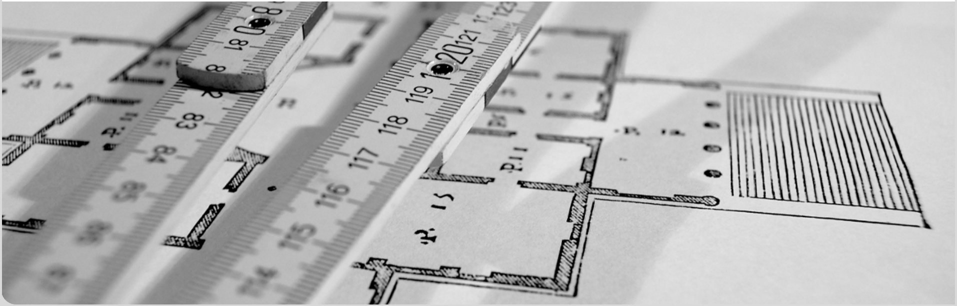
Karlsruhe Institute of Technology

Parallel Algorithm for Closest Pair Problem

Ge Wu | July 22, 2013

INSTITUTE FOR THEORETICAL INFORMATICS



Problem Description

Closest Pair Problem

- Given n **different** unordered points $P = \{p_1, p_2, ..., p_n\}$ in **unit square**:

$$p_i = (x_i, y_i) \in (0, 1) \times (0, 1) \subset \mathbb{R}^2$$

- Find a pair of points with closest euclidean distance between them
- Find any pair if there's a tie.

Background

- O(n log n) lower bound in comparison tree model
- 1976 Rabin proves that + floor function + randomness = ¿ O(n) expected time
- 1979 Steven Fortune and John E. Hopcroft + floor function but no randomness =¿ O(n loglog n)
- 1995 Samir Khuller and Yossi Matias Another O(n) algorithm that uses the floor function and randomization

Sample

- Compute all pair of distances: $O((n^{\frac{2}{3}})^2) = O(n^{\frac{4}{3}})$
- Divide & Conquer: $O(n^{\frac{2}{3}} \log n^{\frac{2}{3}}) = O(n^{\frac{2}{3}} \log n) \subset O(n)$
  More samples possible: $O(\frac{n}{\log n} \log \frac{n}{\log n}) \subset O(n)$
- Other approach?

Partition

Karlsruhe Institute of Technology

- Map points to the cells
- Divide the coordinates by cell length and truncate them to integer
- Partition $\mathbb{G} = \{G_1, G_2, ..., G_k\}$ with:

$$\bigcup_{i=1}^{k} G_i = P, G_i \cap G_j = \emptyset, k <= n$$

- $O(n)$ with hashing

Compute

- Compute distances of all pairs
- Running time: O(n)

Theorem

Let $\mathbf{f_i} = |G_i|$ *with* $1 \leq i \leq k$ *be the number of points in i-th non-empty cell and* c, d *some positive constants, then:*

$$\sum_{i=1}^{k} f_i <= c \cdot n \text{ with probability } 1 - \frac{1}{2^{n^d}}$$

Alternative for Sampling

Pseudocode

0 Cell list $C = \emptyset$, Hash table $H = \emptyset$, Result $rst = \infty$
Number of non-empty cells $k = 0$

1 Choose set S of $n^{\frac{2}{3}}$ samples $\quad O(n^{\frac{8}{9}})$
$\delta = min_{x,y \in S, x \neq y}(dist(x, y))$

2 **for** $i = 1$ **to** n **do** $\quad O(n)$
    $c = cell(p_i)$
    if *not* $H.find(c)$ **then**
        $H.add(\{c, idx = \; ++k\})$
    $L[H.find(c).idx].add(p_i)$

3 **for** $i = 1$ **to** k **do** $\quad O(n)$
    foreach j **in** $\{i\} \cup neighbor\_idx(cell_i)$ **do**
        foreach $\{p_i, p_j\}$ **in** $L_i \times L_j$
            $rst = min(rst, dist(p_i, p_j))$

Parallelization

Karlsruhe Institute of Technology

1 Choose set S of $n^{\frac{2}{3}}$ samples
$\delta = min_{x,y \in S, x \neq y}(dist(x, y))$ — $O(n^{\frac{8}{9}})$

2 **for** $i = 1$ **to** n **do**
  $c = cell(p_i)$ — $O(n)$
  if *not* $H.find(c)$ **then**
    $H.add(\{c, idx = ++k\})$
  $L[H.find(c).idx].add(p_i)$

3 **for** $i = 1$ **to** k **do**
  foreach j **in** $neighbor\_idx(cell_i)$ **do** — $O(n)$
    foreach $\{p_i, p_j\}$ **in** $L_i \times L_j$
      $rst = min(rst, dist(p_i, p_j))$

Proof Idea

Karlsruhe Institute of Technology

Theorem

Let $\mathbf{f_i} = |G_i|$ *with* $1 \leq i \leq k$ *be the number of points in* i*-th non-empty cell and* c, d *some positive constants, then:*

$$\sum_{i=1}^{k} f_i <= c \cdot n \text{ with probability } 1 - \frac{1}{2^{n^d}}$$

Proof Idea

Sampling Lemma

fa

Proof Idea

Lemma 2

fa

Proof Idea

Lemma 3

fa

Reference I

[BS76] Jon Louis Bentley and Michael Ian Shamos. "Divide-and-conquer in multidimensional space". In: *Proceedings of the eighth annual ACM symposium on Theory of computing*. STOC '76. Hershey, Pennsylvania, USA: ACM, 1976, pp. 220–230. DOI: 10.1145/800113.803652. URL: http://doi.acm.org/10.1145/800113.803652.

[CG89] Benny Chor and Oded Goldreich. "On the power of two-point based sampling". In: *Journal of Complexity* 5.1 (1989), pp. 96–106.

[Die+97] Martin Dietzfelbinger et al. "A reliable randomized algorithm for the closest-pair problem". In: *Journal of Algorithms* 25.1 (1997), pp. 19–51.

Reference II

[FH78] Steven Fortune and John E Hopcroft. *A note on Rabin's nearest-neighbor algorithm*. Tech. rep. Cornell University, 1978.

[KM95] Samir Khuller and Yossi Matias. "A simple randomized sieve algorithm for the closest-pair problem". In: *Information and Computation* 118.1 (1995), pp. 34–37.

[KT06] J. Kleinberg and E. Tardos. "Algorithm Design". In: Pearson Education, 2006. Chap. 13 Randomized Algorithms.

[Lip09] Richard J. Lipton. *Rabin Flips a Coin*. 2009. URL: `http://rjlipton.wordpress.com/2009/03/01/rabin-flips-a-coin/` (visited on 07/15/2013).

[Rab76] Michael Oser Rabin. "Probabilistic algorithms". In: *Algorithms and Complexity: New Directions and Recent Results*. Ed. by Joseph Frederick Traub. Academic Press, 1976, pp. 21–39.

Reference III

[Web] URL: http://www.cse.ust.hk/tcsc/comp670r/Class_1_Notes.ppt/.

Load Balancing in Step 3

testc

Implementation of Hashing