigid transformation and a score. We then pick the top-30 matches for each scan, this results in a graph with 140K edges. To create the input data for translation synchronization, we run the state-of-the-art rotation synchronization algorithm described in [2] to estimate a global pose R_i for each scan. The pair-wise measurement $\mathbf{t}_{ij}$ from node i to node j is then given by $R_i^T \mathbf{t}_{ij}^{\text{local}}$, where $\mathbf{t}_{ij}^{\text{local}}$ is the translation vector obtained in pair-wise matching. The average outlier ratio of the pair-wise matches per node is 35%, which is relatively high since the observation graph is fairly sparse. Since $\mathbf{t}_{ij}$ is a 3D vector, we run TranSync three times, one for each coordinate. As illustrated in Figure 1, TranSync is able to recover the the global shape of the underlying scanning trajectory. In contrast, coordinate descent completely fails on this dataset.

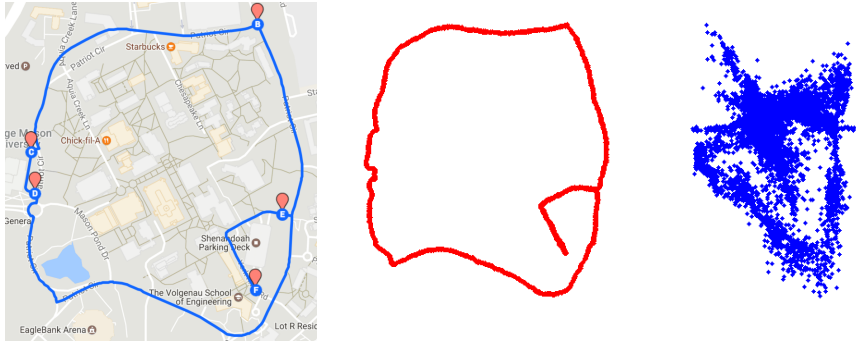[1] http://masc.cs.gmu.edu/wiki/MapGMU

Figure 1: The application of TranSync in joint alignment of 6K Lidar scans around a city block. (a) Snapshot of the underlying scanning trajectory. (b) Reconstruction using TranSync (c) Reconstruction using Coordinate Descent.

Movie	Global ranking (score)					
	MRQE	Hodge-Diff.	Hodge-Ratio	Hodge-Binary	TS-Init	TS-Final
Shakespeare in Love	1(85)	1(0.247)	2(0.078)	1 (0.138)	1(0.135)	1(0.219)
Witness	2(77)	2(0.217)	1(0.088)	3(0.107)	3(0.076)	2(0.095)
October Sky	3(76)	3(0.213)	3(0.078)	2(0.111)	2(0.092)	3(0.0714)
The Waterboy	4(66)	6(-0.464)	6(-0.162)	6(-0.252)	5(-0.134)	4(-0.112)
Interview with the Vampire	5(65)	4(-0.031)	4(-0.012)	4(-0.120)	4 (-0.098)	5(-0.140)
Dune	6(44)	5(-0.183)	5(-0.069)	5(-0.092)	6(-0.216)	6(-0.281)

Table 2: Global ranking of selected six movies via different methods: MRQE, HodgeRank[16] with 1) arithmetic mean score difference, 2) geometric mean score ratio and 3) and binary comparisons, and the initial and final predictions of TranSync. TranSync results in the most consistent result with MRQE.

Ranking from relative comparisons. In the second application, we apply TranSync to predict global rankings of Netflix movies from their relative comparisons provided by users. The Netflix dataset contains 17070 movies that were rated between October, 1998 and December, 2005. We adapt the procedure described in [16] to generate the input data. Specifically, for each pair of movies, we average the relative ratings from the same users within the same month. We only consider a relative measurement if we collect more than 10 such relative ratings. We then apply TranSync to predict the global rankings of all the movies. We report the initial prediction obtained by the first step of TranSync (i.e., all the relative comparisons are used) and the final prediction suggested by TranSync (i.e., after removing inconsistent relative comparisons).

Table 2 compares TranSync with HodgeRank [16] on six representative movies that are studied in [16]. The experimental results show that both predictions appear to be more consistent with MRQE[2] (the largest online directory of movie reviews on the internet) than HodgeRank [16] and its variants, which were only applied on these six movies in isolation. Moreover, the final prediction is superior to the initial prediction. These observations indicate two key advantages of TranSync, i.e., scalability on large-scale datasets and robustness to noisy relative comparisons.

5 Conclusions and Future Work

In this paper, we have introduced an iterative algorithm for solving the translation synchronization problem, which estimates the global locations of objects from noisy measurements of relative locations. We have justified the performance of our approach both experimentally and theoretically under both deterministic and randomized conditions. Our approach is more scalable and accurate than the standard linear programming formulation. In particular, when the pair-wise measurement

---

[2]http://www.mrqe.com

is biased, our approach can still achieve sub-constant recovery rate, while the linear programming approach can tolerate no more than 50% of the measurements being biased.

In the future, we plan to extend this iterative scheme to other synchronization problems, such as synchronizing rotations and point-based maps. Moreover, it would also be interesting to study variants of the iterative scheme such as re-weighted least squares. We would also like to close the gap between the current recovery rate and the lower bound, which exhibits a poly-log factor. This requires developing new tools for analyzing the iterative algorithm.

References

[1] F. Arrigoni, A. Fusiello, B. Rossi, and P. Fragneto. Robust rotation synchronization via low-rank and sparse matrix decomposition. *CoRR*, abs/1505.06079, 2015.

[2] A. Chatterjee and V. M. Govindu. Efficient and robust large-scale rotation averaging. In *2013 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2013.

[3] Y. Chen and E. J. Candès. The projected power method: An efficient algorithm for joint alignment from pairwise differences. *CoRR*, abs/1609.05820, 2016.

[4] Y. Chen, L. J. Guibas, and Q. Huang. Near-optimal joint object matching via convex relaxation. 2014.

[5] T. S. Cho, S. Avidan, and W. T. Freeman. A probabilistic image jigsaw puzzle solver. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[6] F. R. K. Chung and P. Horn. The spectral gap of a random subgraph of a graph. *Internet Mathematics*, 4(2):225–244, 2007.

[7] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 3001–3008, 2011.

[8] R. Cruz, K. Fernandes, J. S. Cardoso, and J. F. P. da Costa. Tackling class imbalance with ranking. In *2016 International Joint Conference on Neural Networks, IJCNN 2016, Vancouver, BC, Canada, July 24-29, 2016*, pages 2182–2187, 2016.

[9] I. Daubechies, R. Devore, M. Fornasier, and C. S. Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Comm. Pure Appl. Math.*

[10] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann. Robust global registration. In *Proceedings of the Third Eurographics Symposium on Geometry Processing*, SGP '05, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association.

[11] D. Goldberg, C. Malon, and M. Bern. A global approach to automatic solution of jigsaw puzzles. In *Proceedings of the Eighteenth Annual Symposium on Computational Geometry*, SCG '02, pages 82–87, 2002.

[12] R. M. Heiberger and R. A. Becker. Design of an s function for robust regression using iteratively reweighted least squares. pages 112–116, 1992.

[13] Q. Huang and L. Guibas. Consistent shape maps via semidefinite programming. *Computer Graphics Forum, Proc. Eurographics Symposium on Geometry Processing (SGP)*, 32(5):177–186, 2013.

[14] Q.-X. Huang, S. Flöry, N. Gelfand, M. Hofer, and H. Pottmann. Reassembling fractured objects by geometric matching. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, pages 569–578, 2006.

[15] D. Huber. *Automatic Three-dimensional Modeling from Reality*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, December 2002.

[16] X. Jiang, L. Lim, Y. Yao, and Y. Ye. Statistical ranking and combinatorial hodge theory. *Math. Program.*, 127(1):203–244, 2011.

[17] V. G. Kim, W. Li, N. Mitra, S. DiVerdi, and T. Funkhouser. Exploring collections of 3D models using fuzzy correspondences. *Transactions on Graphics (Proc. of SIGGRAPH 2012)*, 31(4), Aug. 2012.

[18] W. Marande and G. Burger. Mitochondrial dna as a genomic jigsaw puzzle. *Science*, 318:5849, July 2007.

[19] N. Mellado, D. Aiger, and N. J. Mitra. Super 4pcs fast global pointcloud registration via smart indexing. *Computer Graphics Forum*, 33(5):205–215, 2014.

[20] A. Nguyen, M. Ben-Chen, K. Welnicka, Y. Ye, and L. Guibas. An optimization approach to improving collections of shape maps. In *Eurographics Symposium on Geometry Processing (SGP)*, pages 1481–1491, 2011.

[21] D. Pachauri, R. Kondor, G. Sargur, and V. Singh. Permutation diffusion maps (pdm) with application to the image association problem in computer vision. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 541–549. Curran Associates, Inc., 2014.

[22] D. Pachauri, R. Kondor, and V. Singh. Solving the multi-way matching problem by permutation synchronization. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1860–1868. Curran Associates, Inc., 2013.

[23] R. Roberts, S. N. Sinha, R. Szeliski, and D. Steedly. Structure from motion for scenes with large duplicate structures. pages 3137–3144. Computer Vision and Patter Recognition, June 2011.

[24] Y. Shen, Q. Huang, N. Srebro, and S. Sanghavi. Normalized spectral map synchronization. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4925–4933. 2016.

[25] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. *ACM Trans. Graph.*, 25(3):835–846, July 2006.

[26] L. Wang and A. Singer. Exact and stable recovery of rotations for robust synchronization. *CoRR*, abs/1211.2441, 2012.

[27] C. Zach, M. Klopschitz, and M. Pollefeys. Disambiguating visual relations using loop constraints. In *CVPR*, pages 1426–1433, 2010.

[28] X. Zhou, M. Zhu, and K. Daniilidis. Multi-image matching via fast alternating minimization. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 4032–4040, 2015.

A Proof of Theorem 3.1

Proposition A.1 *Given two graph Laplacian matrices $L_1, L_2 \in \mathbb{R}^{n \times n}$. Suppose $\lambda_n(L_2) < \lambda_2(L_1)$, then*

$$(L_1 - L_2)^\dagger = L_1^\dagger + \sum_{n=1}^{\infty} (L_1^\dagger L_2)^n L_1^\dagger.$$

Proof: By using the identities $L^\dagger L = I - J$, $JL = 0$ in which L is a Laplacian matrix and $J = \frac{1}{n} e \cdot e^T$, we have

$$L_1(I - L_1^\dagger L_2) = L_1 - (I - J)L_2 = L_1 - L_2.$$

Plugging it into $(L_1 - L_2)^\dagger$, we obtain

$$(L_1 - L_2)^\dagger = (L_1(I - L_1^\dagger L_2))^\dagger) = (I - L_1^\dagger L_2)^\dagger L_1^\dagger.$$

With orthogonal decomposition, we can express L_1, L_2 as

$$L_1 = \sum_{i=2}^{n} \lambda_i(L_1) u_i u_i^T, \, L_2 = \sum_{i=2}^{n} \lambda_i(L_2) v_i v_i^T,$$

in which $\{u_i\}, \{v_i\}$ are the corresponding eigenvectors of L_1, L_2 respectively. In this way,

$$L_1^\dagger = \sum_{i=2}^{n} \frac{1}{\lambda_i} u_i u_i^T.$$

By definition of quadratic matrix norm, for any vector $x \in \mathbb{R}^n, \|x\|_2 = 1$, we have

$$\|\sum_{i=2} \lambda_i(L_2) v_i v_i^T x\|_2 \le \lambda_n(L_2),$$

and

$$\|\sum_{i=2} \lambda_i(L_2) v_i v_i^T \sum_{i=2}^{n} \lambda_i(L_1) u_i u_i^T x\|_2 \le \frac{\lambda_n(L_2)}{\lambda_2(L_1)} < 1,$$

which means $\|L_1^\dagger L_2\|_2 < 1$.

Since $\|L_1^\dagger L_2\|_2 < 1$, the matrix series $M = I + \sum_{n=1}^{\infty} (L_1^\dagger L_2)^n$ converges and $(I - L_1^\dagger L_2)M = I$. Thus $I - L_1^\dagger L_2$ is invertible and its inversion is $I + \sum_{n=1}^{\infty} (L_1^\dagger L_2)^n$. Hence we obtain

$$(L_1 - L_2)^\dagger = L_1^\dagger + \sum_{n=1}^{\infty} (L_1^\dagger L_2)^n L_1^\dagger.$$

$\square$

Proposition A.2 *Given two n by n Laplacian matrices L_G, $L_{G'}$ corresponding to graph G, G', which satisfies $G' \subseteq G$. $L_{G'}$ equals to $B_{G'} B_{G'}^T$, where $B_{G'}$ is the edge matrix of G'. The maximum degree among all vertices of G' is less or equal to d, then we have the following inequality:*

$$\|L_G^\dagger B_{G'}\|_\infty \le d \left(\max_k |L_{G,kk}^\dagger| + \max_{i \ne j} |L_{G,ij}^\dagger| + \frac{n}{2} \max_{\substack{i,j,k \\ \text{pairwisely different}}} |L_{G,ki}^\dagger - L_{G,kj}^\dagger| \right)$$

In particular, if G is a clique, the above inequality can be reduced to

$$\|L_G^\dagger B_{G'}\|_\infty \le \frac{d}{n}.$$

$$\|L_G^\dagger B_{G'}\|_\infty = \max_k \{\|e_k^T L_G^\dagger B_{G'}\|_1\}$$

$$= \max_k \{\sum_{(i,j)\in G'} |L_{G,ki}^\dagger - L_{G,kj}^\dagger|\}$$

$$\leq \max_k \sum_{\substack{j\in\mathcal{N}(G',k)}} |L_{G,kk}^\dagger - L_{G,kj}^\dagger| + \max_k \sum_{\substack{i,j\neq k \\ (i,j)\in G'}} |L_{G,ki}^\dagger - L_{G,kj}^\dagger|$$

$$\leq d\max_k |L_{G,kk}^\dagger| + \max_k \sum_{j\in\mathcal{N}(G',k)} |L_{G,kj}^\dagger| + \max_k \sum_{\substack{i,j\neq k \\ (i,j)\in G'}} |L_{G,ki}^\dagger - L_{G,kj}^\dagger|$$

$$\leq d\max_k |L_{G,kk}^\dagger| + d\max_{i\neq j} L_{G,ij}^\dagger + \frac{nd}{2} \max_{\substack{i,j,k \\ \text{pairwisely different}}} |L_{G,ki}^\dagger - L_{G,kj}^\dagger|$$

The last line comes from the fact that the size of neighborhoods $\mathcal{N}(G',k)$ is upper bounded by d so the number of edges in G' should be bounded by $\frac{nd}{2}$ as well. □

Proposition A.3 *Given two n by n Laplacian matrices L_G, $L_{G'}$ corresponding to graph G, G', which satisfies $G' \subseteq G$. The maximal degree among all vertices of G' is equal or less than d, then we claim*

$$\|L_G^\dagger L_{G'}\|_\infty \leq d\left(2\max_k |L_{G,kk}^\dagger| + 2\max_{i\neq j} |L_{G,ij}^\dagger| + n \max_{\substack{i,j,k \\ \text{pairwisely different}}} |L_{G,ki}^\dagger - L_{G,kj}^\dagger|\right)$$

In particular, if G is a clique, the above inequality can be reduced to

$$\|L_G^\dagger L_{G'}\|_\infty \leq \frac{2d}{n}.$$

Proof:

$$\|L_G^\dagger L_{G'}\|_\infty = \max_k \{\sum_j |\sum_i L_{G,ki}^\dagger L_{G',ij}|\}$$

$$= \max_k \{\sum_j |L_{G,kj}^\dagger deg(G',j) - \sum_{i\in\mathcal{N}(G',j)} L_{G,ki}^\dagger|\}$$

$$= \max_k \{\sum_j |\sum_{i\in\mathcal{N}(G',j)} (L_{G,kj}^\dagger - L_{G,ki}^\dagger)|\}$$

$$\leq 2\max_k \sum_{(i,j)\in G'} |L_{G,kj}^\dagger - L_{G,ki}^\dagger|$$

$$= 2\sum_{j\in\mathcal{N}(G',k)} |L_{G,kk}^\dagger - L_{G,kj}^\dagger| + 2\sum_{\substack{(i,j)\in G' \\ i,j\neq k}} |L_{G,ki}^\dagger - L_{G,kj}^\dagger|$$

$$\leq 2d\max_k |L_{G,kk}^\dagger| + 2d\max_{i\neq j} |L_{G,ij}^\dagger| + n \max_{\substack{i,j,k \\ \text{pairwisely different}}} |L_{G,ki}^\dagger - L_{G,kj}^\dagger|$$

□

Proof of Theorem 3.1: At first we show that assuming $\mathbf{x}^{gt} = 0$ will not damage the generality of the proof. To this end, do transformation from original measurements $\mathbf{t}$ to $\mathbf{t}'$ as $t'_{ij} = t_{ij} - x_i^{gt} + x_j^{gt}$. The condition for correct measurement t_{ij} turns to $|t'_{ij}| \leq \sigma$. If $\mathbf{t}'$ with ground truth $\mathbf{0}$, the same truncation strategy as $\mathbf{t}$ with $\mathbf{x}^{gt}$ produces iterative solution $\mathbf{x}'^{(k)}$, and initializaion $\mathbf{x}'^{(0)} = \mathbf{x}^{(0)} - \mathbf{x}^{gt}$, then we assert that

$$\mathbf{x}^{(k)} = \mathbf{x}'^{(k)} + \mathbf{x}^{gt}$$

holds for all $k \in \mathbb{N}$.

By induction we assume $\mathbf{x}^{(k)} = \mathbf{x'}^{(k)} + \mathbf{x}^{gt}$, thus the truncated graph G_k of $\mathbf{x'}^{(k)}$ must be the same as $\mathbf{x}^{(k)}$ since they use the same truncation strategy. Our algorithm provides the next $\mathbf{x}$ and $\mathbf{x'}$ as

$$\mathbf{x}^{(k+1)} = L_{G_k}^\dagger B_{G_k} \mathbf{t}_{G_k}$$

$$\mathbf{x'}^{(k+1)} = L_{G_k}^\dagger B_{G_k} \mathbf{t'}_{G_k}.$$

Since our truncation strategy make sure the correct measurements will not be truncated, G_k is certainly a connected graph. Thus $\mathbf{x}^{gt}$ is the unique precise solution of the linear function $B_{G_k}\mathbf{x} = \mathbf{t}_{G_k}^0$ in which $\mathbf{t}_{G_k}^0$ is the measurements without error on graph G_k. Thus by the process of the derivation of $L_{G_k}^\dagger B_{G_k}$ we know that

$$L_{G_k}^\dagger B_{G_k} t_{G_k}^0 = \mathbf{x}^{gt}.$$

Note $\mathbf{t}_{G_k} = \mathbf{t'}_{G_k} + \mathbf{t}_{G_k}^0$, hence we get the identity

$$\mathbf{x}^{(k+1)} = \mathbf{x'}^{(k+1)} + \mathbf{x}^{gt}.$$

We have seen $\mathbf{x}^{(k)}$ with $\mathbf{x}^{gt}$ behaves completely the same as $\mathbf{x'}^{(k)}$ with $\mathbf{0}$, therefore they must have the same concentration bound, so we assume $\mathbf{x}^{gt} = \mathbf{0}$ below.

Returning to the original proposition. Prove this theorem by induction. Assume

$$\|\mathbf{x}^{(k)}\|_\infty \leq q\sigma + 2p\epsilon c^{k-1}$$

and

$$k \leq -\log\left(\frac{\epsilon(c-4p)}{(1+2q)\,\sigma}\right) / \log c + 1$$

, or

$$(1+2q)\sigma \leq \epsilon(c-4p)c^{k-1}.$$

At k-th iteration, the truncation threshold should be $c^k\epsilon$. Since for any correct measurement t_{ij} we have

$$|t_{ij} - x_i^{(k)} + x_j^{(k)}| \leq \sigma + 2\|\mathbf{x}^{(k)}\|_\infty \leq (1+2q)\sigma + 4p\epsilon c^{k-1} \leq \epsilon c^k,$$

no correct measurement can be truncated. On the other hand, all survived measurement should satisfy

$$|t_{ij}| \leq |t_{ij} - x_i^{(k)} + x_j^{(k)}| + 2|\mathbf{x}^{(k)}|_\infty \leq \epsilon c^k + 2q\sigma + 4p\epsilon c^{k-1}$$

Let G_T be the graph consisting of all edges that are dropped at k-th iteration. According to the previous argument, we can write x^k as

$$\mathbf{x}^{(k+1)} = (L_G - L_{\overline{G}})^\dagger B_{G\backslash\overline{G}} \mathbf{t}^{(k)}$$

in which $L_G, L_{\overline{G}}$ are the Laplacian matrices of G and G_T respectively while $B_{G\backslash\overline{G}}$ is the edge adjacent matrix of $G\backslash\overline{G}$. It is clear that $G_T \subseteq G\backslash G_a$.

Using Proposition A.1, we have

$$(L_G - L_{\overline{G}})^\dagger = \sum_{k=0}^{\infty} (L_G^\dagger L_{\overline{G}})^n L_G^\dagger.$$

Using Proposition A.2, A.3, we obtain upper bounds

$$\|L_G^\dagger B_{G\backslash G_{good}}\|_\infty \leq d_{bad}\alpha = h,$$

$$\|L_G^\dagger L_{\overline{G}}\|_\infty \leq 2d_{bad}\alpha = 2h,$$

and

$$\|L_G^\dagger B_{G_{good}}\|_\infty \leq n\alpha,$$

so that

$$\|\mathbf{x}^{(k+1)}\|_\infty = \|\sum_{k=0}^\infty (L_G^\dagger L_{\overline{G}})^n (L_G^\dagger B_{G\setminus\overline{G}}\mathbf{t}^{(k)})\|_\infty$$

$$\leq \frac{1}{1-2h}\left\|L_G^\dagger B_{G_{good}}\mathbf{t}_{G_{good}} + L_G^\dagger B_{G\setminus\overline{G}\setminus G_{good}}\mathbf{t}_{G\setminus\overline{G}\setminus G_{good}}\right\|_\infty$$

$$\leq \frac{1}{1-2h}\left(n\alpha\|\mathbf{t}_{G_{good}}\|_\infty + d_{bad}\alpha\|\mathbf{t}_{G\setminus\overline{G}\setminus G_{good}}\|_\infty\right)$$

$$\leq \frac{1}{1-2h}\left(n\alpha\sigma + h(\epsilon c^k + 2q\sigma + 4p\epsilon c^{k-1})\right)$$

$$= q\sigma + p\sigma + p(\epsilon c^k + 2q\sigma + 4p\epsilon c^{k-1})$$

$$\leq q\sigma + 2p\epsilon c^k$$

in which the last line used the inductive condition $(1+2q)\sigma \leq \epsilon(c-4p)c^{k-1}$. The correctness of proposition follows immediately by induction.

Continue the iterative process until $\epsilon(c-4p)c^k < (1+2q)\sigma$ (this means $(1+2q)\sigma \leq \epsilon(c-4p)c^{k-1}$, so our inductive argument works for $x^{(k)}$), at which time we obtain the bound on $x^{(k)}$

$$\|\mathbf{x}^{(k+1)}\|_\infty \leq q\sigma + 2p\epsilon c^k \leq \frac{2p+cq}{c-4p}\sigma$$

$\square$

B Analysis of the randomized case

B.1 Proof of Lemma 3.1

The key idea is to leverage the independence of $\{t_{ij}\}$ from $\mathbf{x}^{(0)}$ and redefine the noise model so that the selection of the edges is separated from the measurements. To simplify the notations, we denote

$$\overline{r} = \frac{2\delta}{a+b}, \quad r = \frac{p}{p+(1-p)\overline{r}}.$$

Lemma B.1 *Associate each edge $(i,j) \in \mathcal{E}$ with a random variable w_{ij} given by*

$$w_{ij} = \begin{cases} 1 & \text{with probability} \quad p + \overline{r}(1-p), \\ 0 & \text{with probability} \quad (1-\overline{r})(1-p) \end{cases} \tag{7}$$

Redefine the independent measurement $\hat{t}_{ij}$ along each edge as

$$\hat{t}_{ij} = \overline{t}_{ij} + (1-r)(x_i^{(0)} - x_j^{(0)}), \quad \overline{t}_{ij} := \begin{cases} -(1-r)(x_i^0 - x_j^0) + \sigma U[-1,1] & \text{with probability} \quad r \\ r(x_i^0 - x_j^0) + \overline{r}\zeta_{ij} & \text{with probability} \quad 1-r \end{cases}$$

Then this noise model and the original model are identical.

Proof: It is clear that the probability that an edge is selected is $p + \frac{2\delta}{a+b}(1-p) = p + (1-p)\overline{r}$. Moreover,

$$w_{ij}\overline{t}_{ij} = \begin{cases} 0 & \text{with probability} \quad p \\ (x_i^{(0)} - x_j^{(0)}) + \overline{r}\zeta_{ij} & \text{with probability} \quad \frac{2\delta}{a+b}(1-p) \\ 0 & \text{with probability} \quad \frac{a+b-2\delta}{a+b}(1-p) \end{cases}$$

$\square$

The following proposition provides a decomposition of $\mathbf{x}^{(1)}$ under the new noise model.

Lemma B.2 *Let L_w and B_w denote the truncated Laplacian matrix and vertex-edge adjacency matrix, respectively. Let $\overline{t}$ collect $\overline{t}_{ij}, (i,j) \in \mathcal{G}$. Then*

$$\boldsymbol{x}^{(1)} = (1-r)\boldsymbol{x}^{(0)} + L_w^+ B_w \overline{\boldsymbol{t}}. \tag{8}$$

Proof: Let $\hat{\mathbf{t}}$ collect $\hat{t}_{ij}, (i,j) \in \mathcal{G}$, then

$$
\begin{aligned}
\mathbf{x}^{(1)} = L_\mathbf{w}^+ B_\mathbf{w} \hat{\mathbf{t}} &= L_\mathbf{w}^+ B_\mathbf{w}((1-r)B^T \mathbf{x}^{(0)} + \bar{\mathbf{t}}) \\
&= (1-r)(I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T)\mathbf{x}^{(0)} + L_\mathbf{w}^+ B_\mathbf{w}\bar{\mathbf{t}} \\
&= (1-r)x^{(0)} + L_\mathbf{w}^+ B_\mathbf{w}\bar{\mathbf{t}}.
\end{aligned}
$$

$\square$

Since $L_\mathbf{w}^+ B_\mathbf{w}$ and $\bar{\mathbf{t}}$ are independent, (8) allows us to apply concentration bounds on $\bar{t}_{ij}$. To this end, we first establish the following inequality:

Lemma B.3 *Denote* $\|\mathbf{x}^{(0)}\|_{d,\infty} = \max\limits_{1 \le i,j \le n} |x_i^{(0)} - x_j^{(0)}|$ *and let* $\bar{r}_2 = \max(\|\mathbf{x}^{(0)}\|_{d,\infty}, \bar{r})$. *For fixed* **w**, *we have*

$$
\text{Var}\big(\mathbf{e}_i^T L_\mathbf{w}^+ B_\mathbf{w}\bar{\mathbf{t}}\big) \le \big((1-r)(\frac{1}{3}+r) \cdot \bar{r}_2^2 + \frac{r\sigma^2}{3}\big)L_{\mathbf{w},ii}^+ \tag{9}
$$

Proof: First of all, we have

$$
\begin{aligned}
\text{Var}\big(\bar{t}_{ij}\big) &= (1-r)(\frac{\bar{r}^2}{3} + r(x_i^{(0)} - x_j^{(0)})^2) + \frac{r\sigma^2}{3} \\
&\le (1-r)(\frac{\bar{r}^2}{3} + r\|\mathbf{x}^{(0)}\|_{d,\infty}^2) + \frac{r\sigma^2}{3}
\end{aligned}
$$

It follows that

$$
\begin{aligned}
\text{Var}\big(\mathbf{e}_i^T L_\mathbf{w}^+ B_\mathbf{w}\bar{\mathbf{t}}\big) &= \sum_{(j,k)\in\mathcal{G}} \text{Var}\big(w_{ij}(L_{\mathbf{w},ij}^+ - L_{\mathbf{w},ik}^+)\bar{t}_{ij}\big) \\
&\le \sum_{(j,k)\in\mathcal{G}} w_{ij}(L_{\mathbf{w},ij}^+ - L_{\mathbf{w},ik}^+)^2(1-r)(\frac{\bar{r}^2}{3} + r\|\mathbf{x}^{(0)}\|_{d,\infty}^2) \\
&= \mathbf{e}_i^T L_\mathbf{w}^+ \cdot L_\mathbf{w} \cdot L_\mathbf{w}^+ \mathbf{e}_i^T (1-r)(\frac{\bar{r}^2}{3} + r\|\mathbf{x}^{(0)}\|_{d,\infty}^2) \\
&= \mathbf{e}_i^T L_\mathbf{w}^+ \mathbf{e}_i (1-r)(\frac{\bar{r}^2}{3} + r\|\mathbf{x}^{(0)}\|_{d,\infty}^2) \\
&= (1-r)(\frac{\bar{r}^2}{3} + r\|\mathbf{x}^{(0)}\|_{d,\infty}^2) \cdot L_{\mathbf{w},ii}^+ \\
&\le \big((1-r)(\frac{1}{3}+r) \cdot \bar{r}_2^2 + \frac{r\sigma^2}{3}\big)L_{\mathbf{w},ii}^+.
\end{aligned}
$$

$\square$

Moreover, the range of each summand in $\mathbf{e}_i^T L_\mathbf{w}^+ B_\mathbf{w}\bar{\mathbf{t}}$ is bounded above by

$$
\max_{1 \le j,k} |L_{\mathbf{w},ij}^+ - L_{\mathbf{w},ik}^+|| \max(\bar{t}_{ij}) - \min(\bar{t}_{ij})| \le 2L_{\mathbf{w},ii}^+| \max(\bar{t}_{ij}) - \min(\bar{t}_{ij})|
$$

$$
\le 4L_{\mathbf{w},ii}^+\bar{r}_2.
$$

The following proposition directly follows from the Bernstein inequality:

Fact B.1 *Let* $\|\text{Diag}(L_\mathbf{w}^+)\|_\infty = \max\limits_{1 \le i \le n} L_{\mathbf{w},ii}^+$. *For fixed* **w**, *we have for*

$$
\Pr\big(\|\mathbf{x}^{(1)} - (1-r)\mathbf{x}^{(0)}\| \ge \bar{r}_2\sqrt{\text{Diag}(L_\mathbf{w}^+)t}\big) \le 2n \exp\Big(-\frac{t^2}{2\big((1-r)(\frac{1}{3}+r) + \frac{4}{3}\sqrt{\text{Diag}(L_\mathbf{w}^+) \cdot t}\big)}\Big). \tag{10}
$$

It remains to bound the diagonal entries of $L_\mathbf{w}^+$, which is given below:

15

Lemma B.4 *Let $d_{\min} = \Omega(log^2(n))$ be the minimal degree of $\mathcal{G}$. Suppose $p + \bar{r}(1-p) = \Omega(\log^2(n)/d_{\min})$. Then w.h.p.,*

$$\|\mathrm{Diag}(L_{\boldsymbol{w}}^+)\|_\infty \leq \frac{1 + o(1)}{(p + \bar{r}(1-p))d_{\min}\lambda_2(\overline{L}_\mathcal{G})}, \tag{11}$$

where $\overline{L}_\mathcal{G}$ is the normalized graph Laplacian of $\mathcal{G}$.

Proof: We first show that for the Laplacian matrix L of any graph G,

$$L_{ii}^+ \leq \frac{1}{d_i \lambda_2(\overline{L})},$$

where $\overline{L}$ is the normalized graph Laplacian of G. In fact,

$$L_{ii}^+ = \frac{1}{d_i}\overline{L}_{ii}^+ \leq \frac{1}{d_i}\lambda_n(\overline{L}^+) = \frac{1}{d_i}\lambda_2(\overline{L}).$$

The rest of the proof follows from the concentration of vertex degrees of random subgraphs and Theorem 1 in [6]. □

Now we can complete the proof of Lemma B.1 by setting $t = O(\sqrt{\log(n)})$ in (10).

B.2 Proof of Theorem 3.2

As shown in the previous Section, we assume $\mathbf{x}^{gt} = 0$ with losing generality. Lemma 3.1 tells us that for fixed $\mathbf{x}^{(0)}$, one step of TranSync results in a solution that is closer to the ground-truth solution. We can apply it to a dense samples along the segment between $\mathbf{x}^{gt}$ and $\mathbf{x}^{gt} + \frac{a+b}{2}\mathbf{1}$, e.g., $n\sqrt{n}(a+b)/2$ samples so that the distance between adjacent samples along each axis is at most $\frac{1}{n}$. It is clear that Lemma 3.1 still holds among these sample points.

To prove the convergence of $\mathbf{x}^{(k)}$, we seek to bound

$$\left\|\mathbf{x}^{(k+1)} - \frac{p}{p + (1-p)c^k}\mathbf{x}^{(k)}\right\|_\infty \leq \left\|\overline{\mathbf{x}}^{(k+1)} - \frac{p}{p + (1-p)c^k}\overline{\mathbf{x}}^{(k)}\right\|_\infty \tag{12}$$

$$+ \left\|\mathbf{x}^{(k+1)} - \overline{\mathbf{x}}^{(k+1)}\right\|_\infty + \frac{p}{p + (1-p)c^k}\left\|\mathbf{x}^{(k)} - \overline{\mathbf{x}}^{(k)}\right\|_\infty, \tag{13}$$

where $\overline{\mathbf{x}}^{(k)}$ is chosen to be the closest point of $\mathbf{x}^{(k)}$ to the segment under the L^∞ norm, and $\overline{\mathbf{x}}^{(k+1)}$ is the result of one step of TranSync. We can apply Lemma 3.1 to obtain a bound on $\|\overline{\mathbf{x}}^{(k+1)} - \frac{p}{p+(1-p)c^k}\overline{\mathbf{x}}^{(k)}\|_\infty$. It remains to bound $\|\mathbf{x}^{(k+1)} - \overline{\mathbf{x}}^{(k+1)}\|_\infty$. To this end, we start with the following Lemma.

Lemma B.5 *Given a fixed input $\boldsymbol{t}$, starting from two different points $\boldsymbol{x}^{(k)}$ and $\overline{\boldsymbol{x}}^{(k)}$. Let $\boldsymbol{x}^{(k+1)}$ and $\overline{\boldsymbol{x}}^{(k+1)}$ be the results of applying one step of TranSync. Then*

$$\|\boldsymbol{x}^{(k+1)} - \overline{\boldsymbol{x}}^{(k+1)}\|_\infty \leq \frac{2d_{\max}^{dif}\|L_{\boldsymbol{t},\overline{\boldsymbol{x}}^{(k)}}^+\|_{1,\infty}\left(\|\overline{\boldsymbol{x}}^{(k+1)}\|_\infty + \|\boldsymbol{x}^{(k)} - \overline{\boldsymbol{x}}^{(k)}\|_\infty + \frac{1}{2}c^k\right)}{1 - 2d_{\max}^{dif}\|L_{\boldsymbol{t},\overline{\boldsymbol{x}}^{(k)}}^+\|_{1,\infty}}, \tag{14}$$

where $L_{\mathrm{dif}} = L_{\boldsymbol{t},\overline{\boldsymbol{x}}^{(k)}} - L_{\boldsymbol{t},\boldsymbol{x}^{(k)}}$, and $L_{\boldsymbol{t},\boldsymbol{x}^{(k)}}$ and $L_{\boldsymbol{t},\overline{\boldsymbol{x}}^{(k)}}$) are truncated Laplacians derived from $\boldsymbol{x}^{(k)}$ and $\overline{\boldsymbol{x}}^{(k)}$, respectively. $d_{\max}^{dif}$ is the maximum number of different edges between these two graphs per vertex.

16

Proof: Let $B_{\mathbf{t},\mathbf{x}^{(k)}}$ and $B_{\mathbf{t},\bar{\mathbf{x}}^{(k)}})$ be the corresponding vertex-edge adjacency matrix. Define $B_{\mathrm{dif}} = B_{\mathbf{t},\bar{\mathbf{x}}^{(k)}} - B_{\mathbf{t},\mathbf{x}^{(k)}}$. First, we have

$$
\begin{aligned}
\|\bar{\mathbf{x}}^{(k+1)} - \mathbf{x}^{(k+1)}\|_\infty &= \|\left(L_{\mathbf{t},\bar{\mathbf{x}}^{(k)}} + L_{dif}\right)^+ (B_{\mathbf{t},\bar{\mathbf{x}}^{(k)}} + B_{dif})\mathbf{t} - L_{\mathbf{t},\bar{\mathbf{x}}^{(k)}}^+ B_{\mathbf{t},\bar{\mathbf{x}}^{(k)}}\mathbf{t}\|_\infty \\
&\leq \|\left(\left(L_{\mathbf{t},\bar{\mathbf{x}}^{(k)}} + L_{dif}\right)^+ - L_{\mathbf{t},\bar{\mathbf{x}}^{(k)}}^+\right) B_{\mathbf{t},\bar{\mathbf{x}}^{(k)}}\mathbf{t}\|_\infty + \|\left(L_{\mathbf{t},\bar{\mathbf{x}}^{(k)}} + L_{dif}\right)^+ B_{dif}\mathbf{t}\|_\infty \\
&= \|\sum_{i=0}^{+\infty} \left(L_{\mathbf{t},\bar{\mathbf{x}}^{(k)}}^+ L_{dif}\right)^i L_{\mathbf{t},\bar{\mathbf{x}}^{(k)}}^+ B_{\mathbf{t},\bar{\mathbf{x}}^{(k)}}\mathbf{t}\|_\infty + \|\sum_{i=0}^{+\infty} \left(L_{\mathbf{t},\bar{\mathbf{x}}^{(k)}}^+ L_{dif}\right)^i L_{\mathbf{t},\bar{\mathbf{x}}^{(k)}}^+ B_{dif}\mathbf{t}\|_\infty \\
&\leq \sum_{i=0}^{+\infty} \|L_{\mathbf{t},\bar{\mathbf{x}}^{(k)}}^+ L_{dif}\|_{1,\infty}^i \left(\|L_{\mathbf{t},\bar{\mathbf{x}}^{(k)}}^+ L_{dif}\|_{1,\infty}^i \|\bar{\mathbf{x}}^{(k+1)}\|_\infty + \|L_{\mathbf{t},\bar{\mathbf{x}}^{(k)}}^+ B_{dif}\mathbf{t}\|_\infty\right) \\
&= \frac{\|L_{\mathbf{t},\bar{\mathbf{x}}^{(k)}}^+ L_{\mathrm{dif}}\|_{1,\infty} \|\bar{\mathbf{x}}^{(k+1)}\|_\infty + \|L_{\mathbf{t},\bar{\mathbf{x}}^{(k)}}^+ B_{\mathrm{dif}}\mathbf{t}\|_\infty}{1 - \|L_{\mathbf{t},\bar{\mathbf{x}}^{(k)}}^+ L_{\mathrm{dif}}\|_{1,\infty}} \\
&\leq \frac{\|L_{\mathbf{t},\bar{\mathbf{x}}^{(k)}}^+\|_{1,\infty} d_{\max}^{dif} \left(2\|\bar{\mathbf{x}}^{(k+1)}\|_\infty + \|\mathbf{t}\|_\infty\right)}{1 - 2 d_{\max}^{dif} \|L_{\mathbf{t},\bar{\mathbf{x}}^{(k)}}^+\|_{1,\infty}}
\end{aligned}
$$

Now we can complete the proof as

$$
\|\mathbf{t}\|_\infty \leq \|\mathbf{x}^{(k)}\|_{d,\infty} + c^k \leq 2\|\mathbf{x}^{(k)} - \bar{\mathbf{x}}^{(k)}\|_\infty + c^k.
$$

$\square$

We proceed to control the two remaining quantities $d_{\max}^{dif}$ and $\|L_{\mathbf{t},\bar{\mathbf{x}}^{(k)}}^+\|_{1,\infty}$. In both cases, we leverage the fact that $\bar{\mathbf{x}}^{(k)}$ lies on the line between $\mathbf{0}$ and $\mathbf{1}$ so that we can utilize the independence of t_{ij}. We first provide an upper bound on $d_{\max}^{dif}$.

Lemma B.6 *Suppose $c^k = \Omega(\log^2(n)/n)$. Denote $d_{\max}^{\mathbf{t},\bar{\mathbf{x}}^{(k)}}$ as the maximum degree of graph truncated from $\bar{\mathbf{x}}^{(k)}$, then*

$$
\begin{aligned}
d_{\max}^{dif} &\leq 4\|\mathbf{x}^{(k)} - \bar{\mathbf{x}}^{(k)}\|_\infty d_{\max}^{\mathbf{t},\bar{\mathbf{x}}^k} + \log(n)\sqrt{4\|\mathbf{x}^{(k)} - \bar{\mathbf{x}}^{(k)}\|_\infty d_{\max}^{\mathbf{t},\bar{\mathbf{x}}^k}} \\
&\leq 6\|\mathbf{x}^{(k)} - \bar{\mathbf{x}}^{(k)}\|_\infty \left(p + (1-p)c^k\right)n
\end{aligned}
\tag{15}
$$

almost surely.

Proof: Note that the different edges are incurred if each t_{ij} falls in the two intervals $[(x_i^{(k)} - x_j^{(k)}) - c^k, (\bar{x}_i^{(k)} - \bar{x}_j^{(k)}) - c^k]$ and $[(x_i^{(k)} - x_j^{(k)}) + c^k, (\bar{x}_i^{(k)} - \bar{x}_j^{(k)} + c^k]$. The total length of these two intervals is at most $4\|\mathbf{x}^{(k)} - \bar{\mathbf{x}}^{(k)}\|_\infty$. The first inequality directly follows from the Bernstein inequality. The second inequality follows from the fact that maximum vertex degree of a random sub-graph of edge selection probability $p + (1-p)c^k$ is concentrated around $(p + (1-p)c^k)n$. $\square$

The following lemma provides a concentration of $\|L_{\mathbf{t},\bar{\mathbf{x}}^{(k)}}\|_{1,\infty}$.

Lemma B.7 *Consider random sub-graphs of clique K_n with edge selection probability q, we have*

$$
\|L_{\mathbf{w}}^+\|_{1,\infty} = \frac{1}{qn}\left(2 + \frac{O(1)}{\sqrt{q}}\right).
\tag{16}
$$

with high probability.

Proof: First of all, for any matrix $A \in \mathbb{R}^{n \times n}$,

$$
\|A\|_{1,\infty} = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}| \leq \max_{1 \leq i \leq n} \sqrt{n}\sqrt{\sum_{j=1}^n a_{ij}^2} \leq \sqrt{n}\sigma_{\max}(A).
$$

Since the non-zero eigenvalues of L^+ fall in-between $[\frac{1}{\lambda_n(L)}, \frac{1}{\lambda_2(L)}]$, it follows that

$$\|L\|_{1,\infty} \leq \|L^+ - \frac{1}{pn}(I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T)\|_{1,\infty} + \frac{2}{pn}$$
$$\leq \frac{2}{pn} + \sqrt{n}\max\left(|\frac{1}{\lambda_2(L)} - \frac{1}{pn}|, |\frac{1}{\lambda_n(L)} - \frac{1}{pn}|\right).$$

Since it is well known that the eigenvalues of the graph Laplacian of a Erdős-Rényi graph $G(n, q)$ is concentrated within in the interval $[qn - O(\sqrt{qn}), qn + O(\sqrt{qn})]$, it follows that

$$\|L\|_{1,\infty} \leq \frac{2}{pn} + \frac{O(\sqrt{qn})\sqrt{n}}{p^2n^2}.$$

$\square$

Completing the proof of Theorem 3.2. Now we are ready to prove Theorem 3.2. Denote $\delta_k = \|\mathbf{x}^{(k)} - \bar{\mathbf{x}}^{(k)}\|_\infty$. Combing (14), (15) and (16), we arrive at the following recursion:

$$\delta_{k+1} \leq \frac{15\delta_k(c^k + \delta_k)}{1 - 15\delta_k}. \tag{17}$$

As $\delta_1 = O(\frac{\log(n)}{\sqrt{n}})$. It follows that we can choose a small constant C_2 so that

$$\delta_k < p/128, \quad 1 \leq k \leq \min(C_2\sqrt{\log(n)}, \log(\frac{a+b}{2\sigma})/\log(1/(1-p/2))).$$

and

$$\frac{1}{32} \leq c^{C_2\sqrt{\log(n)}} \leq \frac{1}{16}.$$

It then follows from (17) that

$$\delta_k \leq pc^k/4$$

for sufficiently large n.

It remains to check $\|\mathbf{x}^{(k+1)}\|_\infty \leq \frac{1}{2}c^{k+1}$. In fact, using (13) we have

$$\|\mathbf{x}^{(k+1)}\|_\infty \leq \frac{(1-p)c^k}{2(p+(1-p)c^k)}c^k + c^kO(\log(n)/\sqrt{n}) + \frac{pc^k}{5} + \frac{p}{p+(1-p)c^k}\frac{pc^{k-1}}{5}$$
$$\leq \frac{1}{2}(1-p/2)c^k$$
$$\leq \frac{1}{2}c^{k+1},$$

which ends the proof of Theorem 3.2.