Chapter 5

Time-Space Tradeoffs for Nearest-Neighbor Search

We develop a method which provides a tradeoff between the space complexity of the data structure and the time complexity of the query algorithm. The idea is to compute in the preprocessing phase a decomposition of the d-dimensional unit cube into simple cells and store for each cell C of the decomposition a set $L_C \subseteq P$ of nearest-neighbor candidates. We guarantee that for each query point in the cell C the corresponding set L_C contains the nearest neighbor to q from the data set P. Given a query point q, the query algorithm determines the cell C containing it, and checks the corresponding set L_C by the brute-force method to find the nearest neighbor to q. The size of the decomposition is controlled by a parameter m, which provides a time-space tradeoff for the data structure.

A summary of the results presented in this chapter has appeared in [38].

5.1 The data structure

We consider the decomposition of $[0,1]^d$ in m^d congruent grid-cells which are d-dimensional cubes of side length $\frac{1}{m}$ for a parameter $m \geq 2$ (see Figure 5.1). We build our data structure $\mathcal{D}$ in the *preprocessing* phase. For each cell C the corresponding set L_C of nearest-neighbor candidates is determined. This set includes all possible nearest neighbors from the data set P to points of the cell C. To compute the set L_C, we determine a suitable cube $W(C)$ around the center s of the cell C, and choose the set L_C to equal $W(C) \cap P$. We compute the side length of the cube $W(C)$ as follows. We determine the nearest neighbor n_C from P to the center s of the cell C. The interior of the cube $C_{s,x}$ around s of side length $x = 2\|n_C - s\|_\infty$, contains no data points from P. We choose the cube $W(C)$ to be the cube around s of side length $x + \frac{2}{m}$ (see Figure 5.2). We show in the following that this cube contains all possible nearest neighbors from the data set P to points of the cell C. For each point $r \in C$ the nearest neighbor $n(r)$ from P to r is contained in $W(C)$, the cube around s of side length $2\|n_C - s\|_\infty + \frac{2}{m}$:

$$\begin{aligned}
\|n(r) - s\|_\infty &\leq \|n(r) - r\|_\infty + \|r - s\|_\infty \leq \|n_C - r\|_\infty + \|r - s\|_\infty \\
&\leq \|n_C - s\|_\infty + 2\|r - s\|_\infty \leq \|n_C - s\|_\infty + \frac{1}{m}.
\end{aligned}$$

Given the cube $W(C)$, the set $L_C = W(C) \cap P$ is determined in time $O(nd)$. Since we can determine n_C for each cell C in time $O(nd)$ the total preprocessing time is $O(m^d \cdot nd)$. The storage size of the data structure $\mathcal{D}$ is $\sum_{\text{cell } C} d \cdot |L_C| = O(m^d \cdot nd)$.

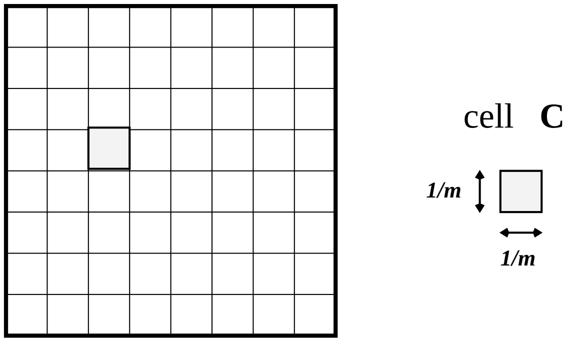

Figure 5.1: Decomposition of the unit cube in congruent cells

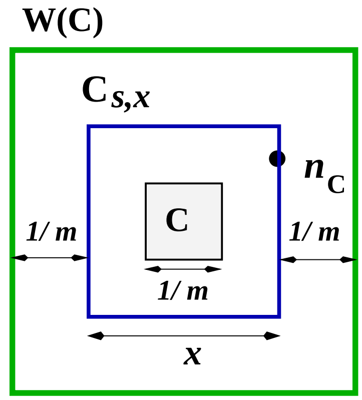

Figure 5.2: Cube W contains all nearest neighbors to cell C from the point set P

The *query algorithm* determines for a query point $q = (q_1, \dots, q_d) \in [0,1]^d$ the cell $C(q)$ containing q. This cell is determined in time $\Theta(d)$ by the values $\lfloor \frac{q_j}{m} \rfloor$, $1 \le j \le d$. Next we determine the nearest neighbor from the set $L_C = W(C) \cap P$ to the query point q, which is also the nearest neighbor from the set P to q. This computation is done by the *brute-force* method in $\Theta(\ d \cdot |\ L_{C(q)}\ |\ )$ time. Instead of the brute-force method we can use the ADAPTIVE METHOD, described in Section 2.2, to determine the nearest neighbor from the set L_C to the query point q.

We determine the expected query time and the expected space complexity of the data structure $\mathcal{D}$, under the assumption that the points of $P = \{p^1, \dots, p^n\}$ are drawn independently at random under uniform distribution. For the expected runtime analysis of the query algorithm we choose the brute-force method to determine the nearest neighbor from the set L_C to q.

5.2 The expected runtime and expected space complexity

To analyze the expected runtime and the expected storage size of the data structure $\mathcal{D}$, we investigate for a fixed cell C of the decomposition the expected number of data points from P contained in the corresponding cube $W(C)$. Let $N(C)$ be the random variable representing the number $|W(C) \cap P|$ of nearest neighbor candidates stored for the cell C.

Let X_C be the continuous random variable for the value $x = 2\|n_C - s\|_\infty$, where n_C is the computed nearest neighbor from P to the center s of the cell C. The value x is the maximum side length of a cube around s containing in its interior no points of P. Note that $x \in [0, 2 - \frac{1}{m}]$. The corresponding cube $W(C)$ of the cell C has center s and side length $x + \frac{2}{m}$. The variable $N(C)$ representing the number of nearest neighbor candidates stored for the cell C depends on the side length variable X_C.

We investigate the conditional expectation $\Psi(X_C) = \mathrm{E}[\ N(C) \mid X_C\ ]$ of $N(C)$ given X_C. The conditional expectation $\Psi(X_C)$ is a random variable. We have $\mathrm{E}[\Psi(X_C)] = E[N(C)]$ by the theorem on conditional expectation (see [34]). Thus,

$$\mathrm{E}[\ N(C)\ ] \ = \ \int_0^{2-\frac{1}{m}} \mathrm{E}[\ N(C) \mid X_C = x\ ] \cdot f_{X_C}(x)\, dx \tag{5.1}$$

where f_{X_C} is the density function of X_C.

We introduce the function $v_C : [0, 2 - \frac{1}{m}] \to [0,1]$ with

$$v_C(x) \ = \ \Pr[\ p \in C_{s,x}\ ]\ ,$$

where p is some random point in $[0,1]^d$ and $C_{s,x}$ is the cube of side length x around the center s of the cell C. Because of uniform distribution the function $v_C(x)$ equals the volume of the box $C_{s,x} \cap [0,1]^d$.

The distribution function F_{X_C} of X_C is given by:

$$F_{X_C}(x) \;=\; \Pr[X_C \le x] \;=\; 1 - \Pr[\, |C_{s,x} \cap P| = 0 \,] \;=\; 1 - (\, 1 - v_C(x)\,)^n \,.$$

Thus, the density function of X_C is given by

$$f_{X_C}(x) \;=\; n \cdot (\, 1 - v_C(x)\,)^{n-1} \cdot v'_C(x) \,. \tag{5.2}$$

The conditional expectation $\mathrm{E}\,[\, N(C) \mid X_C = x \,]$ of $N(C)$ given $X_C = x$ is a function of x. In the following we determine $\mathrm{E}\,[\, N(C) \mid X_C = x \,]$ in terms of the volume $v_C(x)$ of the box $C_{s,x} \cap [0,1]^d$.

Lemma 5.1. *The conditional expectation of $N(C)$ given $X_C = x$ is*

$$E\,[\, N(C) \mid X_C = x \,] \;=\; \begin{cases} 1 + (n-1) \cdot \frac{v_C(x+\frac{2}{m}) - v_C(x)}{1 - v_C(x)} & \text{if } 0 \le v_C(x) < 1 \\ n & \text{if } v_C(x) = 1 \end{cases} \tag{5.3}$$

Proof. Obviously, if $\Pr[p \in C_{s,x}] = v_C(x) = 1$ then the probability for a data point p to be contained in $W(C)$ is also 1, since $W(C)$ is the cube $C_{s,x+\frac{2}{m}}$. Thus, in this case $\mathrm{E}\,[\, N(C) \mid X_C = x \,] = n$.

Now assume $v_C(x) < 1$. The event $\{X_C = x\}$ states that the cube $C_{s,x}$ has at least a data point on its boundary and its interior contains no data points. Let $Y_C \in \{p^1, \ldots, p^n\}$ be the random variable which represents the data point n_C computed for the cell C to be the nearest neighbor of its center s. Since the data points are drawn independently at random we have $\Pr(Y_C = p^i) = \frac{1}{n}$, for all $i \in \{1, \ldots, n\}$. We obtain :

$$\begin{aligned}
\mathrm{E}\,[\, N(C) \mid X_C = x \,] \;&=\; \sum_{i=1}^{n} \Pr\left(p^i \in C_{s,X_C+\frac{2}{m}} \mid X_C = x \right) \\
&=\; \sum_{i=1}^{n} \Big[\Pr(Y_C = p^i) \cdot \Pr\left(p^i \in C_{s,X_C+\frac{2}{m}} \mid X_C = x, Y_C = p^i \right) \\
&\qquad + \Pr(Y_C \neq p^i) \cdot \Pr\left(p^i \in C_{s,X_C+\frac{2}{m}} \mid X_C = x, Y_C \neq p^i \right) \Big] \\
&=\; \sum_{i=1}^{n} \frac{1}{n} \cdot 1 + \left(1 - \frac{1}{n}\right) \cdot \Pr\left(p^i \in C_{s,x+\frac{2}{m}} \mid p^i \notin \mathrm{int} C_{s,x} \right) \\
&=\; 1 + \sum_{i=1}^{n} \left(1 - \frac{1}{n}\right) \cdot \frac{\Pr\left(p^i \in C_{s,x+\frac{2}{m}} \setminus \mathrm{int} C_{s,x} \right)}{\Pr(p^i \notin \mathrm{int} C_{s,x})} \\
&=\; 1 + (n-1) \cdot \frac{v_C(x+\frac{2}{m}) - v_C(x)}{1 - v_C(x)} \,,
\end{aligned}$$

where $\mathrm{int}C_{s,x}$ is the interior of the cube $C_{s,x}$.

$\square$

By (5.1), (5.2) and (5.3) we get:

$$\mathrm{E}\,[\, N(C) \,] \;=\; \int_0^{2-\frac{1}{m}} n \cdot (n-1) \cdot v_C(x + \tfrac{2}{m}) \cdot (1 - v_C(x))^{n-2} \cdot v'_C(x)\; dx \;+ n \cdot \int_0^{2-\frac{1}{m}} h(x)\; dx$$

where $h(x) = v'_C(x) \cdot (1 - v_C(x))^{n-1} + v_C(x) \cdot \big(\, (1 - v_C(x))^{n-1} \big)' = \big(v_C(x) \cdot (1 - v_C(x))^{n-1} \big)'$.

We have $\int_0^{2-\frac{1}{m}} h(x)\, dx = \left[v_C(x) \cdot (1 - v_C(x))^{n-1} \right]_0^{2-\frac{1}{m}} = 0$, since $v_C(0) = 0$ and $v_C(2 - \frac{1}{m}) = 1$. This implies:

$$\mathrm{E}\,[\,N(C)\,] \;=\; \int_0^{2-\frac{1}{m}} n \cdot (n-1) \cdot v_C(x + \tfrac{2}{m}) \cdot (1 - v_C(x))^{n-2} \cdot v_C'(x)\, dx$$

We want to estimate $v_C(x + \frac{2}{m})$ in terms of $v_C(x)$. The probability $v_C(x) \;=\; \Pr[\, p \in W_C^x \,]$ is the product of the side lengths $s_1(x) \leq s_2(x) \leq \ldots \leq s_d(x)$ of the box $C_{s,x} \cap [0,1]^d$.

By (2.3), the side lengths $s_i(x)$ have the following properties:

- $s_j(x) \leq x$ for all $1 \leq j \leq d$,
- $\frac{1}{2} \cdot s_j(x) \leq s_i(x) \leq 2 s_j(x)$ for all $i \neq j \in \{1, \ldots, d\}$.
- $\frac{1}{2} \cdot \lambda(x) \leq s_j(x) \leq 2\lambda(x)$, where $\lambda(x)$ is the geometric mean of $s_j(x), j = 1, \ldots, d$.

The side lengths $s_j(x + \frac{2}{m})$ ($j = 1, \ldots, d$) of the cube $C_{s,x+\frac{2}{m}}$ fulfill

$$\min\left\{\, 1\,,\; s_j(x) + \tfrac{1}{m} \,\right\} \;\leq\; s_j(x + \tfrac{2}{m}) \;\leq\; s_j(x) + \tfrac{2}{m}\;. \tag{5.4}$$

We refer for illustration to Figure 5.3.

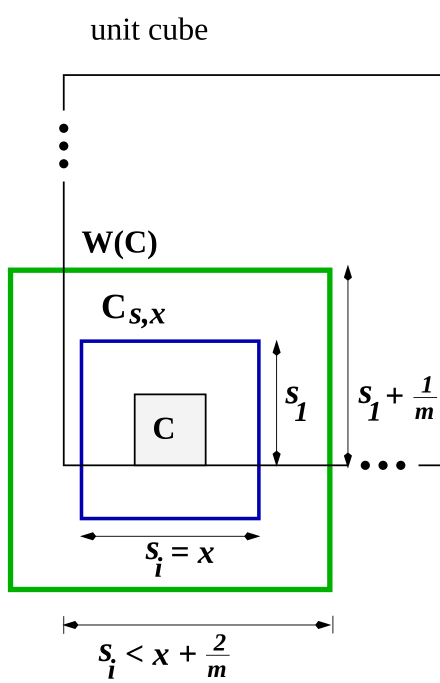

Figure 5.3: Side lengths of $W(C) \cap [0,1]^d$

Lemma 5.2. *Let s_j be the side lengths of the cube $C_{s,x}$ and let λ be their geometric mean. Given $a \in \mathbb{R}$, $a > 0$ we have*

$$\left(\lambda + \tfrac{1}{2}a\right)^d \;\leq\; \prod_{j=1}^{d} (s_j + a) \;\leq\; (\lambda + 2a)^d\;.$$

Proof. Let $\mathcal{D} = \{1, \ldots, d\}$ be the set of dimensions. For some subset $S_{n-k} \subseteq \mathcal{D}$ of size $n-k$, let denote by $\pi(S_{n-k}) = \prod_{j \in S_{n-k}} s_j$ the product of the corresponding side lengths s_j, $j \in S_{n-k}$. We have

$$\prod_{j=1}^{d} (s_j + a) \;=\; \sum_{k=0}^{d} a^k \cdot \sum_{\substack{S_{n-k} \subseteq \mathcal{D} \\ |S_{n-k}| = n-k}} \pi(S_{n-k}) \;, \tag{5.5}$$

Consider a side length s_l, where $l \in S_{n-k}$ for some subset S_{n-k}. By the properties of the side lengths, we obtain:

$$\frac{\lambda^d}{2^k \cdot \pi(S_{n-k})} \;=\; \frac{1}{2^k} \pi(\mathcal{D} \setminus S_{n-k}) \;\leq\; (s_l)^k \;\leq\; 2^k \pi(\mathcal{D} \setminus S_{n-k}) \;=\; \frac{2^k \lambda^d}{\pi(S_{n-k})} \,.$$

which implies

$$\frac{\lambda^{d(d-k)}}{2^{k(d-k)} \cdot (\pi(S_{n-k}))^{d-k}} \;\leq\; (\pi(S_{n-k}))^k \;\leq\; \frac{2^{k(d-k)} \lambda^{d(d-k)}}{(\pi(S_{n-k}))^{d-k}} \,.$$

This implies together with $2^{\frac{k(d-k)}{d}} \leq 2^k$:

$$\frac{1}{2^k} \cdot \lambda^{d-k} \;\leq\; \pi(S_{n-k}) \;\leq\; 2^k \cdot \lambda^{d-k}$$

By (5.5), we obtain

$$\left(\lambda + \tfrac{1}{2} a\right)^d \;=\; \sum_{k=0}^{d} \binom{d}{k} a^k \cdot \frac{1}{2^k} \cdot \lambda^{d-k} \;\leq\; \prod_{j=1}^{d} (s_j + a) \;\leq\; \sum_{k=0}^{d} \binom{d}{k} a^k \cdot 2^k \cdot \lambda^{d-k} \;=\; (\lambda + 2a)^d$$

$\square$

Lemma 5.2 and (5.4) provide an upper bound on $v_C(x + \frac{2}{m})$ in terms of $v_C(x)$:

$$v_C\left(x + \tfrac{2}{m}\right) \;\leq\; \left(\sqrt[d]{v_C(x)} + \tfrac{4}{m}\right)^d \,. \tag{5.6}$$

Let $d'(C, x) = \max\{j \;:\; s_j(x) \leq 1 - \frac{1}{m}\}$. Note that for a given cell C, $d'(C, X_C) \in \{1, \ldots, d\}$ is a random variable depending on X_C. If $s_l(x) > 1 - \frac{1}{m}$, $l \in \{1, \ldots, d\}$ then the side length $s_l(x + \frac{2}{m})$ of the cube $W(C)$ equals 1. Let denote $v'(x) = \prod_{j=1}^{d'} s_j(x)$, if $d'(C, x) = d' \in \{1, \ldots, d\}$. By (5.4), we obtain

$$\left(\sqrt[d']{v'_C(x)} + \tfrac{1}{2m}\right)^{d'} \;\leq\; v_C\left(x + \tfrac{2}{m}\right) \;\leq\; \left(\sqrt[d']{v'_C(x)} + \tfrac{4}{m}\right)^{d'} \qquad \text{if } d'(C, x) = d' \,. \tag{5.7}$$

We focus on the upper bound obtained in (5.6) and we get

$$\int_0^{2 - \frac{1}{m}} n \cdot (n-1) \cdot v_C(x + \tfrac{2}{m}) \cdot (1 - v_C(x))^{n-2} \cdot v'_C(x) \, dx \;\leq\; \mathcal{F}(n, \tfrac{4}{m}) \tag{5.8}$$

where

$$\begin{aligned}
\mathcal{F}(n, a) \;&=\; \int_0^{2 - \frac{1}{m}} n \cdot (n-1) \cdot \left(\sqrt[d]{v_C(x)} + a\right)^d \cdot (1 - v_C(x))^{n-2} \cdot v'_C(x) \, dx \\
&=\; \int_0^1 n \cdot (n-1) \cdot (\sqrt[d]{y} + a)^d \cdot (1 - y)^{n-2} \; dy \\
&=\; \int_0^1 n \cdot (n-1) \cdot \sum_{i=0}^{d} \binom{d}{i} \cdot y^{i/d} \cdot a^{d-i} \cdot (1 - y)^{n-2} \, dy \\
&=\; \sum_{i=0}^{d} n \cdot \binom{d}{i} \cdot a^{d-i} \cdot \int_0^1 y^{i/d} \cdot (1 - y)^{n-2} \cdot (n-1) \, dy
\end{aligned} \tag{5.9}$$

The following lemma solves a useful integral:

Lemma 5.3. *Consider $d \in \mathbb{N}$, $d \geq 2$, $j \in \mathbb{N}$, $j \geq 1$ and $i \in \{1 \ldots, d\}$. The following equality holds:*

$$\int_0^1 y^{i/d} \cdot j \cdot (1-y)^{j-1}\, dy = \binom{j+i/d}{j}^{-1}.$$

Proof. Let $f(j) = \int_0^1 y^{i/d} \cdot j \cdot (1-y)^{j-1}\, dy$. We have:

$$\begin{aligned}
\left(1+\frac{i/d}{j}\right) \cdot f(j) &= f(j) + \int_0^1 (y^{i/d})' \cdot y \cdot (1-y)^{j-1}\, dy \\
&= \left(f(j) - \int_0^1 y^{i/d} \cdot (1-y)^{j-1}\, dy \right) + \left(\int_0^1 y^{i/d} \cdot y \cdot (1-y)^{j-2}(j-1)\, dy \right) \\
&= \int_0^1 y^{i/d} \cdot (1-y)^{j-1}(j-1)\, dy + \left(- \int_0^1 y^{i/d} \cdot (1-y)^{j-1}(j-1)\, dy + f(j-1) \right) \\
&= f(j-1)
\end{aligned}$$

Thus, $f(j) = \frac{j}{j+i/d} \cdot f(j-1)$ which together with $f(1) = \int_0^1 y^{i/d}\, dy \;=\; \frac{1}{1+i/d}$ implies:

$$f(j) = \frac{j}{j+i/d} \cdot \frac{j-1}{j-1+i/d} \cdot \ldots \cdot \frac{1}{1+i/d} = \binom{j+i/d}{j}^{-1}$$

$\square$

By Lemma 5.3, we get:

$$\int_0^1 y^{i/d} \cdot (1-y)^{n-2} \cdot (n-1) \;=\; \binom{n-1+i/d}{n-1}^{-1} \tag{5.10}$$

Proposition 5.1. *Consider $d \in \mathbb{N}$, $d \geq 2$, $j \in \mathbb{N}$, $j \geq 1$ and $i \in \{1 \ldots, d\}$. The following inequalities hold:*

$$\frac{(1/e)^{i/d}}{(\sqrt[d]{n}\,)^i} \;\leq\; \binom{n-1+i/d}{n-1}^{-1} \;\leq\; \frac{e^{i/d}}{(\sqrt[d]{n}\,)^i}\,. \tag{5.11}$$

Proof. We have $e^x \;\leq\; (1+\frac{x}{k})^{k+1}$, for all $x \in [0,1]$ and $k \in \mathbb{N}$, $k \geq 1$. This is based on the fact that $h(x) : [0,1] \to \mathbb{R}$ with $h(x) = (k+1)\ln(1+\frac{x}{k}) - x$ fulfills $h(0) = 0$ and is monotone increasing on $x \in [0,1]$, since $h'(x) = \frac{1-x}{(k+x)(k+1)} \geq 0$ for $x \in [0,1]$. Thus, $e^{\frac{i}{d} \cdot \frac{1}{j+1}} \leq \left(1+\frac{i/d}{j}\right)$ for $i \in \{1 \ldots, d\}$ and we obtain:

$$\binom{n-1+i/d}{n-1}^{-1} \;=\; \prod_{j=1}^{n-1} \frac{j}{j+i/d} \;\leq\; e^{-\frac{i}{d}\sum_{j=1}^{n-1}\frac{1}{j+1}} \;<\; e^{-\frac{i}{d}(\ln n - 1)} \;=\; \frac{e^{i/d}}{(\sqrt[d]{n}\,)^i}\,,$$

which proves the right inequality of the proposition.

Now, consider $h(x) : [0,1] \to \mathbb{R}$ with $h(x) = \sum_{j=1}^{n-1} \ln(j+x) - \sum_{j=1}^{n-1} \ln j - x\ln n - x$. Obviously, $h(0) = 0$ and $h'(x) = \sum_{j=1}^{n-1} \frac{1}{j+x} - \ln n - 1 < 0$. Thus, $h(x) \leq 0$ for all $x \in [0,1]$, and with this $h(i/d) < 0$ which proves the left inequality of the proposition. $\square$

By (5.11), (5.10) and (5.9), we get:

$$\left(\frac{1}{\sqrt[d]{e}}+a\sqrt[d]{n}\right)^d \;\le\; \mathcal{F}(n,a) \;\le\; \left(\sqrt[d]{e}+a\sqrt[d]{n}\,\right)^d$$

Thus, by (5.8) we obtain

$$\mathrm{E}[N(C)] \;\le\; \left(\sqrt[d]{e}+\frac{4\sqrt[d]{n}}{m}\right)^d \;. \tag{5.12}$$

By (5.7), we obtain also a lower bound $\left(\frac{1}{\sqrt[d]{e}}+\frac{\sqrt[d]{n}}{2m}\right)^d \;\le\; \mathrm{E}\,[\,N(C)\mid d'(C,X_C)=d\,]$ under the condition $d'(C,X_C)=d$.

We summarize the results in the following theorem.

Theorem 5.1. *The expected asymptotic runtime $t=t(m)$ of the query algorithm and the expected asymptotic storage size $s=s(m)$ of the data structure $\mathcal{D}=\mathcal{D}(m)$ are given by:*

$$t(m) \;=\; O\left(d\cdot\left(\sqrt[d]{e}+\frac{4\sqrt[d]{n}}{m}\right)^d\right) \tag{5.13}$$

$$s(m) \;=\; O\left(d\cdot m^d\cdot\left(\sqrt[d]{e}+\frac{4\sqrt[d]{n}}{m}\right)^d\right) \tag{5.14}$$

respectively, where m is a parameter.

The time-space tradeoff between the expected storage size $s(m)$ of the data structure and the expected running time $t(m)$ of the query algorithm is controlled by the parameter m. As an example, let $m=m_* := \frac{4d\sqrt[d]{n}}{\sqrt[d]{e}\cdot\ln\ln n}$ and we get:

$$t(m_*) \;=\; O\left(d\cdot\left(1+\frac{4\sqrt[d]{n}}{\sqrt[d]{e}\cdot m_*}\right)^d\right) \;=\; O\left(d\cdot\left(1+\frac{\ln\ln n}{d}\right)^d\right) \;=\; O\left(d\log n\right)\;,$$

$$s(m_*) \;=\; O\left(m_*^d\cdot t(m_*)\right) \;=\; O\left(\left(\frac{4d}{\ln\ln n}\right)^d nd\log n\right).$$

If $d'(C,X_C)=d$ for all cells C, we obtain $t=\Theta(\,d\cdot(1+\frac{\sqrt[d]{n}}{m})^d\,)$ and $s=\Theta(\,d\cdot m^d\cdot(1+\frac{\sqrt[d]{n}}{m})^d\,)$. This provides $m=\Theta(\sqrt[d]{\frac{s}{d}}-\sqrt[d]{n})$ and the tradeoff $t \;=\; \Theta\left(\frac{s}{(\,\sqrt[d]{s}-2\sqrt[d]{nd}\,)^d}\right)$.

The data structure can be easily extended to work in the external-memory model of computation, by storing for each cell C the set L_C of nearest-neighbor candidates in contiguous locations in the external memory.