arXiv:2304.14643v2 [cs.CG] 2 May 2023

弗雷歇距离下多边形曲线的近似最近邻

Siu-WingCheng ！ 香港科技大学计算机科学与工程系，香港，中国

Haoqiang Huang
香港科技大学计算机科学与工程系，香港，中国

摘要

我们针对实数集中的 n 条多边形曲线，在弗雷歇距离下提出了 κ-近似最近邻（ANN）数据结构，其中 $\kappa \in \{1+\varepsilon, 3+\varepsilon\}$ 且 $d \geq 2$。我们假设每条输入曲线至多有 m 个顶点，每条查询曲线至多有 k 个顶点，$k \ll m$，并且给定 k 用于预处理。对于 $(1+\varepsilon)$-ANN，查询时间为 $\tilde{O}(k(mn)^{0.5+\varepsilon}/\varepsilon^d + k(d/\varepsilon)^{O(dk)})$；对于 $(3+\varepsilon)$-ANN，查询时间为 $\tilde{O}(k(mn)^{0.5+\varepsilon}/\varepsilon^d)$。两种情况下，空间和期望预处理时间均为 $\tilde{O}(k(mnd^d/\varepsilon^d)^{O(k+1/\varepsilon^2)})$。在二维和三维中，我们将查询时间改进为：对于 $(1+\varepsilon)$-ANN，为 $O(1/\varepsilon)^{O(k)} \cdot \tilde{O}(k)$；对于 $(3+\varepsilon)$-ANN，为 $\tilde{O}(k)$。两种情况下，空间和期望预处理时间均改进为 $O(mn/\varepsilon)^{O(k)} \cdot \tilde{O}(k)$。为便于表述，我们将界中仅依赖于 d 的因子视为 $O(1)$。大-$\tilde{O}$ 记号中隐藏的多对数因子的幂取决于 d。

2012年 美国计算机协会学科分类计算理论 → 计算几何

关键词和短语多边形曲线、弗雷歇距离、近似最近邻

数字对象标识符10.4230/LIPIcs.ICALP.2023.23

1 引言

给定一组轨迹，最近邻问题旨在高效地找出与查询轨迹最相似的那条轨迹。轨迹通常表示为多边形曲线，而最近邻问题在各种应用中都十分常见 [20, 21, 22]。

针对多边形曲线，人们提出了各种相似性度量。我们关注弗雷歇距离 [3]，近年来它受到了广泛关注。其定义如下。曲线 τ 的一个参数化是一个函数 $\rho: [0,1] \to \mathbb{R}^d$，当 t 从 0 增加到 1 时，点 $\rho(t)$ 沿单调方向从 τ 的起点移动到终点。对于两个不同的值 t_1 和 t_2，可能有 $\rho(t_1) = \rho(t_2)$。曲线 τ 和 σ 的两个参数化 ρ 和 ϱ 分别诱导出一个匹配 M：对于所有 $t \in [0,1]$，M 将 $\rho(t)$ 与 $\varrho(t)$ 匹配。一个点可以与多个点匹配。在 M 下，τ 与 σ 之间的距离为 $d_{\mathcal{M}}(\tau,\sigma) = \max_{t\in[0,1]} d(\rho(t)\ \varrho(t))$，其中 $d(\cdot,\cdot)$ 表示欧几里得距离。弗雷歇距离为 $d_F(\tau,\sigma) = \min_{\mathcal{M}} d_{\mathcal{M}}(\tau,\sigma)$。我们将使距离达到最小的匹配称为弗雷歇匹配。

设 $T = \{\tau_1, \ldots, \tau_n\}$ 为一个包含 n 条多边形曲线的集合，每条曲线至多有 m 个顶点。给定任意值 $\kappa \geq 1$，κ-近似最近邻（ANN）问题旨在构建一种数据结构，使得对于任意查询曲线 σ，我们都能快速报告一条曲线 $\tau_l \in T$，满足 $d_F(\sigma,\tau_l) \leq \kappa \cdot \min_{\tau_i \in T} d_F(\sigma,\tau_i)$。我们假设每条查询曲线至多有 k 个顶点，且 k 在预处理阶段已知。在已有文献中，如果 $k = m$，则称为对称版本；如果 $k < m$，则称为非对称版本。如果查询曲线由用户手绘得到，那么很可能 $k \ll m$，这正是我们设计数据结构所针对的应用场景。我们将相关的 (κ,δ)-ANN 问题定义如下：对于任意查询曲线，我们报告“否”或一条曲线$\tau_l \in T$，满足 $d_F(\sigma,\tau_l) \leq \kappa\delta$；如果我们报告“否”，则必须满足 $\min_{\tau_i \in T} d_F(\sigma,\tau_i) > \delta$。

关于离散弗雷歇距离下的近似最近邻问题，已有许多研究成果d_F。离散弗雷歇距离将 d_F 的定义限制为满足以下条件的参数化 ρ 和 ϱ：τ的每个顶点˜都与 σ的至少一个顶点匹配，反之亦然。因此，$d_F(\tau,\ \sigma) \leq d_F(\tau,\ \sigma)$。

© Siu-Wing Cheng 和 Haoqiang Huang；依据知识共享署名4.0许可协议授权。第50届自动机、语言与编程国际学术会议（ICALP 2023）。编辑：Uriel Feige；文章编号：23；页码：23:1–23:23。莱布尼茨国际信息学会议论文集，Dagstuhl城堡—莱布尼茨信息学中心，Dagstuhl出版社，德国

LIPICS

可能出现 $d_F(\tau,\ \sigma) \ll d_F(\tau,\ \sigma)$的情况；例如， σ是一条很长的水平线段，而 τ是 σ 附近的一条平行线段，其中央多了一个顶点。 d_F 的优点在于，可以使用一个简单的动态规划算法 [12]来计算。

Indyk 和 Motwani [18] 以及 Har-Peled [15] 证明，对于度量空间中的点，(κ, δ)-近似最近邻问题的解可以导出 $\kappa(1+O(\varepsilon))$-近似最近邻问题的解。该结果在期刊版本 [16]中得到了简化。该方法具有足够的通用性，也适用于 d_F 和 d_F下的多边形曲线。第2节中的定理 1 给出了本文背景下的确定性结果；该归约会使空间和查询时间增加多对数因子。如果采用失败概率为 f 的概率型(κ, δ)-近似最近邻解，定理 1中的界同样成立，且该近似最近邻解的失败概率为 $O(f \log n)$。

Indyk [17] 提出了首个在 d_F下的(κ, δ)-近似最近邻解，其中 $\kappa = O(\log\ m + \log\log\ n)$；该解适用于 $k = m$ 且顶点来自离散点集 X的情形。它使用$O\big(|X|^{\sqrt{m}}(m^{\sqrt{m}}n)^2\big)$ 空间，并在 $O\big(m^{O(1)} \log n\big)$ 时间内回答查询。[1] Driemel 和Silverstri [11] 针对 d_F 提出了失败概率为$1/n$的概率型(κ, δ)-近似最近邻解；对于$k = m$的情形，它们实现了以下$(\kappa$，查询时间，空间)组合： $\big(4d^{3/2}m$、 $O(m)\ O(n \log\ n + mn)\big)$、 $\big(4d^{3/2}$、 $O(2^{4dm}m\log\ n)\ O(2^{4md}n \log\ n + mn)\big)$，以及任意整数 $t \geq 1$下的$\big(4d^{3/2}m/t$、 $O(2^{2t}m^t \log\ n)\ O(2^{2t}m^{t-1}n\log\ n + mn)\big)$。后来，两个研究团队将近似比降至 $1+\varepsilon$。Filtser等人 [14] 提出了两个在 d_F下的确定性$(1+\varepsilon, \delta)$-近似最近邻数据结构；其中一个在 $O(kd)$时间内回答查询，使用 $n \cdot O(\frac{1}{\varepsilon})^{kd}$空间，且期望预处理时间为 $O(mn(d\log\ m + O(\frac{1}{\varepsilon})^{kd}))$；另一个在 $O(kd\log\ \frac{dkn}{\varepsilon})$时间内回答查询，使用 $n \cdot O(\frac{1}{\varepsilon})^{kd}$空间，且最坏情况预处理时间为 $O(mn\log\frac{n}{\varepsilon} \cdot (d\log\ m + O(\frac{1}{\varepsilon})^{kd}))$。Emiris 和 Psarros [13]针对 d_F下的 $k = m$情形，构造了失败概率为 $1/2$ 的概率型$(1+\varepsilon)$-近似最近邻和$(1+\varepsilon, \delta)$-近似最近邻数据结构。$(1+\varepsilon)$-近似最近邻数据结构在 $\tilde{O}(d2^{4m}m^{O(1/\varepsilon)})$时间内回答查询，使用$\tilde{O}(dm^2n) \cdot (2 + d/\log\ m)^{O(dm^{O(1/\varepsilon)}\log(1/\varepsilon))}$空间和预处理时间。$(1+\varepsilon, \delta)$-近似最近邻数据结构在 $O(d2^{4m}\log n)$时间内回答查询，使用 $O(dn)\ +(mn)^{O(m/\varepsilon^2)}$空间和预处理时间。将查询时间、空间和预处理时间增加一个 $O(\log n)$因子后，可将失败概率降至 $1/n$ 。

大多数已知的 d_F 结果都是针对 $\mathbb{R}$ 的。对于 $\mathbb{R}$ （时间序列）中的曲线，Driemel 和 Psarros [10] 提出了首个满足 d_F 的 $(\kappa,\ \delta)$-近似最近邻数据结构，其 $(\kappa$, 查询时间, 空间) 的组合如下： $\big(5+\varepsilon,\ O(k)\ O(mn)\ +n \cdot O(\frac{1}{\varepsilon})^k\big)$, $\big(2+\varepsilon,\ O(2^k k)\ O(mn)\ +n \cdot O(\frac{m}{k\varepsilon})^k\big)$, 以及 $\big(24k+1,\ O(k\log n)\ O(n\log\ n + mn)\big)$。最后一种是概率型的，其失败概率为 $1/\text{poly}(n)$。之后，Bringman 等人 [5] 得到了改进的解，其 $(\kappa$, 查询时间, 空间) 的组合如下： $\big(1+\varepsilon,\ O(2^k k)\ n \cdot O(\frac{m}{k\varepsilon})^k\big)$, $(2+\varepsilon,\ O(k)\ n \cdot O(\frac{m}{k\varepsilon})^k)$, $\big(2+\varepsilon,\ O(2^k k)\ O(mn)\ +n \cdot O(\frac{1}{\varepsilon})^k\big)$, $\big(2+\varepsilon,\ O(\frac{1}{\varepsilon})^{k+2},\ \overline{O}(mn)\big)$, 以及$\big(3+\varepsilon,\ O(k)\ \overline{O}(mn)\ +n \cdot O(\frac{1}{\varepsilon})^k\big)$。他们还给出了以正交向量假设为条件的下界：对于所有 $\varepsilon,\ \varepsilon' \in (0,\ 1)$，当 $1 \ll k \ll \log\ n$ 且 $m = kn^{\Theta(1/k)}$ 时，在 $\mathbb{R}$ 中无法实现组合 $\big(2-\varepsilon,\ O(n^{1-\varepsilon'}),\ \text{poly}(n)\big)$；当 $m = k = \Theta(\log\ n)$ 时，在 $\mathbb{R}$ 中无法实现组合$\big(3-\varepsilon,\ O(n^{1-\varepsilon'}),\ \text{poly}(n)\big)$；当 $1 \ll k \ll \log\ n$ 且 $m = kn^{\Theta(1/k)}$ 时，在 $\mathbb{R}^2$ 中无法实现组合 $\big(3-\varepsilon,\ O(n^{1-\varepsilon'}),\ \text{poly}(n)\big)$。Mirzanezhad [19] 描述了一种针对 $\mathbb{R}^d$ 的 $(1+\varepsilon,\ \delta)$-近似最近邻数据结构，其在 $O(kd)$ 时间内回答查询，并使用 $O(n \cdot \max\{\sqrt{d}/\varepsilon, \sqrt{dD}/\varepsilon^2\}^{dk})$ 空间，其中 D 是输入曲线集合的直径。若未给定 k，则近似比和空间分别增至 $5+\varepsilon$ 和 $n \cdot O(\frac{1}{\varepsilon})^{dm}$。第一个解的空间复杂度中， D 没有上界。为便于比较，我们在表 1 中总结了所有这些已有结果。

我们在 d_F 下，为 $\kappa \in \{1+\varepsilon, 3+\varepsilon\}$ 和 $d \geq 2$ 开发 $(\kappa,\ \delta)$ 近似最近邻数据结构，适用于 $\mathbb{R}^d$。我们

[1] 文献 [17]中也给出了一种权衡。

距离	空间	查询时间	近似
连续弗雷歇距离 , $\mathbb{R}$	$O(mn) + n \cdot O\left(\frac{1}{\varepsilon}\right)^k$	$O(k)$	$(5+\epsilon,\delta)$-近似最近邻 [10]
	$O(mn) + n \cdot O\left(\frac{m}{k\varepsilon}\right)^k$	$O(2^k k)$	$(2+\epsilon,\delta)$-近似最近邻 [10]
	$O(n$对数 $n+mn)$	$O(k \log n)$	$(24k+1,\ \delta)$-近似最近邻 [10][a]
	$n \cdot O\left(\frac{m}{k\varepsilon}\right)^k$	$O(2^k k)$	$(1+\varepsilon,\ \delta)$-近似最近邻 [5]
	$n \cdot O\left(\frac{m}{k\varepsilon}\right)^k$	$O(k)$	$(2+\varepsilon,\ \delta)$-近似最近邻 [5]
	$O(mn) + n \cdot O\left(\frac{1}{\varepsilon}\right)^k$	$O(2^k k)$	$(2+\varepsilon,\ \delta)$-近似最近邻 [5]
	$O(mn)$	$O\left(\frac{1}{\varepsilon}\right)^{k+2}$	$(2+\varepsilon,\ \delta)$-近似最近邻 [5]
	$O(mn) + n \cdot O\left(\frac{1}{\varepsilon}\right)^k$	$O(k)$	$(3+\varepsilon,\delta)$-近似最近邻 [5]
连续弗雷歇距离 , $\mathbb{R}^d$	$O\left(n \cdot$最大值$\{\sqrt{d}/\varepsilon, \sqrt[4]{dD/\varepsilon^2}\}^{dk}\right)$	$O(kd)$	$(1+\varepsilon,\delta)$-近似最近邻 [19]
	$n \cdot O\left(\frac{1}{\varepsilon}\right)^{dm}$	$O(kd)$	$(5+\varepsilon,\ \delta)$-近似最近邻 [19]
	$\tilde{O}\left(k(mnd^d/\varepsilon^d)_{O(k+1/\varepsilon^2)}\right)$	$\tilde{O}\left(k(mn)^{0.5+\varepsilon}/\varepsilon^d + k(d/\varepsilon)^{O(dk)}\right)$	$(1+\varepsilon,\ \delta)$-近似最近邻，定理 9
	$\tilde{O}\left(k(mnd^d/\varepsilon^d)_{O(k+1/\varepsilon^2)}\right)$	$\tilde{O}\left(k(mn)^{0.5+\varepsilon}/\varepsilon^d\right)$	$(3+\varepsilon,\ \delta)$-近似最近邻，定理 11
连续弗雷歇距离，$\mathbb{R}^2$ 以及 $\mathbb{R}^3$	$O\left(\frac{1}{\varepsilon}\right)^{4d(k-1)+1}(mn)^{4(k-1)}k\log^2 n$	$O\left(\frac{1}{\varepsilon}^{2d\ (k-2)}\right)k$对数$\frac{mn}{\varepsilon}$对数 n	$(1+\varepsilon,\delta)$-近似最近邻，定理 9
	$O\left(\frac{1}{\varepsilon}\right)^{2d(k-1)+1}(mn)^{2(k-1)}k\log^2 n$	$O\left(k$对数$\frac{mn}{\varepsilon}$对数$n\right)$	$(3+\varepsilon,\ \delta)$-近似最近邻，定理 11
离散 Fréchet 距离，$\mathbb{R}^d$	$O\left(\lvert X\rvert^{\sqrt{m}}(m^{\sqrt{m}}n)^2\right)$	$O\left(m^{O(1)}$对数$n\right)$	$(O(\log\ m+\log\log\ n)\ \delta)$-近似最近邻 [17]
	$O(n$对数 $n+mn)$	$O(m)$	$(4d^{3/2}m,\delta)$-近似最近邻 [11]
	$O(2^{4md}n$对数 $n+mn)$	$O(2^{4dm}m$对数 $n)$	$(4d^{3/2},\ \delta)$-近似最近邻 [11]
	$O(2^{2t}m^{t-1}n$对数 $n+mn)$	$O(2^{2t}m^t$ 对数$n)$	$(4d^{3/2}m/t,\ \delta)$-近似最近邻 [11]
	$n \cdot O\left(\frac{1}{\varepsilon}\right)^{kd}$	$O(kd)^b$	$(1+\varepsilon,\ \delta)$-近似最近邻 [14]
	$O(dn) + (mn)_{O(m/\varepsilon^2)}$	$O\left(d2^{4m}\log n\right)$	$(1+\varepsilon,\ \delta)$-近似最近邻[c] [13]

[a] 失败概率为 $1/\mathrm{poly}(n)$。 [b] 通过使用哈希表实现字典，可达到相应的查询时间。使用字典树实现字典时，查询时间为 $O(kd\log\frac{dkn}{\varepsilon})$。 [c] 失败概率为 1 2。 **表 1** 我们提出的数据结构与先前结果的比较。

假设每条查询曲线最多有 k 个顶点，$k \ll m$，并且已给定 k 用于预处理。为简化界限，我们假设本文中 $k \geq 3$ 。我们有三个设计目标。首先，查询时间关于 mn 呈次线性。其次，空间复杂度仅取决于输入参数。第三，空间复杂度既不与 $\min\{m^{\Omega(d)}, n^{\Omega(d)}\}$ 成正比，也不随 $\min\{m, n\}$ 呈指数增长。若能消除对 d 的所有指数级依赖当然更好，但目前我们还做不到。

对于 $(1+\varepsilon,\delta)$-近似最近邻，我们实现了 $\tilde{O}(k(mn)^{0.5+\varepsilon}/\varepsilon^d + k(d/\varepsilon)^{O(dk)})$ 的查询时间。对于 $(3+\varepsilon,\delta)$-近似最近邻，我们消除了对 k 的指数级依赖，并获得 $\tilde{O}(k(mn)^{0.5+\varepsilon}/\varepsilon^d)$ 的查询时间。在这两种情况下，空间和期望预处理时间均为 $\tilde{O}(k(mnd^d/\varepsilon^d)^{O(k+1/\varepsilon^2)})$。为便于表述，我们将界中仅依赖于 d 的任何因子视为 $O(1)$，~。大-O 表示法中隐藏的多对数因子的幂取决于 d。在二维和三维中，我们将查询时间改进为：对于 ~$(1+\varepsilon,\delta)$-近似最近邻，为 $O(1/\varepsilon)^{O(k)} \cdot \tilde{O}(k)$；对于 $(3+\varepsilon,\delta)$-近似最近邻，为 $O(k)$。在这两种情况下，空间和期望预处理时间均改进为 $O(mn/\varepsilon)^{O(k)} \cdot \tilde{O}(k)$。利用 [16] （第2节中的定理 1）中的归约，我们得到 $(1+\varepsilon)$-近似最近邻和 $(3+\varepsilon)$-近似最近邻数据结构，代价是查询时间和空间分别增加一个 $O(\log\ n)$ 因子和一个 $O(\frac{1}{\varepsilon}\log^2 n)$ 因子。更精确的界见定理 9 和定理 11。

我们的 $(1+\varepsilon,\delta)$-近似最近邻结果基于两个新思路。首先，我们提出一种新颖的查询曲线编码，该编码基于输入顶点邻域中的局部网格。其次，我们建立了与近似线段射击问题的联系，并高效地解决了该问题。我们将在第 2 节和第 4 节介绍这些思路。我们的 $(3+\varepsilon)$-近似最近邻结果则是通过简化编码得到的。我们将在第 3 节介绍这一结果。

我们采用字长 RAM 模型。我们用 $(v_{i,\ 1}, \ldots\ldots, v_{i,\ m})$ 表示 τ_i 从起点到终点的顶点序列——τ_i 的方向是从 $v_{i,\ 1}$ 到 $v_{i,\ m}$。我们用 $\tau_{i,\ a}$ 表示边 $v_{i,\ a}v_{i,\ a+1}$。对于任意两个点 x，$y \in \tau_i$，若 x 未沿着 出现在 v_i之后，我们称 $x \leq_{\tau_i} y$

沿着 τ_i位于 y 之后，而 $\tau_i[x,y]$ 表示 x 与 y之间的子曲线。给定两个子集$X,Y \subseteq \tau_i$，当且仅当对每个点 $x \in X$ 及每个点 $y \in Y$，$x \leq_{\tau_i} y$时，$X \leq_{\tau_i} Y$成立。以原点为中心、半径为 r 的球记作 B_r。给定两个子集 $X,Y \subset \mathbb{R}^d$，$d(X,Y) = \min_{x\in X, y\in Y} d(x,y)$；它们的闵可夫斯基和为 $X \oplus Y = \{x+y\colon x \in X, y \in Y\}$；若 $X = \{p\}$，为简便起见，记作 $p \oplus Y$。对于任意 x，$y \in \mathbb{R}^d$表示从 x 到 y的有向线段，而 aff(xy) 是 xy 的有向支撑直线，其方向与 xy一致。

2(1 + O(ε) δ) (()) (())-近似最近邻

Har-Peled 等人 [16, 定理 2.10] 证明了从 $(1+\varepsilon)$-近似最近邻问题到 $(1+\varepsilon,\ \delta)$-近似最近邻问题的归约。尽管该结果针对的是度量空间中的点，并采用了用于 $(1+\varepsilon,\ \delta)$-近似最近邻问题的概率数据结构，但该方法具有足够的通用性，能够适用于 $\mathbb{R}^d$ 中基于 d_F 或 $\tilde{d}_F$ 的多边形曲线，以及任何用于 $(1+\varepsilon,\ \delta)$-近似最近邻问题的确定性解。下面我们将在本文的背景下重新表述他们的结果。

▶定理 *1* ([16])。设 T 是 $\mathbb{R}^d$中的一组 n 多边形曲线。如果存在一个用于 T 在 d_F 或 d_F 下的 (κ, δ)-近似最近邻问题的数据结构˜，其空间复杂度为 S、查询时间为 Q、删除时间为 D，预处理时间为 P，那么就存在一个用于 T在 d_F或 $\tilde{d}_F$下的 $\kappa(1+O(\varepsilon))$-近似最近邻数据结构，其空间复杂度为 $O(\frac{1}{\varepsilon} S \log^2 n)$、查询时间为 $O(Q \log n)$，期望预处理时间为 $O\big(\varepsilon^{-1} \log^2 n P + (Q+D) n \log n\big)$。

根据定理 1，我们可以专注于（$1+\varepsilon,\ \delta$）-近似最近邻问题。不失一般性，我们假设 T 中的每条曲线恰好有 m 个顶点，并且每条查询曲线恰好有 k 个顶点。如有必要，可以以任意方式添加额外顶点来满足这一假设，而不影响弗雷歇距离。

我们预处理的总体思路是，识别与每个 $\tau_i \in T$ 的弗雷歇距离为 $(1+O(\varepsilon))\delta$ 的所有查询曲线，将具有相似结构特征的曲线分组，为每组分配一个唯一的键值，并将这些键值存储在字典树 $\mathcal{D}$ 中。一个查询曲线可能属于多个分组。每个键值都与 T 中生成该键值的曲线子集相关联。相应地，给定查询曲线 σ，我们为 σ 生成所有可能的键值，并使用这些键值搜索$\mathcal{D}$ 。如果检索到 T 中的某条曲线，它就是所需的答案；否则，我们报告“否”。

需要克服两个挑战。首先，不可能检查所有可能的查询曲线。为了保证运行时间有限，我们只能检查某种空间离散化。为了控制离散化误差，可以用具有适当单元宽度的网格覆盖输入曲线；然而，网格大小以及数据结构大小将因此取决于某些非组合参数。我们提出了查询曲线的粗编码，使其数量为 $O(\sqrt{d}/\varepsilon)^{4d(k-1)}(mn)^{4(k-1)}$ 。一个查询曲线可能有$O(\sqrt{d}/\varepsilon)^{2d(k-2)}$ 种粗编码。第二个挑战是在查询时间高效生成查询曲线的所有可能粗编码。我们将粗编码生成归约为近似线段射击问题。当我们力求避免空间复杂度中出现任何形如 $m^{\Omega(d)}$ 或 $n^{\Omega(d)}$ 的因子时，这一步在四维及更高维中成为瓶颈。这也是查询时间中出现$(mn)^{0.5+\varepsilon}$ 项的原因。在二维和三维中，可以更高效地解决近似线段射击问题。

在本节的其余部分，我们将介绍粗编码以及一个 $(1+O(\varepsilon)\ \delta)$-近似最近邻数据结构，该结构使用近似线段射击预言机。根据 [8] 中的结果，可以在二维和三维中解决近似线段射击问题。我们将在第 4 节中解决四维及更高维度中的近似线段射击问题。

2.1 查询曲线的粗编码

设想在 $\mathbb{R}^d$ 中有一个单元宽度为 $\varepsilon\delta/\sqrt{d}$ 的无限网格。对于任意子集 $R \subset \mathbb{R}^d$，我们用 $G(R)$ 表示与 R 相交的网格单元集合。令 $\mathcal{G}_1 = \bigcup_{i\in[n],\ a\in[m]} G(v_{i,\ a} \oplus B_\delta)$。令$\mathcal{G}_2 = \bigcup_{i\in[n],\ a\in[m]} G(v_{i,\ a} \oplus B_{(2+12\varepsilon)\delta})$。$\mathcal{G}_1$ 和 $\mathcal{G}_2$ 均具有 $O(mn/\varepsilon^d)$ 大小。

曲线 $\sigma = (w_1, w_2, \ldots, w_k)$ 的粗编码是一个三元组 F = (A, B, C)。分量 C 是由网格单元点对组成的序列 $((c_j,1,c_j,2))_{j\in[k-1]}$，满足 $(c_j,1,c_j,2) \in (G_1 \times G_1) \cup \{\text{null}\}$。A 和 B 都是长度为 $k-1$ 的数组，且 A 和 B 中的每个元素都属于 $G_2 \cup \{\text{null}\}$。在描述实现这一直觉的约束之前，我们先说明 (A, B, C) 设计背后的直觉。

设曲线 $\tau_i \in T$ 是 σ 的一个 $(1 + O(\varepsilon))$-近似最近邻。数据结构需要支持预处理，此时查询曲线 σ 尚不可用；还需要支持查询过程，在此过程中，为避免对 mn 的线性依赖，我们不希望直接查询 T 中的输入曲线。

在预处理中，我们使用网格单元对作为可能的查询曲线边的替代对象。这样做的优势在于，我们可以枚举所有可能的网格单元对，从而涵盖所有可能的查询曲线边。具体而言，对于 $j \in [k-1]$，如果 $(c_j,1,c_j,2) \neq \text{null}$，则它是 w_jw_{j+1} 的替代对象，因此 w_jw_{j+1} 应经过靠近 c_j,1 和 c_j,2 的位置。每个非空的 $(c_j,1,c_j,2)$ 都对应于 τ_i 的一个连续顶点子序列 $v_{i,a}, \ldots, v_{i,b}$，这些顶点在弗雷歇匹配中与 w_jw_{j+1} 中的点相匹配。当然，我们并不知道弗雷歇匹配，因此需要枚举并处理所有可能的情况。此外，由于在预处理中 w_jw_{j+1} 是未知的，$v_{i,a}, \ldots, v_{i,b}$ 只能与一条线段相匹配，该线段连接 c_j,1 中的一个顶点 x_j 和 c_j,2 中的一个顶点 y_j，使得对于某个子线段 $x'_jy'_j \subseteq x_jy_j$，$d_F(x'_jy'_j, \tau_i[v_{i,a}, v_{i,b}]) \leq (1 + O(\varepsilon))\delta$。这一性质将在后续的数据结构构造中得到保证。

在查询时间，给定 $\sigma = (w_1, \ldots, w_k)$，我们将进行近似线段射击查询，以确定一个单元对序列 $((c_j,1,c_j,2))_{j\in[k-1]}$。不过，我们并不总是使用 $(c_j,1,c_j,2)$ 作为边 w_jw_{j+1} 的替代对象。如上一段所述，一个非空的 $(c_j,1,c_j,2)$ 表示输入曲线顶点的连续子序列与 w_jw_{j+1} 的匹配；然而，我们还必须允许将 σ 的顶点连续子序列与一条输入边匹配。因此，在确定 $((c_j,1,c_j,2))_{j\in[k-1]}$ 后，我们仍可选择直接使用 $(c_j,1,c_j,2)$，或将其替换为 null 值。出于技术原因，$(c_1,1,c_1,2)$ 和 $(c_{k-1},1,c_{k-1},2)$ 始终保持非空，因此我们有 2^{k-3} 种可能的单元对序列。取其中一个序列。若 $(c_r,1,c_r,2)$ 和 $(c_s,1,c_s,2)$ 是两个非空点对，且对于 $j \in [r+1, s-1]$，$(c_j,1,c_j,2)$ 为 null，这意味着 τ_i 的任何顶点都没有与 w_jw_{j+1} 匹配。因此，σ 的顶点 $w_{r+1}, \ldots, w_s$ 与 τ_i 的边 $v_{i,b}v_{i,b+1}$ 匹配，其中 b 是在当前枚举中与 w_rw_{r+1} 匹配的 τ_i 的最后一个顶点。我们使用单元对 $A[r]$ 和 $B[s]$ 作为边 $v_{i,b}v_{i,b+1}$ 的替代对象。因此，我们要求 $A[r]$ 和 $B[s]$ 分别靠近 c_r,2 和 c_s,1，因为 $(c_r,1,c_r,2)$ 是 w_rw_{r+1} 的替代对象，而 $(c_s,1,c_s,2)$ 是 w_sw_{s+1} 的替代对象。我们必须尝试 $A[r]$ 和 $B[s]$ 在 c_r,2 和 c_s,1 附近的所有可能位置。$A[r]$ 和 $B[s]$ 可以作为 T 中多条曲线的边的替代对象，这使我们能够在查询时间同时将 σ 与多条输入曲线进行比较。需要满足的约束是，$w_{r+1}, \ldots, w_s$ 可以与连接 $A[r]$ 的一个顶点 x_r 和 $B[s]$ 的一个顶点 x_s 的线段匹配，使得对于某个子线段 $x'_rx'_s \subseteq x_rx_s$，$d_F(x'_rx'_s, \sigma[w_{r+1}, w_s]) \leq (1 + O(\varepsilon))\delta$。图1展示了上述直觉的示意图。

我们给出实现上述直觉的 $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ 的约束。当 $(c_{j,\ 1},\ c_{j,\ 2}) \neq$null 时，$c_{j,\ 1}$ 的一种自然选择是：从 w_j 走向w_{j+1}时遇到的第一个网格单元，也就是线段射击。在 $\mathbb{R}^d$ 中，当 $d \in \{2, 3\}$ 时，存在针对盒子 [8] 的射线投射数据结构。在高维中，已知针对单个凸

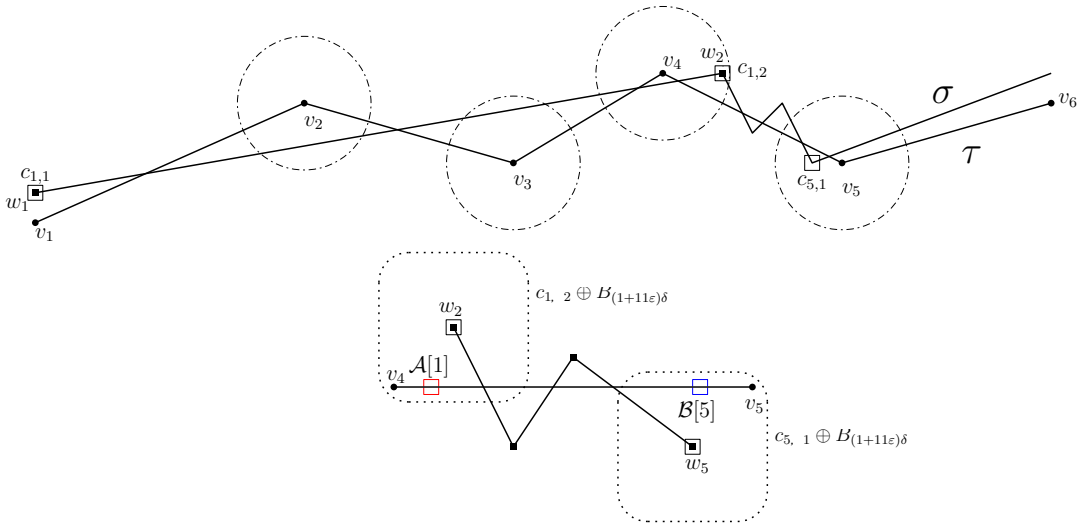

图1 推导 σ粗编码的基本直觉。为便于记号表示，我们使用 τ 代替τ_i。假设 $d_F(\tau,\ \sigma) \le \delta$，且顶点 v_1、v_2、v_3、v_4通过弗雷歇匹配与线段 w_1w_2匹配。 w_1w_2必须与以 τ的顶点为中心的一些球相交，这意味着 w_1w_2 与G_1相交。令$c_1,\ 1$和$c_1,\ 2$分别为G_1 中沿 w_1w_2方向与 w_1w_2 相交的第一个和最后一个单元。（$c_1,\ 1,\ c_1,\ 2$）可以作为边 w_1w_2 的替代对象，因为我们可以通过检查 v_1、v_2、v_3、v_4是否能够与一条连接$c_1,\ 1$和$c_1,\ 2$的顶点的线段正确匹配，来验证它们是否能够与 w_1w_2正确匹配。对于 σ中所有具有与其匹配的 τ顶点的边，这一思想都可以推广。子曲线 $\sigma[w_2, w_5]$与 τ的一条边 v_4v_5匹配。我们引入$\mathsf{A}[1] \subset G(c_1,\ _2 \oplus B_{(1+11\varepsilon)\delta})$和$\mathsf{B}[5] \subset G(c_5,\ _1 \oplus B_{(1+11\varepsilon)\delta})$。（$\mathsf{A}[1], \mathsf{B}[5]$）可以作为 v_4v_5的替代对象。（$\mathsf{A}[1], \mathsf{B}[5]$）能够充分编码 $\sigma[w_2, w_5]$，因为对于连接$\mathsf{A}[1]$ 的一个顶点 x_1与$\mathsf{B}[5],$的一个顶点 x_5 的每条线段 x_1x_5，都存在一个子线段$x_1'x_5' \subset x_1x_5$，使得 $d_F(x_1'x_5',\ \sigma[w_2, w_5]) \le (1 + O(\varepsilon))\delta$。

多面体和超平面排列 [1] 的射线发射结果；即使在这类情况下，只有当空间复杂度至少为输入规模的 $\Omega(d)$ 次幂时，查询时间才会显著地低于线性。对于我们的情况，这将是 $(mn)^{\Omega(d)}$。下面我们定义一个 λ-线段查询问题，以近似射线发射问题，并将在第 4 节中给出 $\lambda = 11\varepsilon\delta$ 的高效解法，避免空间复杂度中出现 $(mn)^{\Omega(d)}$ 项。如前所述， [8] 中的射线发射结果足以适用于二维和三维。

λ-线段查询。对象集合 O 在 $\mathbb{R}^d$ 中被预处理为一个数据结构，使得对于任意有向查询线段 pq， λ-线段查询 pq 在 O 上都会返回以下答案之一：

- 如果 pq 与 O中的某个对象相交，那么当我们从 p 沿着 q行进时，令 x 为与 O 中某个对象的第一个交点。在这种情况下，查询会返回一个对象 $o \in O$，使得 px 与 $o \oplus B_\lambda$ 相交。图 2 展示了示意图。否则，查询会返回 null，或返回一个对象
- $o \in O$，使得 $d(o,\ pq) \le \lambda$。

现在我们可以陈述关于 $(\mathcal{A},\ \mathcal{B},\ \mathcal{C})$ 的三个约束。

- **约束1：**

(a) $(c_1,\ _1,\ c_1,\ _2)$ 和 $(c_{k-1},\ _1,\ c_{k-1},\ _2)$ 都属于 $\mathcal{G}_1 \times \mathcal{G}_1$。(b) 对于 $j \in [k-1]$，如果 $(c_j,\ _1,\ c_j,\ _2) \neq$ null，则：(i) $c_j,\ _1$ 和 $c_j,\ _2$ 分别是带有 w_jw_{j+1}和 $w_{j+1}w_j$ 、在 $\mathcal{G}_1,$上执行的 ($11\varepsilon\delta$)-线段查询所返回的网格单元；

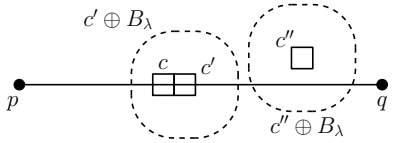

图 2 以 pq 为盒子的 λ-线段查询 $\{c, c', c''\}$ 可以返回 c 或 c'，但不能返回 c''。

(ii) $w_j w_{j+1} \cap (c_{j,1} \oplus B_{11\varepsilon\delta})$ 中的最小点在 $w_j w_{j+1} \cap (c_{j,2} \oplus B_{11\varepsilon\delta})$ 中的最大点前面（相对于 $\leq_{w_j w_{j+1}}$）。

约束 2:
(a) $\mathcal{B}[1]$ 和 $\mathcal{A}[k-1]$ 属于 $\mathcal{G}_2$。 (b) $w_1 \in \mathcal{B}[1]$ 和 $w_k \in \mathcal{A}[k-1]$。

约束 3:
(a) 对于 $j \in [2, k-2]$，如果 $(c_j,1,c_j,2) = \text{null}$，那么 A$[j]$ 和 B$[j]$ 均为 null。 (b) 对于 $j \in [k-1]$，如果 $(c_j,1, c_j,2) \neq \text{null}$，那么 A$[j]$ 和 B$[j]$ 属于 G2, $d(c_j,1, \text{B}[j]) \leq (1+11\varepsilon)\delta$，且 $d(c_j,2, \text{A}[j]) \leq (1+11\varepsilon)\delta$。 (c) 令 J 为满足 $r < s$、$(c_r,1, c_r,2) \neq \text{null}$、$(c_s,1,c_s,2) \neq \text{null}$ 且 $(c_j,1,c_j,2) = \text{null}$ 的 $(r, s) \in [k-1] \times [k-1]$ 的集合，其中 $j \in [r+1, s-1]$。对于每个 $(r, s) \in$ J，令 x_r 和 x_s 分别为 A$[r]$ 和 B$[s]$ 中按照其坐标的字典序最小的顶点，则存在 $x'_r, x'_s \in x_r x_s$，使得 $x'_r \leq_{x_r x_s} x'_s$ 且 $d_F(x'_r x'_s, \sigma[w_{r+1}, w_s]) \leq (1+\varepsilon)\delta$。

我们注意到，如果 $w_j w_{j+1}$ 与单元并集的内部相交，那么根据约束 1(b)(i)，对于 $(c_{j,1}, c_{j,2})$，$_1$，满足约束 1(b)(ii)。当 $w_j w_{j+1}$ 不与$\mathcal{G}_1$，中单元并集的内部相交时，$(11\varepsilon\delta)$-线段查询可能会返回两个违反约束 1(b)(ii) 的单元。在这种情况下，输入顶点距离 $w_j w_{j+1}$ 太远，无法与 $w_j w_{j+1}$ 上距离不超过 δ的任何点匹配，因此我们可以将 $(c_{j,1}, c_{j,2})$ 设为 null。

接下来的结果表明，任何距离曲线 $\tau_i \in T$ 较近的查询曲线 σ 都有一种具有某些附加性质的粗编码。这些性质将在分析中发挥作用。令$\mathcal{M}$表示 σ 与某个 $\tau_i \in T$之间的一种匹配。对于任意子曲线 $\sigma' \subseteq \sigma$，我们用$\mathcal{M}(\sigma')$ 表示与 σ' 通过$\mathcal{M}$匹配的 τ_i 的子曲线。

▶ 引理 2. 设 $\sigma = (w_1, \ldots, w_k)$ 是一条包含 k 个顶点的曲线。设 M 是 σ 与 $\tau_i \in T$ 之间的匹配，使得 $d_{\mathcal{M}}(\tau_i, \sigma) \leq \delta$。设 $\tilde{\pi}_j = \{v_{i,a} : a \in [m-1], v_{i,a} \in M(w_j w_{j+1}) \setminus M(w_j)\}$，对所有 $j \in [k-1]$ 均成立。对所有 $j \in [k-2]$，$\pi_{k-1} = \{v_{i,m}\} \cup \tilde{\pi}_{k-1}$，定义 $\pi_j = \tilde{\pi}_j$；并令 $\pi_0 = \{v_{i,1}, \ldots, v_{i,m}\} \setminus \bigcup_{j=1}^{k-1} \pi_j$。存在一个针对 σ 的粗编码 (A, B, C)，满足以下性质。

(i) 对于 $j \in [2, k-1]$，$\pi_j = \emptyset$，当且仅当 $(c_j,1, c_j,2) =$ 为 *null*。*(ii)* 对所有 $(r, s) \in J$，若 $r = 1$ 且 $\pi_1 = \emptyset$，则令 $b_1 = 1$；否则，令 $b_r = \max\{b : v_{i,b} \in \pi_r\}$。对于所有 $(r, s) \in J$，存在一个点 $z \in A[r] \cap \tau_{i,b_r}$，以及另一个点 $z' \in B[s] \cap \tau_{i,b_r}$，使得 $z \leq_{\tau_{i,b_r}} z'$。

证明。我们如下定义分量 C。已知 $v_{i,1} \in$M(w_1)，$w_1 w_2$ 与 G_1，中单元并集的内部相交，因此使用 $w_1 w_2$ 对 G_1 进行 $(11\varepsilon\delta)$-线段查询必定会返回某个单元；我们将其定义为 c_1,1。类似地，使用 $w_2 w_1$ 对 G_1 进行 $(11\varepsilon\delta)$-线段查询必定会返回某个单元；我们将其定义为 c_1,2。点对 $(c_{k-1}$,1, c_{k-1},2) 也按照类似于 $v_{i,m} \in$M(w_k) 的方式定义。考虑任意一个 $j \in [2, k-1]$。若对于某个 $a \in [m]$，有 $v_{i,a} \in \pi_j$，则$v_{i,a} \in$ M$(w_j w_{j+1})$，这意味着 $w_j w_{j+1}$ 与 $v_{i,a} \oplus B_\delta$ 相交，因此也与 G_1 中单元并集的内部相交。因此，可以通过 $(11\varepsilon\delta)$-线段查询来定义 $(c_j$,1,c_j,2)

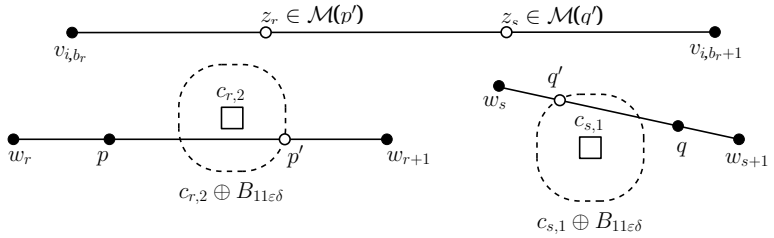

图 3 p、p'、q'、q、z_r 和 z_s 的示意图。

与之前一样，使用 $w_j w_{j+1}$ 和 $w_{j+1} w_j$。另一方面，若 $\pi_j = \emptyset$，则将 $(c_{j,1}, c_{j,2})$ 定义为 null。因此，约束 1 和引理中的性质（i）均得到满足。

接下来，我们定义$\mathcal{A}$ 和$\mathcal{B}$，使其满足约束 2 和约束 3。

由于 $v_{i,1} \in \mathsf{M}(w_1)$ 且 $v_{i,m} \in \mathsf{M}(w_k)$，$d(w_1, v_{i,1}) \le \delta$ 和 $d(w_k, v_{i,m}) \le \delta$ 均至多为 δ。因此，w_1 位于 $G(v_{i,1} \oplus B_\delta) \subset G(v_{i,1} \oplus B_{(2+12\varepsilon)\delta}) \subset \mathsf{G}_2$ 中的一个单元内；我们将此单元记为 $\mathsf{B}[1]$。类似地，我们将 G_2 中包含 w_k 的单元定义为 $\mathsf{A}[k-1]$。由此满足约束 2。

对于 $j \in [2, k-2]$，若 $(c_j, 1, c_j, 2) = \text{null}$，则令 $\mathsf{A}[j]$ 和 $\mathsf{B}[j]$ 为 null，从而满足约束 3(a)。$\mathsf{B}[1]$ 和 $\mathsf{A}[k-1]$ 已经定义，且它们属于 G_2。由于 w_1 位于 $G(v_{i,1} \oplus B_\delta) \subset \mathsf{G}_1$，中的一个单元内，根据 $(11\varepsilon\delta)$-线段查询可得 $w_1 \in c_{1,1} \oplus B_{11\varepsilon\delta}$。于是，$d(c_1, 1, \mathsf{B}[1]) \le 11\varepsilon\delta$，因为 $w_1 \in \mathsf{B}[1]$。类似地，$d(c_{k-1}, 2, \mathsf{A}[k-1]) \le 11\varepsilon\delta$。因此，$\mathsf{B}[1]$ 和 $\mathsf{A}[k-1]$ 满足约束 3(b)。接下来还需讨论 $j \in [1, k-2]$ 对应的 $\mathsf{A}[j]$ 以及 $j \in [2, k-1]$ 对应的 $\mathsf{B}[j]$。

考虑任意一个 $j_* \in [k-1]$，使得 $(c_{j_*}, 1, c_{j_*}, 2) \ne \text{null}$。回想一下，J 是满足以下条件的 $(r, s) \in [k-1] \times [k-1]$ 的集合：$r < s$，$(c_r, 1, c_r, 2) \ne \text{null}$，$(c_s, 1, c_s, 2) \ne \text{null}$，且对于 $j \in [r+1, s-1]$，$(c_j, 1, c_j, 2) = \text{null}$。因此，如果 $j_* \le k-2$，它必定恰好作为 J 中某个元素的第一个值出现；如果 $j_* \ge 2$，它也必定恰好作为 J 中另一个元素的第二个值出现。因此，只需为每个 $(r, s) \in \mathsf{J}$ 定义 $\mathsf{A}[r]$ 和 $\mathsf{B}[s]$，并验证约束 3(b) 和 3(c) 得到满足即可。

任取一个（r, s）$\in \mathsf{J}$。若 $\pi_r \ne \emptyset$，则可以定义 $b_r = \max\{b : v_{i,b} \in \pi_r\}$。若 $\pi_r = \emptyset$，则 $r = 1$，因为对于任意 $r > 1$，$\pi_r \ne$，根据（i），由于（$c_r, 1, c_r, 2$）$\ne$，由J的定义可知其为null。在 $r = 1$ 且 $\pi_1 = \emptyset$的情况下，将 b_1 定义为1。因此，对于所有（r, s）$\in \mathsf{J}$，b_r 都有良好定义。根据 b_r 的定义可知，$b_r = \max\{b : v_{i,b} \in \mathsf{M}(w_r w_{r+1})\}$。由于（$c_s, 1, c_s, 2$）$\ne$为null，且对于 $j \in [r+1, s-1]$，（$c_j, 1, c_j, 2$）$=$也为null，根据（i），$\pi_s \ne \emptyset$ 且 $\pi_j = \emptyset$，对于$j \in [r+1, s-1]$。由此可得$v_{i,b_r+1} \in \pi_s$，它是$\mathsf{M}(w_s w_{s+1})$的子集。任取一点 $p \in w_r w_{r+1}$ 和一点 $q \in w_s w_{s+1}$，使得$v_{i,b_r} \in \mathsf{M}(p)$且$v_{i,b_r+1} \in \mathsf{M}(q)$。

我们声称 $p w_{r+1} \cap (c_r, 2 \oplus B_{11\varepsilon\delta})$ 和 $w_s q \cap (c_s, 1 \oplus B_{11\varepsilon\delta})$ 均非空。由于 $c_r, 2$ 是通过以 $w_{r+1} w_r$ 为参数的 $(11\varepsilon\delta)$-线段查询返回的 G_1 中的单元，对于 $w_{r+1} w_r$ 与 G_1，中任意单元之间的任意交点 x，根据定义，我们有 $x w_{r+1} \cap (c_r, 2 \oplus B_{11\varepsilon\delta}) \ne \emptyset$。根据我们对 p 的选择，我们有 $p \in w_r w_{r+1} \cap (v_{i,b_r} \oplus B_\delta)$；这意味着 p 是 $w_r w_{r+1}$ 与 $G(v_{i,b_r} \oplus B_\delta) \subset \mathsf{G}_1$ 中某个单元之间的交点。因此，我们可以用 p 替换 x，并得出 $p w_{r+1} \cap (c_r, 2 \oplus B_{11\varepsilon\delta}) \ne \emptyset$。类似地，我们得到 $w_s q \cap (c_s, 1 \oplus B_{11\varepsilon\delta}) \ne \emptyset$。

根据我们的断言，当我们从 w_{r+1} 走到 p时，会在某个点 p'处经过 $c_r, 2 \oplus B_{11\varepsilon\delta}$；当我们从 w_s 走到 q时，会在某个点 q'处经过 $c_s, 1 \oplus B_{11\varepsilon\delta}$。取两个点 $z_r \in \mathsf{M}(p')$ 和 $z_s \in \mathsf{M}(q')$。根据定义，对于某个 $\tau_{i_r} \in T$ 和某个索引$a_r \in [m]$，有 $c_r, 2 \in G(v_{i_r, a_r} \oplus B_\delta)$。$c_r, 2$ 的单元宽度为 $\varepsilon\delta/\sqrt{d}$，因此 $c_r, 2 \subset v_{i_r, a_r} \oplus B_{(1+\varepsilon)\delta}$。根据三角不等式，$p' \in v_{i_r, a_r} \oplus B_{(1+12\varepsilon)\delta}$，因此 $z_r \in v_{i_r, a_r} \oplus B_{(2+12\varepsilon)\delta}$，这意味着 z_r 包含在 $G(v_{i_r, a_r} \oplus B_{(2+12\varepsilon)\delta}) \subset \mathsf{G}_2$ 中的一个单元内。类似地，我们也可以推得 z_s 包含在 G_2 中的一个单元内。我们将 G_2 中分别包含 z_r 和 z_s 的单元定义为 $\mathsf{A}[r]$ 和 $\mathsf{B}[s]$。图 3 展示了示意图。

由于 $d(c_{r,2}, z_r) \le d(c_{r,2}, p') + d(p', z_r) \le (1+11\varepsilon)\delta$，我们得到 $d(c_{r,2}, \mathcal{A}[r]) \le (1+11\varepsilon)\delta$。类似地，$d(c_{s,1}, \mathcal{B}[s]) \le (1+11\varepsilon)\delta$。这就满足了约束 3(b)。

由于$v_{i,b_r} \in \mathrm{M}(p)$和 $p \le_{w_r w_{r+1}} p'$，我们有$v_{i,b_r} \le_{\tau_i} \mathrm{M}(p')$。类似地，我们有 $\mathrm{M}(q') \le_{\tau_i} v_{i,b_{r+1}}$。因此，$v_{i,b_r} \le_{\tau_i} \mathrm{M}(p') \le_{\tau_i} \mathrm{M}(q') \le_{\tau_i} v_{i,b_{r+1}}$。由于 $z_r \in \mathrm{M}(p')$且 $z_s \in \mathrm{M}(q')$，z_r和 z_s 满足引理的性质 (ii)。z_r 与 $\mathrm{A}[r]$的任意顶点 x_r 之间的距离至多为 $\varepsilon\delta$。z_s 与 $\mathrm{B}[s]$的任意顶点 x_s 之间的距离也如此。因此，我们可以使用 $x_r x_s$ 和 $z_r z_s$ 之间的线性插值 I 作为匹配，从而得到 $d_{\mathcal{I}}(x_r x_s, z_r z_s) \le \varepsilon\delta$。结合 M 和 I 可知，$\sigma[p', q']$ 和 $x_r x_s$ 之间存在距离不超过 $(1+\varepsilon)\delta$ 的匹配。由于 $\sigma[w_{r+1}, w_s] \subseteq \sigma[p', q']$，我们由此验证了约束 3(c)。◀

2.2 数据结构的组织与构造

我们在 $O(mn/\varepsilon^d)$ 时间和空间内构造 $\mathcal{G}_1$ 和 $\mathcal{G}_2$。我们需要一个用于 $\mathcal{G}_2$ 的点定位数据结构，其组织方式如下：顶层树的叶节点对应于单元在第一坐标轴上的区间。每个叶节点关联着投影到该叶节点区间上的单元，而这些单元存储在第二层树中，其叶节点对应于这些单元在第二坐标轴上的区间。以此类推，可得到 $d = O(1)$ 层，使用 $O(|\mathcal{G}_2|) = O(mn/\varepsilon^d)$ 空间和 $O\big((mn/\varepsilon^d)\log\frac{mn}{\varepsilon}\big)$ 预处理时间。一次点定位需要 $O(\log\frac{mn}{\varepsilon})$ 时间。

$(1+O(\varepsilon))\delta$-近似最近邻数据结构是一个字典树$\mathcal{D}$。存储在$\mathcal{D}$ 中的每个键都是一个候选粗编码，它是一个 3 元组 $(\mathcal{A},\mathcal{B},\mathcal{C})$，与粗编码相同。对于候选粗编码，必须满足约束 1(a)、2(a)、3(a) 和 3(b)，但忽略约束 1(b)、2(b) 和 3(c)。之所以需要这种区别，是因为约束 1(b)、2(b) 和 3(c) 要求使用查询曲线，而预处理时无法获得查询曲线。对于每个候选粗编码 E，令 T_E 为输入曲线的子集，其中的曲线与任意具有粗编码 E 的查询曲线之间的弗雷歇距离不超过 $(1+O(\varepsilon))\delta$。我们稍后将讨论如何获取 T_E中的曲线。

$\mathcal{D}$ 中的每个键 E 的大小为 $O(k)$，因为 E 在 $\mathcal{G}_1$ 和 $\mathcal{G}_2$ 中存储了 $O(k)$ 个单元。作为一棵字典树，$\mathcal{D}$ 是一棵有根树，其层数与键 E 的长度相同。在 D 中进行搜索，归结为访问 $\mathcal{D}$ 中某个内部节点的相应子节点。键 E 的每个分量都是 $\mathcal{G}_2 \cup \{\text{null}\}$ 或 $(\mathcal{G}_1 \times \mathcal{G}_1) \cup \{\text{null}\}$ 中的一个元素；共有 $O(m^2n^2/\varepsilon^{2d})$ 种可能。我们在 $\mathcal{D}$ 的每个内部节点处维护一个字典，以便在 $O(\log\frac{mn}{\varepsilon})$ 时间内找到要访问的相应子节点。因此，$\mathcal{D}$ 的总搜索时间为 $O(k\log\frac{mn}{\varepsilon})$。

要界定 $\mathcal{D}$ 的大小，请注意，$\mathcal{D}$ 的叶节点处的每个键 E 都会在该叶节点祖先的字典中产生 $O(k)$ 个条目。共有 $O(\sqrt{d}/\varepsilon)^{4d(k-1)}(mn)^{4(k-1)}$ 个候选粗编码。因此，内部节点处字典所占用的总空间为 $O(\sqrt{d}/\varepsilon)^{4d(k-1)}(mn)^{4(k-1)}k$。我们将证明，如果查询曲线的粗编码为 E，那么 T_E 中的任意曲线与该查询曲线之间的弗雷歇距离至多为 $(1+O(\varepsilon))\delta$。因此，我们只需在对应于 E 的叶节点处存储 T_E 中的一条曲线，并且只存储该曲线的索引即可。因此，$\mathcal{D}$ 的总空间复杂度为 $O(\sqrt{d}/\varepsilon)^{4d(k-1)}(mn)^{4(k-1)}k$。

$\mathcal{D}$ 的构造过程如下。我们将$\mathcal{D}$ 初始化为空。根据约束 1(a)、2(a)、3(a) 和 3(b)，枚举所有可能的候选粗编码。取一个可能的候选粗编码 E。将集合 T_E 初始化为空。我们遍历每条输入曲线 τ_i，以确定是否将 τ_i 纳入 T_E。如果最终满足 $T_E \neq \emptyset$，则将 E 连同 T_E 中的一条曲线插入$\mathcal{D}$。接下来，我们讨论如何检查是否将 τ_i 纳入 T_E。

设 E 为 $(\mathcal{A},\mathcal{B},\mathcal{C})$。我们生成满足以下性质的 划分 $\{v_{i,1}, \ldots, v_{i,m}\}$ 的所有可能形式。

23:10 弗雷歇距离下多边形曲线的近似最近邻

划分： 一个由 k 个互不相交的子集组成的序列（π_0、π_1、……、π_{k-1}），满足 $\bigcup_{j=0}^{k-1}\pi_j=\{v_{i,1}$、……、$v_{i,m}\}$，$v_{i,1}\in\pi_0$、$v_{i,m}\in\pi_{k-1}$、$\pi_j$ 对某些 $j\in[k-2]$ 可以为空；并且对于任意 $v_{i,a}\in\pi_j$ 以及任意 $v_{i,b}\in\pi_l$，若 $j<l$，则 $a<b$。

少于 m^{k-1} 个划分。给定一个划分 $(\pi_0,\ldots\ldots,\pi_{k-1})$，$\pi_0$中的顶点将与 $v_{i,1}$匹配；对于 $j\in[k-1]$，π_j 中的顶点将与 $w_jw_{j+1}\setminus\{w_j\}$中的点匹配，其中 w_jw_{j+1} 是查询曲线的第 j 条边；$v_{i,m}$ 以及 τ_i 中可能的其他顶点将与 w_k匹配。提及 w_jw_{j+1} 只是概念上的；我们在预处理中无需知道查询曲线。

下面我们描述针对每个划分的四项测试。一旦遇到通过全部四项测试的划分，我们就将 τ_i 插入 T_E。如果某个划分未通过任何一项测试，我们就转到下一个划分。如果最终没有划分能通过全部四项测试，我们就不将 τ_i 纳入 T_E。

第一项测试是：对于 $j\in[2,k-1]$，当且仅当 $(c_{j,1},c_{j,2})=$ 为 null 时，条件成立。该测试耗时 $O(k)$。我们不将 π_1 纳入此测试，因为根据约束1(a)，$(c_{1,1},c_{1,2})\neq$ 为 null，而 π_1 是否为空取决于所枚举的划分。

在第二项测试中，对于 $j\in[k-1]$，若 $\pi_j\neq\emptyset$，令 a_j 和 $b_j\in[m]$ 分别为满足 v_{i,a_j}、$v_{i,b_j}\in\pi_j$ 的最小和最大索引。直觉上，v_{i,a_j}、……、v_{i,b_j} 可以在距离 $(1+O(\varepsilon))\delta$ 以内与 w_jw_{j+1} 的替代对象 $(c_j,1,c_j,2)$ 匹配。第二项测试按如下方式检查这一性质。注意，$(c_j,1,c_j,2)$ 不为 null：根据约束1(a)，$(c_1,1,c_1,2)$ 不为 null；对于 $j\in[2,k-1]$，根据第一项测试，$(c_j,1,c_j,2)$ 不为 null。按照其坐标的字典序，选取 $c_j,1$ 中最小的顶点 x_j，以及 $c_j,2$ 中最小的顶点 y_j。若 $x_jy_j\cap(v_{i,a_j}\oplus B_{(1+12\varepsilon)\delta})$ 或 $x_jy_j\cap(v_{i,b_j}\oplus B_{(1+12\varepsilon)\delta})$ 为空，则测试失败。否则，计算 $x_jy_j\cap(v_{i,a_j}\oplus B_{(1+12\varepsilon)\delta})$ 中的最小点 x'_j，以及 $x_jy_j\cap(v_{i,b_j}\oplus B_{(1+12\varepsilon)\delta})$ 中相对于 $\leq_{x_jy_j}$ 的最大点 y'_j。若 $x'_j\leq_{x_jy_j}y'_j$ 不成立，则测试失败。假设 $x'_j\leq_{x_jy_j}y'_j$。计算 $d_F(x'_jy'_j,\tau_i[v_{i,a_j},v_{i,b_j}])$，并检查其是否不大于 $(1+12\varepsilon)\delta$。若对所有 $j\in[k-1]$，上述检查均成功，则第二项测试成功；否则测试失败。该测试耗时 $O(m\log m)$，其中主要耗时在于对所有 $j\in[k-1]$ 计算 $d_F(x'_jy'_j,\tau_i[v_{i,a_j},v_{i,b_j}])$。

第三项检验是 $\mathcal{B}[1]\in G(v_{i,1}\oplus B_\delta)$ 和 $\mathcal{A}[k-1]\in G(v_{i,m}\oplus B_\delta)$，这归结为检查 $d(v_{i,1},\mathcal{B}[1])$ 和 $d(v_{i,m},\mathcal{A}[k-1])$ 是否均不超过 δ。

第四项测试涉及集合 J，该集合由满足 $r<s$ 的 $(r,s)\in[k-1]\times[k-1]$ 构成，其中 $(c_r,1,c_r,2)\neq$为空，$(c_s,1,c_s,2)\neq$为空，且对于 $j\in[r+1,s-1]$，$(c_j,1,c_j,2)=$为空。注意，$|\mathbb{J}|\leq k-1$，并且可以在 $O(k)$ 时间内构造。对于每个 $(r,s)\in$J，如果 $r=1$ 且 $\pi_1=\emptyset$，则令 $b_1=1$；否则，令 $b_r=\max\{b:v_{i,b}\in\pi_r\}$。由此可得 $b_r+1=\min\{a:v_{i,a}\in\pi_s\}$。我们检查是否满足以下条件：$\tau_i,_{b_r}\cap\mathrm{A}[r]\neq\emptyset$，$\tau_i,_{b_r}\cap\mathrm{B}[s]\neq\emptyset$，并且当我们从 v_{i,b_r} 移动到 v_{i,b_r+1} 时，先于 $\mathrm{B}[s]$ 遇到 $\mathrm{A}[r]$。（回顾一下直觉：点对 $\mathrm{A}[r]$ 和 $\mathrm{B}[s]$ 充当边 $\tau_{i,b_r}=v_{i,b_r}v_{i,b_r+1}$ 的替代对象。）如果对于任意 $(r,s)\in$J，检查未通过，则测试失败。否则，测试成功。该测试的运行时间为 $O(k)$。

下面的结果概述了$\mathcal{D}$ 的构造，以及$\mathcal{D}$中每个候选粗编码的四个性质。

►引理 *3.* 字典树 *D* 的大小为 $O(\sqrt{d/\varepsilon})^{4d(k-1)}(mn)^{4(k-1)}k$，可在$O(\sqrt{d/\varepsilon})^{4d(k-1)}(mn)^{4(k-1)}(k\log\frac{mn}{\varepsilon}+m^k\log m)$ 时间内构造。我们可以在 $O(k\log\frac{mn}{\varepsilon})$ 时间内使用粗编码搜索 *D*。对于每个候选粗编码 $E=(A,B,C)$，曲线$\tau_i\in T$属于 T_E 当且仅当存在一个对 τ_i顶点的划分 $(\pi_0,\ldots,\pi_{k-1})$，满足以下四个性质。对于 $j\in[k-1]$，若 $j=1$ 且 $\pi_1=\emptyset$，则令 $b_1=1$；否则，若 $\pi_j\neq\emptyset$，则令 $a_j=\min\{a:v_{i,a}\in\pi_j\}$，并令 $b_j=\max\{b:v_{i,b}\in\pi_j\}$。

(i) </style}对于 ,, 当且仅当 $(c_{j,1},c_{j,2})=null$。(ii) 对于 $j\in[k-1]$，如果 $\pi_j\neq\emptyset$，令 x_j和 y_j 分别为 $c_{j,1}$ 和 $c_{j,2}$ 中按照其坐标的字典序排列最小的顶点，据此

存在 $x''_j, y''_j \in x_j y_j$，使得 $x''_j \leq_{x_j y_j} y''_j$ 且 $d_F(x''_j y''_j, \tau_i[v_{i,a_j}, v_{i,b_j}]) \leq (1+12\varepsilon)\delta$。*(iii)* $B[1] \in G(v_{i,1} \oplus B_\delta)$ 和 $A[k-1] \in G(v_{i,m} \oplus B_\delta)$。*(iv)* 对每个 $(r, s) \in J$，$\tau_{i,b_r} \cap A[r] \neq \emptyset$，$\tau_{i,b_r} \cap B[s] \neq \emptyset$，并且当我们从 v_{i,b_r} 走到 v_{i,b_r+1}时，经过 $A[r]$ 的时间不晚于经过 $B[s]$ 。

2.3 查询

在查询时，我们给定一条曲线 $\sigma = (w_1, \ldots\ldots, w_k)$。我们枚举 σ 的所有粗编码；对于每个枚举出的粗编码 E，我们在字典树 $\mathcal{D}$ 中搜索 E；如果找到 E，就返回存储在 T_E 中且与 E 一同保存的曲线，作为查询的答案；如果在 $\mathcal{D}$ 中找不到 σ 的任何粗编码，则返回“否”。

如引理 3 所述，在 $\mathcal{D}$ 中的每次搜索需要 $O(k \log \frac{mn}{\varepsilon})$ 时间。枚举 σ 的粗编码，需要按照第 2.1 节中的约束 1(b)(i)，在 $\mathcal{G}_1$ 上求解 $(11\varepsilon\delta)$ 段查询。我们稍后将讨论一种高效的解决方法。

对于 $j \in [k-1]$，我们在 G_1 上分别使用 $w_j w_{j+1}$ 和 $w_{j+1} w_j$ 进行两次 $(11\varepsilon\delta)$ 段查询，以获得 u_j,1 和 u_j,2。如果任一查询返回空值，则将 $(u_j,1,u_j,2)$ 定义为空值。如果 $(u_j,1,u_j,2) \neq$ 为空值，并且 $w_j w_{j+1} \cap (u_j,1 \oplus B_{11\varepsilon\delta})$ 中的最小点相对于 $\leq_{w_j w_{j+1}}$ 不在 $w_j w_{j+1} \cap (u_j,2 \oplus B_{11\varepsilon\delta})$ 中的最大点之前，则不满足约束 1(b)(ii)。此时必定是 $w_j w_{j+1}$ 未与 G_1，中单元并集的内部相交，而 $(11\varepsilon\delta)$ 段查询恰好返回了两个违反约束 1(b)(ii) 的单元。在这种情况下，输入顶点距离 $w_j w_{j+1}$ 太远，无法与 $w_j w_{j+1}$ 中距离不超过 δ 的任何点匹配，因此我们将 $(u_j,1,u_j,2)$ 重置为空值。

在定义了 $j \in [k-1]$ 的 $(u_j,1,u_j,2)$ 后，我们按如下方式生成 σ 的粗编码。点对 $(c_1,1,c_1,2)$ 和 $(c_{k-1},1,c_{k-1},2)$ 分别定义为 $(u_1,1,u_1,2)$ 和 $(u_{k-1},1,u_{k-1},2)$。对于 $j \in [2, k-2]$，我们通过令 $(c_j,1, c_j,2)$ 为 $(u_j,1,u_j,2)$ 或 null，枚举所有可能的 C。这样总共得到 2^{k-3} 个可能的 C。我们查询 G_2 的点定位数据结构，以找到分别包含 w_1 和 w_k 的单元 $\mathsf{B}[1]$ 和 $\mathsf{A}[k-1]$。然后，对于枚举出的每个 C，我们根据第 2.1 节中的约束 3(a) 和 3(b)，分别为 $j \in [1, k-2]$ 枚举 $\mathsf{A}[j]$，并为 $j \in [2, k-1]$ 枚举 $\mathsf{B}[j]$。此次枚举会产生 $O(\sqrt{d}/\varepsilon)^{2d(k-2)}$ 个 (A, B, C) 元组。对于枚举出的每个 (A, B, C)，我们检查其是否满足约束 3(c)；根据以下结果，这可以在 $O(k \log k)$ 时间内完成。

▶ **引理 4**。 任取 $(r、 s) \in J$。令 x_r 和 x_s 分别为 $A[r]$ 和 $B[s]$中按其坐标的字典序排列最小的顶点。我们可以在 $O((s-r)\log(s-r))$ 时间内检查是否存在 x'_r、$x'_s \in x_r x_s$，使得 $x'_r \leq_{x_r x_s} x'_s$ 以及 $d_F(x'_r x'_s$、$\sigma[w_{r+1}, w_s]) \leq (1+\varepsilon)\delta$。

▶ **引理 5**。 查询时间为 $O(kQ_{\text{seg}}) + O(\sqrt{d}/\varepsilon)^{2d(k-2)} k \log \frac{mn}{\varepsilon}$，其中 Q_{seg} 表示回答一次 $(11\varepsilon\delta)$段查询所需的时间。

2.4 近似保证

首先，我们证明：如果 σ 与某条输入曲线的弗雷歇距离在 δ 以内，则存在一个关于 σ 的粗编码 E，使得 $T_E \neq \emptyset$。因此， E 以及 T_E 中的一条曲线被存储在$\mathcal{D}$中。

▶ **引理 6.** 如果 $d_F(\tau_i, \sigma) \leq \delta$，那么 $\tau_i \in T_E$ 存在 σ的一种粗编码 E。

证明。 设$\mathcal{M}$ 是 τ_i 与 σ之间的弗雷歇匹配。设 E 为引理 2 中针对 σ 指定的粗编码。对于任意子曲线 $\sigma' \subseteq \sigma$，我们用$\mathcal{M}(\sigma')$表示 τ_i中与 σ' 通过$\mathcal{M}$匹配的子曲线。对于 $j \in [k-1]$，令 $\tilde{\pi}_j = \{v_{i,a}: a \in [m-1], v_{i,a} \in \mathcal{M}(w_j w_{j+1}) \backslash \mathcal{M}(w_j)\}$。

为 $j \in [k-2]$, $\pi_{k-1} = \{v_{i,m}\} \cup \tilde{\pi}_{k-1}$定义 $\pi_j = \tilde{\pi}_j$，以及 $\pi_0 = \{v_{i,1}, \ldots, v_{i,m}\} \setminus \bigcup_{j=1}^{k-1} \pi_j$。我们得到 τ_i顶点的一个划分$(\pi_0, \ldots, \pi_{k-1})$。

我们证明 E、τ_i 和 $(\pi_0, \ldots\ldots, \pi_{k-1})$ 满足引理 3(i)–(iv)，从而将 τ_i 置于 T_E中。引理 3(i) 直接由引理 2(i) 得出，

任取满足 $\pi_j \neq \emptyset$ 的 $j \in [k-1]$。令 π_j 为 $\{v_{i,a}, v_{i,a+1}, \ldots, v_{i,b}\}$。根据$\pi_j$的定义，$\pi_j$ 中的每个顶点都属于 $\mathsf{M}(w_j w_{j+1})$，因此 $\tau_i[v_{i,a}, v_{i,b}] \subset \mathsf{M}(w_j w_{j+1})$。于是，必定存在两个点 p、$q \in w_j w_{j+1}$，使得 $p \leq_{w_j w_{j+1}} q$ 且 $d_F(pq, \tau_i[v_{i,a}, v_{i,b}]) \leq \delta$。如果 $j = 1$，则根据约束1(a)，$(c_1, 1, c_1, 2) \neq$为 null；如果 $j \in [2, k-1]$，则根据引理 2(i)，$(c_j, 1, c_j, 2) \neq$为 null，因为$\pi_j \neq$为 null。因此，c_j,1 和 c_j,2 是 G_1 中的单元，分别由带有$w_j w_{j+1}$和 $w_{j+1} w_j$的 $(11\varepsilon\delta)$-线段查询返回。我们已经证明 $d_F(pq, \tau_i[v_{i,a}, v_{i,b}]) \leq \delta$；因此，$p$包含在 $G(v_{i,a} \oplus B_\delta) \subset \mathsf{G}_1$的一个单元中。由于 c_j,1 是带有 $w_j w_{j+1}$的 $(11\varepsilon\delta)$-线段查询所返回的单元，必定存在一个点 $z_p \in w_j w_{j+1} \cap (c_j, {}_1 \oplus B_{11\varepsilon\delta})$，使得 $z_p \leq_{w_j w_{j+1}} p$。类似地，我们可以得出结论，必定存在一个点 $z_q \in w_j w_{j+1} \cap (c_j, {}_2 \oplus B_{11\varepsilon\delta})$，使得 $q \leq_{w_j w_{j+1}} z_q$。也就是说，$z_p \leq_{w_j w_{j+1}} p \leq_{w_j w_{j+1}} q \leq_{w_j w_{j+1}} z_q$。令 x_j 和 y_j 分别为 c_j,1 和 c_j,2 中按照其坐标字典序排列最小的顶点。$d(z_p, x_j)$ 和 $d(z_q, y_j)$ 均不大于 $12\varepsilon\delta$。从 $z_p z_q$ 到 $x_j y_j$的线性插值将 p 和 q 分别映射到 $x_j y_j$上的两个点 x_j'' 和 y_j''，使得 $x_j'' \leq_{x_j y_j} y_j''$。此外，线性插值所增加的距离至多为 $12\varepsilon\delta$，由此得到 $d_F(x_j'' y_j'', \tau_i[v_{i,a}, v_{i,b}]) \leq d_F(x_j'' y_j'', pq) + d_F(pq, \tau_i[v_{i,a}, v_{i,b}]) \leq (1 + 12\varepsilon)\delta$。因此，引理 3(ii) 得证。

网格单元 $\mathcal{B}[1]$ 和 $\mathcal{A}[k-1]$ 分别定义为包含 w_1 和 w_k。此外，$v_{i,1} \in \mathcal{M}(w_1)$ 和 $v_{i,m} \in \mathcal{M}(w_k)$。因此，$\mathcal{B}[1] \in G(v_{i,1} \oplus B_\delta)$ 和 $\mathcal{A}[k-1] \in G(v_{i,m} \oplus B_\delta)$，满足引理 3(iii)。

对于任意点对 $(r, s) \in \mathcal{J}$，根据引理 2(ii)，存在两个点 $z \in \mathcal{A}[r] \cap \tau_{i,\, b_r}$和$z' \in \mathcal{B}[s] \cap \tau_{i,\, b_r}$ 使得 $z \leq_{\tau_{i,\, b_r}} z'$。由于$\mathcal{A}[r]$ 和 $\mathcal{B}[s]$ 内部不相交（除非二者相等），因此，当我们从 $v_{i,\, b_r}$ 走向 $v_{i,\, b_r+1}$时，必须先于$\mathcal{B}[s]$ 到达$\mathcal{A}[r]$，从而满足引理 3(iv)。◀

我们证明，如果 E 是 σ的粗编码，那么 T_E 中的每条曲线都与 σ接近。

▶ **引理 7**。设 E是 σ的粗编码。对于每个 $\tau_i \in T_E$，$d_F(\tau_i, \sigma) \leq (1 + 24\varepsilon)\delta$。

证明。（概要）假设 $T_E \neq \emptyset$，否则引理陈述显然成立。任取 $\tau_i \in T_E$。我们构造 τ_i与 σ之间的一个匹配 M，使得 $d_{\mathcal{M}}(\tau_i, \sigma) \leq (1 + 24\varepsilon)\delta$。由于 $T_E \neq \emptyset$，存在 τ_i的顶点的一个划分 $(\pi_0, \ldots, \pi_{k-1})$，满足引理 3(i)–(iv)。利用这些性质，我们可以将 τ_i 的顶点匹配到 σ 上的点，然后将 σ 的顶点匹配到 τ_i上的点。之后，σ 和 τ_i 由其顶点及其匹配伙伴的像划分为线段。我们使用线性插值来匹配对应的线段。更多细节见附录。◀

在二维和三维中，[8] 中用于盒子的射线投射数据结构可用作 $(11\varepsilon\delta)$ 线段查询数据结构。对于任意固定的 $\mu \in (0,1)$，它具有 $O(\log |\mathcal{G}_1|) = O(\log \frac{mn}{\varepsilon})$ 查询时间，以及$\bar{}O(|\mathcal{G}_1|^{2+\mu}) = O((mn)^{2+\mu}/\varepsilon^{d(2+\mu)})$ 空间和预处理时间。如果查询线段不与 $\mathcal{G}_1$，中的任何单元相交，我们就返回 null。在四维和高维中，我们将在第 4 节证明以下结果。

▶ **引理 8.** 我们可以构造一个数据结构，其空间和预处理时间为 $O(\sqrt{d}/\varepsilon)^{O(d/\varepsilon^2)} \cdot \tilde{O}((mn)^{O(1/\varepsilon^2)})$，使得对于查询曲线 σ 的任意一条有向边 e，该数据结构要么能判定 $\min_{\tau_i \in T} d_F(\sigma, \tau_i) > \delta$，要么能针对 G_1 上包含 e 的$(11\varepsilon\delta)$段查询给出正确答案。查询时间为 $\tilde{O}((mn)^{0.5+\varepsilon}/\varepsilon^d)$。

将本节的结果与 [8], 引理 8 中的射线发射结果结合起来，再应用定理 1，可得到 ($1+\varepsilon$)-近似最近邻数据结构。定理 1 使用了 $(1+\varepsilon,\delta)$-近似最近邻数据结构的删除代价。由于我们没有更高效的方案，因此每次删除时都从头开始重建数据结构。

▶ **定理 9.** 对于任意 $\varepsilon \in (0,0.5)$，存在一个 $(1+O(\varepsilon))$-近似最近邻数据结构，用于计算 T 关于弗雷歇距离的近似最近邻，并具有以下性能保证：

- $d \in \{2,3\}$：查询时间 $= O\left(\frac{1}{\varepsilon}\right)^{2d(k-2)} k \log \frac{mn}{\varepsilon} \log n$，空间 $= O\left(\frac{1}{\varepsilon}\right)^{4d(k-1)+1}(mn)^{4(k-1)} k \log^2 n$，期望预处理时间 $= O\left(\frac{1}{\varepsilon}\right)^{4d(k-1)}(mn)^{4(k-1)}\left(k \log \frac{mn}{\varepsilon} + m^k \log m\right) n \log n$.
- $d \geq 4$：查询时间 $= \tilde{O}\left(\frac{1}{\varepsilon^d} k(mn)^{0.5+\varepsilon}\right) + O\left(\frac{\sqrt{d}}{\varepsilon}\right)^{2d(k-2)} k \log \frac{mn}{\varepsilon} \log n$，空间 $= O\left(\frac{\sqrt{d}}{\varepsilon}\right)^{4d(k-1)}(mn)^{4(k-1)} k \cdot \frac{1}{\varepsilon} \log^2 n + O\left(\frac{\sqrt{d}}{\varepsilon}\right)^{O(d/\varepsilon^2)} \cdot \tilde{O}((mn)^{O(1/\varepsilon^2)})$，期望预处理时间 $= O\left(\frac{\sqrt{d}}{\varepsilon}\right)^{4d(k-1)}(mn)^{4(k-1)}\left(k \log \frac{mn}{\varepsilon} + m^k \log m\right) n \log n + O\left(\frac{\sqrt{d}}{\varepsilon}\right)^{O(d/\varepsilon^2)} \cdot \tilde{O}((mn)^{O(1/\varepsilon^2)})$。

3 $(3+O(\varepsilon)\ \delta)$ (( )) (( ))-近似最近邻

给定一条查询曲线 $\sigma = (w_1, w_2, \ldots\ldots, w_k)$，对于 $j \in [k-1]$，我们像之前一样，使用 $w_j w_{j+1}$ 和 $w_{j+1} w_j$ 在 $\mathcal{G}_1$ 上求解 $(11\varepsilon\delta)$-线段查询。令 $((c_{j,1}, c_{j,2}))_{j\in[k-1]}$ 表示这些查询的结果。回想一下，根据约束1，每个 $(c_{j,1}, c_{j,2})$ 都属于 $(\mathcal{G}_1 \times \mathcal{G}_1) \cup \{\text{null}\}$。

假设 $((c_j,1,c_j,2))_{j\in[k-1]}$ 中有 $k_0 \leq k-1$ 个非空点对。提取这些非空点对，构成序列 $((c_{j_r},1,c_{j_r},2))_{r\in[k_0]}$。注意，由约束1(a)可知，$j_1 = 1$ 和 $j_{k_0} = k-1$。对于 $r \in [k_0]$，我们通过连接 $c_{j_r},1$ 和 $c_{j_r},2$ 的中心，以及 $c_{j_r},2$ 和 $c_{j_{r+1}},1$ 的中心，构造一条折线曲线 σ_0。折线曲线 σ_0 充当 σ 的替代对象。它至多有 $2k-2$ 个顶点。在查询时间，我们将使用 σ_0 作为键搜索字典树，以获得 $(3+O(\varepsilon)\ \delta)$-近似最近邻查询的答案。因此，无需枚举，从而避免了查询时间对 k 的指数级依赖。

在预处理中，我们枚举 G_1 中所有长度为 $l \in [2, k-1]$ 的 $2l$ 单元序列。对于每个序列，我们构造一条折线曲线 σ'，连接序列中各单元的中心，并找到距离 σ' 最近的输入曲线 τ_i。如果 $d_F(\sigma', \tau_i) \leq (1+12\varepsilon)\delta$，我们就在字典树 D 中存储 (σ', i)。D 中有 $O(\sqrt{d}/\varepsilon)^{2d(k-1)}(mn)^{2(k-1)}$ 个条目。我们按照第 2.2 节所述的相同方式组织字典树 D。D 所需的空间为 $O(\sqrt{d}/\varepsilon)^{2d(k-1)}(mn)^{2(k-1)}k$。D 的搜索时间为 $O(k\log \frac{mn}{\varepsilon})$。预处理时间为 $O(\sqrt{d}/\varepsilon)^{2d(k-1)}(mn)^{2(k-1)}(k \log \frac{mn}{\varepsilon} + kmn\log(km)) = O(\sqrt{d}/\varepsilon)^{2d(k-1)}(mn)^{2k-1} k \log \frac{mn}{\varepsilon}$，这是由于需要为枚举出的每个序列计算最近输入曲线。

在查询时，我们以 $O(kQ_{\text{seg}})$ 的时间从 σ 构造 σ_0，其中 Q_{seg} 是回答一个 $(11\varepsilon\delta)$-线段查询所需的时间。我们在 $O(k^2 \log k)$ 时间内计算 $d_F(\sigma, \sigma_0)$。如果 $d_F(\sigma, \sigma_0) > (2+12\varepsilon)\delta$，则报告“否”。否则，我们使用 σ_0 在 $O(k\log \frac{mn}{\varepsilon})$ 时间内搜索 $\mathcal{D}$。如果搜索失败，则报告“否”。否则，搜索会返回某个 $i \in [n]$ 对应的 (σ_0, i)。

▶ **引理 10。** 如果 $d_F(\sigma, \sigma_0) \leq (2+12\varepsilon)\delta$，且使用 σ_0 在 D 中搜索后返回 (σ_0, i)，那么 $d_F(\sigma, \tau_i) \leq (3+24\varepsilon)\delta$。否则，最小值 $\tau_i \in T\ d_F(\sigma, \tau_i) > \delta$。

结合本节的结果、二维和三维中的射线射击结果 [8], 引理 8以及定理 1，我们得到以下定理。

▶ **定理 11**。 对于任意 $\varepsilon \in (0,\ 0.5)$，存在一个$(3+O(\varepsilon))$-近似最近邻数据结构，用于处理 T 关于弗雷歇距离的问题，并具有以下性能保证：

- $d \in \{2, 3\}$：查询时间 $= O(k \log \frac{mn}{\varepsilon} \log n)$，空间 $= O\left(\frac{1}{\varepsilon}\right)^{2d(k-1)+1}(mn)^{2(k-1)} k \log^2 n$，期望预处理时间 $= O\left(\frac{1}{\varepsilon}\right)^{2d(k-1)}(mn)^{2k-1} kn \log \frac{mn}{\varepsilon} \log n$。
- $d \geq 4$：查询时间 $= \tilde{O}\left(\frac{1}{\varepsilon^d} k(mn)^{0.5+\varepsilon}\right)$，空间 $= O\left(\frac{\sqrt{d}}{\varepsilon}\right)^{2d(k-1)}(mn)^{2(k-1)} k \cdot \frac{1}{\varepsilon} \log^2 n + O\left(\frac{\sqrt{d}}{\varepsilon}\right)^{O(d/\varepsilon^2)} \cdot \tilde{O}((mn)^{O(1/\varepsilon^2)})$，期望预处理时间 $= O\left(\frac{\sqrt{d}}{\varepsilon}\right)^{2d(k-1)}(mn)^{2k-1} kn \log \frac{mn}{\varepsilon} \log n + O\left(\frac{\sqrt{d}}{\varepsilon}\right)^{O(d/\varepsilon^2)} \cdot \tilde{O}((mn)^{O(1/\varepsilon^2)})$。

4 $(11\varepsilon\delta)$()()-线段查询及引理 8 的证明

我们将介绍引理 8 中的 $(11\varepsilon\delta)$-线段查询数据结构。首先介绍主要思路，然后再给出详细内容。设 $w_j w_{j+1}$ 是一条查询线段，它在预处理时未知。这里有三个构建模块。

首先，直觉上，我们要用$\mathcal{G}_1$中的单元对来捕获所有可能查询线段的支撑直线。理想情况下，我们可以找出被 aff$(w_j w_{j+1})$ 相交的一对单元，但这似乎和射线射击问题一样困难。出于技术原因，我们需要在输入顶点周围更大的邻域中使用比$\mathcal{G}_1$, 更多的网格单元，因此定义$\mathcal{G}_3 = \bigcup_{i\in[n],\ a\in[m]} G(v_{i,\ a} \oplus B_{(1+6\varepsilon)\delta})$。

我们找到$\mathcal{G}_1$ 的一个网格顶点 x，它是 aff$(w_j w_{j+1})$ 的 $(1+\varepsilon)$-近似最近网格顶点。我们将证明，如果 $d(x,\ \text{aff}(w_j w_{j+1})) > (1+\varepsilon)\varepsilon\delta$，则 $(11\varepsilon\delta)$-线段查询的答案为空；否则，我们可以找到$\mathcal{G}_3$ 中的一个单元 $\gamma \in$，它靠近 x 且与 aff$(w_j w_{j+1})$ 相交。我们可以使用$\mathcal{G}_1$ 中的任意其他单元 $c \in$ 与 γ 组成一对，作为经过 c 和 γ 附近的查询线段支撑直线的替代对象。

其次，给定查询时的 $w_j w_{j+1}$，在所有可能的 c 选择中，我们需要高效地找到合适的一个（或多个），使得 (c,γ) 是 aff$(w_j w_{j+1})$ 的替代对象。我们以 w_{j+1} 位于 w_j 与 aff$(w_j w_{j+1}) \cap \gamma$ 之间的情形来说明这些想法。注意，w_j 可能不接近 G_1 中的任何单元。为了使 $\min_{\tau_i \in T} d(\sigma, \tau_i) \leq \delta$，$w_j$ 必须位于距某条输入边 τ_i 不超过 δ 的范围内。我们使用与 $\tau_{i,a}$ 平行且彼此间距至少为 $\Theta(\varepsilon\delta)$ 的直线，对 aff$(\tau_{i,a}) \oplus B_{O(\delta)}$ 作极大装填。该装填中有 $O(\varepsilon^{1-d})$ 条直线，并且 aff$(\tau_{i,a}) \oplus B_\delta$ 中的每个点到装填中某条直线的距离都不超过 $O(\varepsilon\delta)$。将 w_j 投影到近似最近的直线上，可以近似确定 w_j 的位置。因此，我们应将装填中的直线划分为适当的线段，以便给定 w_j 及其在装填中近似最近的直线后，能够高效地找到包含 w_j 投影的线段，并获取该线段的一些预计算信息。

第三，设 ℓ 为上述装填中提到的一条直线，对于每个可能的单元 $c \in \mathcal{G}_1,$，我们使用几何构造 $F(c,\gamma) = \{x \in \mathbb{R}^d\colon \exists\ y \in \gamma$ 使得 $xy \cap c \neq \emptyset\}$，该构造在 [6] 中定义，可以在 $O(1)$ 时间内计算得到。$(\ell \oplus B_{2\varepsilon\delta}) \cap F(c,\gamma)$ 在 ℓ 中的投影，是 ℓ 上的点集，满足：如果 w_j 的投影位于其中，那么 (c,γ) 就是 aff$(w_j w_{j+1})$ 的替代对象。因此，对于 c 的所有可能选择，$(\ell \oplus B_{2\varepsilon\delta}) \cap F(c,\gamma)$ 的投影端点将 ℓ 划分为我们所需的线段。每条线段可能代表 c 的多种选择。对于每条线段，我们存储靠近该线段的单元 c'，因为理想的选择是从 w_j 向 w_{j+1} 行进时最先遇到的单元。

如上所述，我们使用了两种涉及

直线的近似最近邻数据结构。第一种由 Andoni 等人提出 [4]，它存储一组点 P，给定一条查询直线后，可在 $\tilde{O}(d^3|P|^{0.5+\varepsilon})$ 时间内返回距离该查询直线最近的 $(1+\varepsilon)$-近似点。它使用 $\tilde{O}\left(d^2|P|^{O(1/\varepsilon^2)}\right)$ 空间和预处理时间。第二种结果由 Agarwal 等人提出，[2]它存储一组 L 直线，给定一个查询点后，可在 $\tilde{O}(1)$ 时间内返回距离该查询点最近的 2-近似直线。它使用 $\tilde{O}(|L|^2)$ 空间和期望预处理时间。

4.1 数据结构组织

我们将 ε 限制为从 (0，0．5) 中选取。我们为 $\mathcal{G}_1$ 的网格顶点构建 Andoni 等人 [4] 的数据结构，从而对于任意查询直线，都可以在 $\tilde{O}((mn)^{0.5+\varepsilon}/\varepsilon^d)$ 时间内返回 $(1+\varepsilon)$-近似最近网格顶点。我们将该数据结构记为 D_{anp}。它需要 $O(\sqrt{d}/\varepsilon)^{O(d/\varepsilon^2)}\cdot\tilde{O}((mn)^{O(1/\varepsilon^2)})$ 的空间和预处理时间。

对于每条输入边 τ_i、$_a$，定义直线集合 $L_{i,a}$ 如下。设 H 为经过 v_i、$_a$ 且正交于 aff(τ_i、$_a$) 的超平面。在 H 中取一个 $(d-1)$ 维网格，以 v_i、$_a$ 为一个网格顶点，单元宽度为 $\varepsilon\delta/\sqrt{d-1}$。集合 $L_{i,a}$ 包含所有正交于 H 且经过 H 中该网格的某个顶点、与 v_i、$_a$ 的距离在 $(1+2\varepsilon)\delta$ 以内的直线。集合 $L_{i,a}$ 的大小为 $O(\varepsilon^{1-d})$，并且可以在 $O(\varepsilon^{1-d})$ 时间内构造。此外，圆柱体 aff(τ_i、$_a$) $\oplus B_\delta$ 中的每个点到集合 $L_{i,a}$ 中某条直线的距离都不超过 $\varepsilon\delta$。

定义 $\mathcal{L}=\bigcup_{i\in[n],a\in[m-1]}L_{i,a}$。$\mathcal{L}$ 的大小为 $O(mn/\varepsilon^{d-1})$，且可在 $O(mn/\varepsilon^{d-1})$ 时间内构造。我们为$\mathcal{L}$ 构建 Agarwal 等人 [2] 的数据结构，从而对于˜任意查询点，都能在$\mathcal{L}$ 中以 $O(1)$ 时间返回一条 2-近似最近直线。我们将此数据结构记为 D_{anl}。它使用 $\tilde{O}((mn)^2/\varepsilon^{2d-2})$ 空间和期望预处理时间。

回忆一下，$\mathcal{G}_3=\bigcup_{i\in[n],a\in[m]}G(v_{i,a}\oplus B_{(1+6\varepsilon)\delta})$。

对于每个 $\gamma\in\mathcal{G}_3$ 以及每个 $c\in\mathcal{G}_1$，，我们构造 $F(c,\gamma)=\{x\in\mathbb{R}^d:\exists y\in\gamma$ 使得 $xy\cap c\neq\emptyset\}$，它是空集或一个大小为 $O(1)$ 的无界凸多面体，可在$O(1)$ 时间内通过闵可夫斯基和构造 [6]。所需总时间为 $O((mn)^2/\varepsilon^{2d})$。

对于每个 $\gamma\in\mathsf{G}_3$，、每个 $c\in\mathsf{G}_1$，以及每条直线 $\ell\in\mathsf{L}$，计算交集 $(\ell\oplus B_{2\varepsilon\delta})\cap F(c,\gamma)$，并将其正交投影到 ℓ中的一条线段上。任取一条直线 $\ell\in\mathsf{L}$。所得线段的端点在 ℓ 中将 ℓ 划分为规范线段。ℓ中共有 $O((mn)^2/\varepsilon^{2d})$ 条规范线段。对于每个单元 $\gamma\in\mathsf{G}_3$ 和每条规范线段 $\xi\subseteq\ell$，计算集合 $C_{\gamma,\xi}$，其中包含所有满足 ξ包含在 $(\ell\oplus B_{2\varepsilon\delta})\cap F(c,\gamma)$ 在 ℓ上的投影中的单元 $c\in\mathsf{G}_1$。在 ξ 中任取一个点，并记为 p_ξ。每个 $C_{\gamma,\xi}$ 的大小为 $O(mn/\varepsilon^d)$。遍历 G_3 中的所有单元以及 L 中所有直线上的所有规范线段所需的总时间为 $\tilde{O}((mn)^5/\varepsilon^{5d-1})$。

令 p_γ 为单元 γ的中心。定义 $c_{\gamma,\xi}$ 为 $C_{\gamma,\xi}$ 中的单元，使得 $p_\xi p_\gamma\cap(c_{\gamma,\xi}\oplus B_{5\varepsilon\delta})$ 在 $\{p_\xi p_\gamma\cap($ $c\oplus B_{5\varepsilon\delta}):c\in C_{\gamma,\xi}\}$中距离 p_ξ最近。在$\mathcal{L}$ 中所有直线上的所有单元和所有规范线段上计算$c_{\gamma,\xi}$ 的总时间为 $O((mn)^5/\varepsilon^{5d-1})$。

最后，对于每条直线 $\ell\in\mathcal{L}$，我们将 ℓ 中的规范线段存储在T_ℓ [7]区间树中。它使用线性空间，且预处理时间为线性。对于 ℓ中的任意查询点，我们可以在 O中以 O(log $\frac{mn}{\varepsilon}$) 时间搜索 T_ℓ，找到包含该查询点的规范线段。对于存储在 T_ℓ中的每条规范线段 ξ，我们维护一个字典 T_ξ，其中存储集合$\{(\gamma,c_{\gamma,\xi}):\gamma\in\mathcal{G}_3\}$，并以 为键。对于任意单元 G_3，，我们可以在 O中以 O(log $\frac{mn}{\varepsilon}$) 时间搜索 T_ξ，并报告 $c_{\gamma,\xi}$。这些区间树和字典的总大小为 $O((mn)^4/\varepsilon^{4d-1})$，并且可以在 $\tilde{O}((mn)^4/\varepsilon^{4d-1})$ 时间内构建。

我们需要支持在 $\mathcal{G}_1$ 上执行 $(11\varepsilon\delta)$ 线段查询所需的数据结构为 D_{anp}, D_{anl}, T_ℓ，用于 $\ell\in\mathcal{L}$，以及 T_ξ，用于所有规范线段 ξ's。

▶ **引理 12.** 我们可以为 $\ell\in L$ 构造 D_{anp}, D_{anl}, T_ℓ，并为每个 $\ell\in L$ 以及 $O(\sqrt{d}/\varepsilon)^{O(d/\varepsilon^2)}\cdot\tilde{O}((mn)^{O(1/\varepsilon^2)})$ 空间中的每条规范线段 $\xi\subset\ell$ 构造 T_ξ，同时满足相应的预处理时间。

在 $c_{\gamma,\xi}$ 的定义中，我们可能会问：如果对于某个$c \in C_{\gamma,\xi}$，$p_\xi p_\gamma$ 与 $c \oplus B_{5\varepsilon\delta}$ 不相交，会怎样？我们证明这种情况不可能发生。我们还确立了其他一些性质。

►**引理 13**。设 γ 是 G_3中的一个单元。设 ξ 是一条规范线段。设 L_ξ 是一个圆柱体，其轴为 ξ，半径为 $2\varepsilon\delta$。

(i) 对于 G_1，中的每个单元 $c \in$，若对于某些点 $x \in L_\xi$ 和 $y \in \gamma$，有 $c \cap xy \neq \emptyset$，则 $c \in C_{\gamma,\,\xi}$。**(ii)** 对于每个点 $x \in L_\xi$、每个点 $y \in \gamma$ 以及每个单元 $c \in C_{\gamma,\,\xi}$, $xy \cap (c \oplus B_{5\varepsilon\delta}) \neq \emptyset$。**(iii)** 设 λ 是一个大于或等于 $11\varepsilon\delta$ 的任意值。从点 $x \in L_\xi$ 走向点 $y \in \gamma$ 时，无论如何选择 x 和 y，我们都不可能在早于 $c_{\gamma,\,\xi} \oplus B_\lambda$ 的时候遇到任何 $c \in C_{\gamma,\,\xi}$。

4.2 回答查询

给定查询曲线 σ的一条有向线段 $w_j w_{j+1}$，我们通过以下步骤，使用 $w_j w_{j+1}$ 在$\mathcal{G}_1$ 上回答（$11\varepsilon\delta$）线段查询。

- 步骤 1：我们使用 aff($w_j w_{j+1}$) 查询 D_{anp}，报告 G_1 的一个网格顶点 x。这需要 $\tilde{O}((mn)^{0.5+\varepsilon}/\varepsilon^d)$ 时间。步骤 2：我们检查距离 $d(x, \text{aff}(w_j w_{j+1}))$。如果 $d(x, \text{aff}(w_j w_{j+1})) > (1+\varepsilon)\varepsilon\delta$，那么 aff($w_j w_{j+1}$) 到 G_1，中最近网格顶点的距离大于 $\varepsilon\delta$，这意味着 aff($w_j w_{j+1}$) 不与 G_1 中的任何单元相交。在这种情况下，我们报告 null。˜我们还检查距离 $d(w_j, L)$ 和 $d(w_{j+1}, L)$。我们在 $O(1)$ 时间内使用 w_j 查询 D_{anl}，以找到一条直线 $\ell_j \in L$。如果 $d(w_j, \ell_j) > 2\varepsilon\delta$，则 $d(w_j, L) > \varepsilon\delta$，这意味着对于任意 $\tau_i \in T$ 和任意 $a \in [m-1]$，w_j 到 aff(τ_i,a) 的距离都大于 δ。因此，对于所有 $\tau_i \in T$，$d_F(\sigma, \tau_i) > \delta$，所以我们对（$\kappa, \delta$）-近似最近邻查询报告“否”。类似地，我们在 $O(1)$ 时间内使用 w_{j+1} 查询 D_{anl}，以找到一条直线 $\ell_{j+1} \in L$。如果 $d(w_{j+1}, \ell_{j+1}) > 2\varepsilon\delta$，我们就对（$\kappa, \delta$）-近似最近邻查询报告“否”。
- 步骤 3：假设 $d(x, \text{aff}(w_j w_{j+1})) \leq (1+\varepsilon)\varepsilon\delta$，$d(w_j, \ell_j) \leq 2\varepsilon\delta$，且 $d(w_{j+1}, \ell_{j+1}) \leq 2\varepsilon\delta$。接着，我们用 $O(\varepsilon^{-d})$ 时间检查 $G(x \oplus B_{2\varepsilon\delta})$ 中的单元，找出一个与 aff($w_j w_{j+1}$) 相交的单元。令 γ 为这个单元。我们不知道 γ 是否属于$\mathcal{G}_1$。然而，由于 x 是$\mathcal{G}_1$的网格顶点，γ 与某个输入曲线顶点的距离至多为 $(1+3\varepsilon)\delta$。因此，γ 必定是$\mathcal{G}_3$中的一个单元。根据 w_j 和 γ的相对位置，共有三种情况。
  - 步骤 3(a)：$w_j \in \gamma \cap$ aff($w_j w_{j+1}$)。我们断言 $G_1 \cap G(w_j \oplus B_{7\varepsilon\delta})$ 非空，并报告其中任意一个单元作为 ($11\varepsilon\delta$)-线段查询的答案。此步骤耗时 $O(\varepsilon^{-d})$。步骤 3(b)：沿着从 w_j 到 w_{j+1}定向的 aff($w_j w_{j+1}$)，w_j 位于 aff($w_j w_{j+1}$) 之前。我们查询 T_{ℓ_j}，找出
  - 包含 w_j在 ℓ_j中的投影的规范线段 $\xi \subset \ell_j$。然后，我们用 γ 查询 T_ξ，返回 $c_{\gamma,\xi}$ 作为 ($11\varepsilon\delta$)-线段查询的答案。所需时间为 $O(\log \frac{mn}{\varepsilon})$。步骤 3(c)：沿着从 w_j 到 w_{j+1} 定向的 aff($w_j w_{j+1}$)，$\gamma \cap$aff($w_j w_{j+1}$) 位于 w_j 之前。我们查询 $T_{\ell_{j+1}}$，找出包含 w_{j+1}在 ℓ_{j+1}中的投影的规范线段 $\xi \subset \ell_{j+1}$。然后，我们用 γ 查询 T_ξ，得到 $c_{\gamma,\xi}$。
  - 我们断言 $G(c_{\gamma,\xi} \oplus B_{5\varepsilon\delta}) \subset G_3$，并且 $G(c_{\gamma,\xi} \oplus B_{5\varepsilon\delta})$ 中有一个单元与 $w_j w_{j+1}$ 相交。在 $O(\varepsilon^{-d})$ 时间内，任选一个这样的单元 ˆγ。将 γ 替换为 ˆγ后，步骤 3(a) 或 3(b) 之一适用。无论适用哪种情况，我们都转到该情况，将 γ 替换为 ˆγ，以返回 ($11\varepsilon\delta$)-线段查询的答案。所需时间为 $\tilde{O}(\varepsilon^{-d})$。

►**引理 14.** 回答一次$(11\varepsilon\delta)$线段查询需要 $\tilde{O}((mn)^{0.5+\varepsilon}/\varepsilon^d)$ 时间。

引理 12 和引理 14 给出了引理 8 中 $(11\varepsilon\delta)$-线段查询数据结构的性能。在附录 G 中，我们证明了引理 8 中查询输出的正确性。

5 结论

我们提出了 $(1+\varepsilon)$-近似最近邻和 $(3+\varepsilon)$-近似最近邻数据结构，它们能够实现次线性的查询时间，且空间复杂度既不与 $\min\{m^{\Omega(d)}, n^{\Omega(d)}\}$ 成正比，也不会随 $\min\{m,n\}$ 呈指数增长。$(1+\varepsilon)$-近似最近邻的查询时间为 $\tilde{O}(k(mn)^{0.5+\varepsilon}/\varepsilon^{O(d)} + k(d/\varepsilon)^{O(dk)})$，而 $(3+\varepsilon)$-近似最近邻的查询时间为 $\tilde{O}(k(mn)^{0.5+\varepsilon}/\varepsilon^{O(d)})$。在二维和三维中，$(1+\varepsilon)$-近似最近邻的查询时间可改进至 $\tilde{O}(k/\varepsilon^{O(k)})$，而 $(3+\varepsilon)$-近似最近邻的查询时间可改进至 $\tilde{O}(k)$。如何降低对 d 和 k 的指数依赖仍是一个开放问题。

参考文献

1 P.K. Agarwal 和 J. Matoušek。射线发射与参数搜索。《SIAM 计算学报》，22(4):794–806，1993年。 2 P.K. Agarwal、N. Rubin 和 M. Sharir。高维平坦空间中的近似最近邻搜索。载于《欧洲算法研讨会论文集》，第 4:1–4:13 页，2017年。 3 H. Alt 和 M. Godau。计算两条多边形曲线之间的弗雷歇距离。《国际计算几何与应用杂志》，5:75–91，1995年。 4 A. Andoni、P. Indyk、R. Krauthgamer 和 H.L. Nguyen。高维空间中的近似直线最近邻。载于《ACM-SIAM 离散算法研讨会论文集》，第 293–301 页，2009年。 5 K. Bringmann、A. Driemel、A. Nusser 和 I. Psarros。弗雷歇距离下时间序列近似近邻搜索的紧确界。载于《ACM-SIAM 离散算法研讨会论文集》，第 517–550 页，2022年。 6 S.-W. Cheng 和 H. Huang。弗雷歇距离下的曲线简化与聚类。载于《ACM-SIAM 离散算法研讨会论文集》，第 1414–1432 页，2023年。 7 T.H. Cormen、C.E. Leiserson、R.L. Rivest 和 C. Stein。《算法导论》。麻省理工学院出版社，第二版，2001年。 8 M. de Berg、D. Halperin、M. Overmars、J. Snoeyink 和 M. van Kreveld。高效射线发射与隐藏面消除。Algorithmica，12:30–53，1994年。 9 A. Driemel、A. Krivošija 和 C. Sohler。弗雷歇距离下的时间序列聚类。载于《ACM-SIAM 离散算法研讨会论文集》，第 766–785 页，2016年。 10 A. Driemel 和 I. Psarros。弗雷歇距离下时间序列的 $(2+\epsilon)$-近似最近邻。arXiv 预印本 arXiv:2008.09406v5，2021年。 11 A. Driemel 和 F. Silvestri。曲线的局部敏感哈希。载于《国际计算几何研讨会论文集》，第 37:1–37:16 页，2017年。 12 T. Eiter 和 H. Mannila。离散弗雷歇距离的计算。技术报告 CD-TR 94/64，Christian Doppler 专家系统实验室，维也纳工业大学，奥地利，1994年。 13 I.Z. Emiris 和 I. Psarros。应用于曲线邻近问题的欧几里得度量乘积：对离散弗雷歇距离和动态时间规整距离的统一处理。《ACM 空间算法与系统汇刊》，6(4):1–20，2020年。 14 A. Filtser、O. Filtser 和 M.J. Katz。曲线的近似最近邻：简单、高效且具有确定性。载于《自动机、语言与编程国际研讨会论文集》，第 48:1–48:19 页，2020年。 15 S. Har-Peled。一种近线性规模的沃罗诺伊图替代方案。载于《IEEE 计算机科学基础年会论文集》，第 94–103 页，2001年。 16 S. Har-Peled、P. Indyk 和 R. Motwani。近似最近邻：朝着消除维数灾难迈进。《计算理论》，8:321–350，2012年。 17 P. Indyk。通过乘积度量实现弗雷歇距离的近似最近邻算法。载于《计算几何年会论文集》，第 102–106 页，2002年。 18 P. Indyk 和 R. Motwani。近似最近邻：朝着消除维数灾难迈进。载于《ACM 计算理论年会论文集》，第 604–613 页，1998年。 19 Mirzanezhad M. 关于弗雷歇距离下曲线之间的近似最近邻查询。arXiv 预印本 arXiv:2004.08444，2020年。 20 C. Shahabi、M. Kolahdouzan 和 M. Sharifzadeh。一种用于移动对象数据库中k-最近邻搜索的道路网络嵌入技术。GeoInformatica，7:255–273，2003年。 21 Z. Song 和 N. Roussopoulos。移动查询点的 K-最近邻搜索。载于《空间与时间数据库国际研讨会论文集》，第 79–96 页，2001年。 22 Y. Tao 和 D. Papadias。时空数据库中的参数化查询。载于《ACM 数据管理国际会议论文集》，第 334–345 页，2002年。

引理 3 的证明

D 的性能分析可由前面的讨论得出。在四个充要条件中，(i)、(iii) 和 (iv) 可分别由第一、第三和第四项测试直接推出。下面证明第二项测试与条件 (ii) 等价。显然，若第二项测试成功，则条件 (ii) 成立。剩下需要分析另一个方向。根据条件 (ii)，$x''_j \leq_{x_j y_j} y''_j$, $x''_j \in v_{i,a_j} \oplus B_{(1+12\varepsilon)\delta}$，以及 $y''_j \in v_{i,b_j} \oplus B_{(1+12\varepsilon)\delta}$。在第二项测试中，我们相对于 $\leq_{x_j y_j}$，计算 $x_j y_j \cap (v_{i,a_j} \oplus B_{(1+12\varepsilon)\delta})$ 中的最小点 x'_j 和 $x_j y_j \cap (v_{i,b_j} \oplus B_{(1+12\varepsilon)\delta})$ 中的最大点 y'_j。由此可得$x'_j \leq_{x_j y_j} x''_j \leq_{x_j y_j} y''_j \leq_{x_j y_j} y'_j$。我们将 $x''_j y''_j$ 与$\tau_i[v_{i,a_j}, v_{i,b_j}]$之间的弗雷歇匹配扩展为 $x'_j y'_j$与 $\tau_i[v_{i,a_j}, v_{i,b_j}]$之间的匹配 M：将 v_{i,a_j}与 $x'_j x''_j$中的所有点匹配，并将 v_{i,b_j}与 $y''_j y'_j$ 中的所有点匹配。由于 $x'_j, x''_j \in v_{i,a_j} \oplus B_{(1+12\varepsilon)\delta}$，根据凸性可知 $x'_j x''_j \subset v_{i,a_j} \oplus B_{(1+12\varepsilon)\delta}$。类似地，$y''_j y'_j \subset v_{i,b_j} \oplus B_{(1+12\varepsilon)\delta}$。因此，$d_F(x'_j y'_j, \tau_i[v_{i,a_j}, v_{i,b_j}]) \leq d_{\mathcal{M}}(x'_j y'_j, \tau_i[v_{i,a_j}, v_{i,b_j}]) \leq (1+12\varepsilon)\delta$。

B 引理 4 的证明

如果 $x_r x_s \cap (w_{r+1} \oplus B_{(1+\varepsilon)\delta})$ 或 $x_r x_s \cap (w_s \oplus B_{(1+\varepsilon)\delta})$ 为空，则所需的 x'_r 和 x'_s 不存在。否则，令 p 为 $x_r x_s \cap (w_{r+1} \oplus B_{(1+\varepsilon)\delta})$ 中关于 $\leq_{x_r x_s}$ 的最小点。令 q 为 $x_r x_s \cap (w_s \oplus B_{(1+\varepsilon)\delta})$ 中的最大点。我们可以检查 $p \leq_{x_r x_s} q$，并在 $O((s-r)\log(s-r))$ 时间内计算 $d_F(pq, \sigma[w_{r+1}, w_s])$。我们断言，存在一条满足该引理的线段 $x'_r x'_s \subseteq x_r x_s$，当且仅当 pq 满足该引理。反向显然成立。正向可以用与引理 3 的证明相同的方法证明。

C 引理 5 的证明

我们花费 $O(\log \frac{mn}{\varepsilon})$ 的时间来获得 B[1] 和 A[$k-1$]。然后，我们花费 $O(kQ_{\text{seg}})$ 的时间，为 $j \in [k-1]$ 获得 (u_j,1, u_j,2)，并为 σ 的每种粗编码花费 $O(k \log \frac{mn}{\varepsilon})$ 的搜索时间。在设置 (c_j,1,c_j,2) 时，对于 $j \in [2, k-2]$，将其设为 (u_j,1, u_j,2) 或 null，共有 2^{k-3} 种组合。这就产生了 2^{k-3} 种可能的 C。根据约束 3(b)，对于 $j \in [1, k-2]$，我们有 A[j] $\in G(c_j,2 \oplus B_{(1+11\varepsilon)\delta})$；对于 $j \in [2, k-1]$，我们有 B[j] $\in G(c_j,1 \oplus B_{(1+11\varepsilon)\delta})$。因此，对于每个枚举出的 C，设置 A 和 B 有 $O(\sqrt{d/\varepsilon})^{2d(k-2)}$ 种方式。总而言之，枚举并检查的 (A, B, C) 的总数为 $O(\sqrt{d/\varepsilon})^{2d(k-2)} 2^{k-3} = O(\sqrt{d/\varepsilon})^{2d(k-2)}$。根据引理 4，检查每一种情况需要 $O(k \log k)$ 的时间。

D 引理 7 的证明

假设 $T_E \neq \emptyset$，否则引理陈述是平凡成立的。任取 $\tau_i \in T_E$。我们构造一个 τ_i 与 σ 之间的匹配 $\mathcal{M}$，使得 $d_{\mathcal{M}}(\tau_i, \sigma) \leq (1+24\varepsilon)\delta$。由于 $T_E \neq \emptyset$，存在一个 τ_i 的顶点划分 $(\pi_0, \ldots, \pi_{k-1})$，满足引理 3(i)–(iv)。

我们首先按如下方式将 τ_i 的顶点与 σ 上的点匹配。我们分别将 $v_{i,1}$ 和 $v_{i,m}$ 与 w_1 和 w_k 匹配。根据约束 2 和引理 3(iii)，我们有 $d(v_{i,1}, w_1) \leq (1+\varepsilon)\delta$，以及 $d(v_{i,m}, w_k) \leq (1+\varepsilon)\delta$。根据引理 3(ii)，对于 $j \in [k-1]$，若 $\pi_j \neq \emptyset$ 且 $\pi_j = \{v_{i,a_j}, v_{i,a_j+1}, \ldots, v_{a,b_j}\}$，则存在一条线段 $x''_j y''_j$，其端点是 c_j,1 和 c_j,2 的顶点，使得 $v_{i,a_j}, \ldots, v_{i,b_j}$ 可以按顺序匹配到 $x''_j y''_j$ 上的某些点 $p_{i,a_j}, \ldots, p_{i,b_j} \in x''_j y''_j$，且距离不超过 $(1+12\varepsilon)\delta$。根据定义，(c_1,1,c_1,2) $\neq$ null；引理 3(i) 表明，若 $j \in [2, k-1]$ 且 $\pi_j \neq \emptyset$，则 (c_j,1, c_j,2) $\neq$ null。然后，约束 1(b) 确保，若 $j = 1$ 或

$\pi_j \neq \emptyset$，则存在两个点 $z_j \in w_j w_{j+1} \cap (c_{j,1} \oplus B_{11\varepsilon\delta})$ 和 $z'_j \in w_j w_{j+1} \cap (c_{j,2} \oplus B_{11\varepsilon\delta})$，使得 $z_j \leq_{w_j w_{j+1}} z'_j$。由于 x''_j 和 y''_j 分别是 $c_{j,1}$ 和 $c_{j,2}$ 的顶点，因此我们有 $d(z_j, x''_j) \leq 12\varepsilon\delta$，以及 $d(z'_j, y''_j) \leq 12\varepsilon\delta$。因此，在 $x''_j y''_j$ 和 $z_j z'_j$ 之间进行线性插值，可将 $p_{i,a_j}, \ldots, p_{i,b_j}$ 映射到点 $q_{i,a_j}, \ldots, q_{i,b_j} \in z_j z'_j$，且距离不超过 $12\varepsilon\delta$。总之，对于 $l \in [a_j, b_j]$，我们可以将 $v_{i,l}$ 与 $q_{i,l}$ 匹配，并且 $d(v_{i,l}, q_{i,l}) \leq d(v_{i,l}, p_{i,l}) + d(p_{i,l}, q_{i,l}) \leq (1 + 24\varepsilon)\delta$。这样就完成了将 τ_i 的顶点与 σ 上的点进行匹配。

接下来，我们将 σ 的顶点按如下方式匹配到 τ_i 上的点。顶点 w_1 和 w_k 分别已与 $v_{i,1}$ 和 $v_{i,m}$ 匹配。任取一个顶点 w_j，其中 $j \in [2, k-1]$。存在唯一的 $(r, s) \in \mathsf{J}$，使得 $r < j \leq s$。如果 $r = 1$ 和 $\pi_1 = \emptyset$，则令 $b_1 = 1$；否则，令 $b_r = \max\{b : v_{i,b} \in \pi_r\}$。根据引理 3(iv)，存在两个点 $y_r \in \tau_{i,b_r} \cap \mathsf{A}[r]$ 和 $y_s \in \tau_{i,b_r} \cap \mathsf{B}[s]$，使得 $y_r \leq_{\tau_{i,b_r}} y_s$。根据约束 3(c)，存在一个将 $w_{r+1}, \ldots, w_s$ 匹配到线段 $x_r x_s$ 中点 $z_{r+1}, \ldots, z_s$ 的匹配，其中该线段位于 $\mathsf{A}[r]$ 和 $\mathsf{B}[s]$ 的顶点之间，并且对于所有 $l \in [r+1, s]$，有 $z_{r+1} \leq_{x_r x_s} z_{r+2} \leq_{x_r x_s} \cdots \leq_{x_r x_s} z_s$ 和 $d(z_l, w_l) \leq (1+\varepsilon)\delta$。$x_r x_s$ 与 $y_r y_s$ 之间的线性插值将 $z_{r+1}, \ldots, z_s$ 映射到 τ_{i,b_r} 中的点，且距离不超过 $\varepsilon\delta$。将该线性插值与把 $w_{r+1}, \ldots, w_s$ 匹配到 $z_{r+1}, \ldots, z_s$ 的匹配相结合，就得到一个将 $w_{r+1}, \ldots, w_s$ 按顺序匹配到 τ_{i,b_r} 上点的匹配，其距离不超过 $(1 + 2\varepsilon)\delta$。这样就完成了将 σ 的顶点匹配到 τ_i 上点的工作。

至此，我们已获得 τ_i 和 σ 上的顶点及其匹配伙伴。在 σ 中，σ 的顶点与 τ_i 的顶点的匹配伙伴将 σ 划分为一系列线段。类似地，τ_i 的顶点与 σ 的顶点的匹配伙伴将 τ_i 划分为一系列线段。我们通过在 τ_i 和 σ 上每一对对应线段之间进行线性插值，完成匹配。由此得到的距离不超过将 τ_i 的顶点匹配到 σ 中的点所得匹配的距离上界 $(1 + 24\varepsilon)\delta$。

E 引理 10 的证明

设 σ_0 是为 σ 生成的折线曲线。如果在 D 中的搜索返回 (σ_0, i)，则 D 的构造保证 $d_F(\sigma_0, \tau_i) \leq (1 + 12\varepsilon)\delta$。因此，$d_F(\sigma, \tau_i) \leq d_F(\sigma, \sigma_0) + d_F(\sigma_0, \tau_i) \leq (3 + 24\varepsilon)\delta$。接下来只需证明：如果 $\min_{\tau_i \in T} d_F(\sigma, \tau_i) \leq \delta$，那么 $d_F(\sigma, \sigma_0) \leq (2 + 12\varepsilon)\delta$，并且使用 σ_0 在 D 中进行搜索将会成功。

设 τ_i 是实数集 T 中满足 $d_F(\sigma, \tau_i) \leq \delta$ 的曲线。设 $\mathcal{M}$ 是 σ 与 τ_i 之间的弗雷歇匹配。设 $((c_{j_r,1}, c_{j_r,2}))_{r \in [k_0]}$ 是从 σ 构造 σ_0 时得到的非空单元对。根据构造，对于 $j \in [j_r + 1, j_{r+1} - 1]$，$(c_{j,1}, c_{j,2}) =$ 为空，因此 $w_j w_{j+1}$ 不与 $\mathcal{G}_1$，中的任何单元相交，这意味着 $w_j w_{j+1}$ 到任意输入顶点的距离都大于 δ。因此，$\mathcal{M}$ 必须将 $\sigma[w_{j_r+1}, w_{j_{r+1}}]$ 与 τ_i 的某条边匹配，设该边为 τ_{i,a_0}。

根据 $(11\varepsilon\delta)$-线段查询的定义，从 w_{j_r+1} 走到 w_{j_r} 时，我们必定会在某一点（设为 x_r）到达 $c_{j_r,2} \oplus B_{11\varepsilon\delta}$。同理，从 $w_{j_{r+1}}$ 走到 $w_{j_{r+1}+1}$ 时，我们必定会在某一点（设为 y_{r+1}）到达 $c_{j_{r+1},1} \oplus B_{11\varepsilon\delta}$。

我们声称，$v_{i,a}$ 与 $\mathcal{M}$ 中的一个点匹配，该点位于 σ 中，且不在 x_r 之后。否则，当我们从 w_{j_r+1} 走到 w_{j_r} 时，在到达 x_r 之前，必定会经过 $G(v_{i,a} \oplus B_\delta)$ 中的一个单元，但这与 $c_{j_r,2}$ 是以 $w_{j_r+1} w_{j_r}$ 进行的 $(11\varepsilon\delta)$-线段查询的输出这一事实矛盾。类似地，$v_{i,a+1}$ 与 $\mathcal{M}$ 中的一个点匹配，该点位于 σ 中，且不在 y_{r+1} 之前。图 4 展示了示意图。

因此，M 将子曲线 $\sigma[x_r, y_{r+1}]$ 与线段 $p_r q_{r+1} \subseteq \tau_{i,a}$ 匹配。所以 $d(p_r, x_r)$ 和 $d(q_{r+1}, y_{r+1})$ 至多为 δ，这意味着 $d_F(p_r q_{r+1}, x_r y_{r+1}) \leq \delta$。因此，我们可以将 M 与 $p_r q_{r+1}$ 和 $x_r y_{r+1}$ 之间的弗雷歇匹配结合起来，从而证明 $d_F(\sigma[x_r, y_{r+1}], x_r y_{r+1}) \leq 2\delta$。将匹配从 x_r 延伸到 c_{j_r} 的中心 z'_r，并从 y_{r+1} 延伸到 $c_{j_{r+1}}$ 的中心 z_{r+1}，即可得到 $d_F(\sigma[x_r, y_{r+1}], z'_{j_r} z_{r+1}) \leq (2 + 12\varepsilon)\delta$。显然，

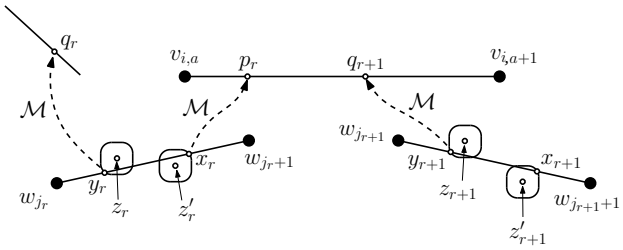

图 4 从左到右，弯曲正方形分别为 $c_{j_r,\ 1} \oplus B_{11\varepsilon\delta}$, $c_{j_r,\ 2} \oplus B_{11\varepsilon\delta}$, $c_{j_{r+1},\ 1} \oplus B_{11\varepsilon\delta}$ 和 $c_{j_{r+1},\ 2} \oplus B_{11\varepsilon\delta}$。

$d_F(y_r x_r, z_r z'_r) \le 12\varepsilon\delta$，因为 $d(y_r, z_r)$ 和 $d(x_r, z'_r)$ 至多为 $12\varepsilon\delta$。因此，我们可以结合 $y_r x_r$ 和 $z_r z'_r$ 之间针对 $r \in [k_0]$ 的弗雷歇匹配，以及 $\sigma[x_r, y_{r+1}]$ 和 $z'_r z_{r+1}$ 之间针对 $r \in [k_0 - 1]$ 的弗雷歇匹配，从而得出 $d_F(\sigma, \sigma_0) \le (2 + 12\varepsilon)\delta$。因此， σ_0 将通过查询过程的检查。

我们此前已证明 $d_F(p_r q_{r+1}, x_r y_{r+1}) \le \delta$。取 $p_r q_{r+1}$ 与 $x_r y_{r+1}$ 之间的一个弗雷歇匹配。将其从 x_r 延伸至 z'_r，并从 y_{r+1} 延伸至 z_{r+1}，得到 $d_F(p_r q_{r+1}, z'_r z_{r+1}) \le (1+12\varepsilon)\delta$。回想一下，M 将子曲线 $\sigma[x_r, y_{r+1}]$ 与 $p_r q_{r+1} \subseteq \tau_{i,a}$ 匹配。因此，M 将 $\sigma[y_r, x_r] = y_r x_r$ 与 $\tau_i[q_r, p_r]$ 匹配，所以 $d_F(y_r x_r, \tau_i[q_r, p_r]) \le \delta$。由此， $d_F(z_r z'_r, \tau_i[q_r, p_r]) \le d_F(y_r x_r, \tau_i[q_r, p_r]) + \max\{d(y_r, z_r)\ d(x_r, z'_r)\} \le (1 + 12\varepsilon)\delta$。由此可得， $d_F(\sigma_0, \tau_i) \le (1 + 12\varepsilon)\delta$。因此，我们的预处理必须存储了一条与输入曲线的弗雷歇距离至多为 $(1 + 12\varepsilon)\delta$ 且满足 σ_0 的曲线，届时将报告该曲线。

F 引理13的证明

考虑 (i)。 $y \in \gamma$ 的存在性意味着 $c \cap xy \neq \emptyset$，因此 $x \in F(c,\ \gamma)$。然后，由于 $x \in L_\xi$，我们得到 $x \in (\mathrm{aff}(\xi) \oplus B_{2\varepsilon\delta}) \cap F(c,\ \gamma)$。此外， ξ 包含 x 的投影，因为 $x \in L_\xi$。因此，由于 ξ 是规范线段，$(\mathrm{aff}(\xi) \oplus B_{2\varepsilon\delta}) \cap F(c,\ \gamma)$ 在 $\mathrm{aff}(\xi)$ 上的投影必定包含 ξ。根据定义，可得 $c \in C_{\gamma,\ \xi}$。

考虑 (ii)。令 x 为 L_ξ 中的任意一点。令 y 为 γ 中的任意一点。根据 ξ 的定义，存在一点 $x' \in L_\xi \cap F(c,\ \gamma)$，使得 $d(x,\ x') \le 4\varepsilon\delta$。由于 $x' \in F(c,\ \gamma)$，存在 $y' \in \gamma$，使得 $x'y' \cap c \neq \emptyset$。我们有 $d(y,\ y') \le \varepsilon\delta$，因为它们都属于 γ；因此，线性插值将 $x'y'$ 中的每一点映射到 $x'y$ 中距离不超过 $\varepsilon\delta$ 的一点。因此，$x'y \cap (c \oplus B_{\varepsilon\delta}) \neq \emptyset$。类似地，由于 $d(x,\ x') \le 4\varepsilon\delta$，我们得到 $xy \cap (c \oplus B_{5\varepsilon\delta}) \neq \emptyset$。这就证明了 (ii)。

反设 (iii) 对 $\lambda = 11\varepsilon\delta$ 不成立。因此存在 $x \in \xi$ 和 $y \in \gamma$，使得当我们从 x 走到 y 时，在到达 $c_{\gamma,\xi} \oplus B_{11\varepsilon\delta}$ 之前，会先经过单元 $c \in C_{\gamma,\xi}$。由此可知，若将 $c_{\gamma,\xi} \oplus B_{11\varepsilon\delta}$ 缩短距离 $\varepsilon\delta$，它就会与 c 不相交。此外，根据 (ii)，$c_{\gamma,\xi} \oplus B_{10\varepsilon\delta}$ 与 xy 相交。也就是说， c 与 $c_{\gamma,\xi} \oplus B_{10\varepsilon\delta}$ 不相交，并且相对于 $\le_{xy}$， $c \cap xy$ 严格先于 $xy \cap (c_{\gamma,\xi} \oplus B_{10\varepsilon\delta})$。 c 与 $c_{\gamma,\xi} \oplus B_{10\varepsilon\delta}$ 不相交，这意味着 $c \oplus B_{5\varepsilon\delta}$ 与 $c_{\gamma,\xi} \oplus B_{5\varepsilon\delta}$ 不相交。根据 (i)， $c \oplus B_{5\varepsilon\delta}$ 和 $c_{\gamma,\xi} \oplus B_{5\varepsilon\delta}$ 均与 xy 相交。因此，相对于 $\le_{xy}$， $xy \cap (c \oplus B_{5\varepsilon\delta})$ 严格先于 $xy \cap (c_{\gamma,\xi} \oplus B_{5\varepsilon\delta})$。

将这条行走的终点从 y 线性移动到 p_γ。根据 (ii)，与 $c \oplus B_{5\varepsilon\delta}$ 和 $c_{\gamma,\xi} \oplus B_{5\varepsilon\delta}$ 的交点在整个移动过程中始终非空。因此， $c \oplus B_{5\varepsilon\delta}$ 和 $c_{\gamma,\xi} \oplus B_{5\varepsilon\delta}$ 的交点顺序在整个移动过程中不会改变。类似地，我们将行走的起点从 x 线性移动到 p_ξ，由此可得， $c \oplus B_{5\varepsilon\delta}$ 和 $c_{\gamma,\xi} \oplus B_{5\varepsilon\delta}$ 的交点顺序在整个移动过程中不会改变。但这样一来，我们就应该更倾向于让 c 为 $c_{\gamma,\xi}$，这就产生了矛盾。

一旦我们证明 (iii) 对 $\lambda = 11\varepsilon\delta$ 成立，那么对于更大的 λ 也成立

因为 $c_{\gamma,\,\xi} \oplus B_\lambda$ 只会变得更大。

G 引理 8 的证明

考虑第 4.2 节中的查询过程。

步骤 2 符合 $(11\varepsilon\delta)$-线段查询和 (κ,δ)-近似最近邻查询的输出要求，因此能够正确执行。接下来需要证明步骤 3 能够返回正确答案。我们首先给出两个技术性结果。第一个结果给出了步骤 3 中 γ 与输入顶点之间距离的上界。

▶ **引理 15.** 步骤 *3* 中的单元 γ 与 仿射包$(w_j w_{j+1})$ 相交，且与某个输入顶点的距离小于 $(1+6\varepsilon)\delta$。因此，$\gamma \in G_3$。

证明。假设当我们从步骤 2 进入步骤 3 时，γ 是 G_3 中计算得到的单元。步骤 1 中由 D_{anp} 返回的网格顶点 x 与 $\text{aff}(w_j w_{j+1})$ 的距离至多为 $(1+\varepsilon)\varepsilon\delta$。因此，$G(x \oplus B_{2\varepsilon\delta})$ 中必有某个单元与 $\text{aff}(w_j w_{j+1})$ 相交。当我们从步骤 2 进入步骤 3 时，必定会得到与 $\text{aff}(w_j w_{j+1})$ 相交的单元 $\gamma \in G(x \oplus B_{2\varepsilon\delta})$。由于 x 是 G_1，的网格顶点，γ 与某个输入顶点之间的距离至多为 $(1+3\varepsilon)\delta$。

另一种可能是，γ 在跳转到步骤 3(a) 或 3(b) 之前，于步骤 3(c) 中得到 $\hat{}\gamma$。为避免混淆，我们用 γ 表示从步骤 2 进入步骤 3 时计算得到的 $\mathcal{G}_3$ 中的单元。根据定义，$c_{\gamma,\,\xi}$ 属于 $\mathcal{G}_1$，这意味着 $c_{\gamma,\,\xi}$ 与某个输入顶点 $v_{i,\,a}$ 的距离不超过 δ。由于 $\hat{}\gamma \in G(c_{\gamma,\,\xi} \oplus B_{5\varepsilon\delta})$，我们有 $d(v_{i,\,a},\ \hat{}\gamma) \le (1+6\varepsilon)\delta$。因此，$\hat{}\gamma \in G(v_{i,\,a} \oplus B_{(1+6\varepsilon)\delta}) \subset \mathcal{G}_3$。◀

第二个结果表明，如果 γ 和 c 不相交，那么 $F(\gamma,\ c)$ 和 $F(c,\ \gamma)$ 也不相交。

▶ **引理 16。** 若 $\gamma \cap c = \emptyset$，则 $F(\gamma,c) \cap F(c,\gamma) = \emptyset$。

证明。反设 $F(\gamma,c) \cap F(c,\gamma) \neq \emptyset$。取一点 $x \in F(\gamma,c) \cap F(c,\gamma)$。根据 $F(c,\gamma)$ 的定义，我们可以找到一点 $y \in \gamma$，使得 $xy \cap c \neq \emptyset$。类似地，根据 $F(\gamma,c)$ 的定义，我们可以找到另一点 $z \in c$，使得 $xz \cap \gamma \neq \emptyset$。取两点 $y' \in xy \cap c$ 和 $z' \in xz \cap \gamma$。注意到 x、y、z、y'、z' 都位于由 xy 和 xz 张成的同一平面内。因此 $yz' \cap y'z \neq \emptyset$。由于 c 是凸的，且 y'、$z \in c$，我们有 $y'z \subset c$。类似地，我们有 $yz' \subset \gamma$。但这样一来，$yz' \cap y'z \subseteq c \cap \gamma$，这与假设 $c \cap \gamma = \emptyset$ 矛盾。◀

现在，我们已经可以证明，查询过程要么发现 $\min_{\tau_i \in T} d(\sigma,\tau_i) > \delta$，要么为（$11\varepsilon\delta$）线段查询返回正确答案。

引理 8 的证明。我们首先说明，如果查询过程返回一个单元，那么该单元满足 $(11\varepsilon\delta)$-线段查询的正确答案要求。单元 c_{ans} 要么在步骤 3(a) 中返回，要么在步骤 3(b) 中返回。如果 c_{ans} 是在步骤 3(a) 中返回的，那么根据构造，$c_{\text{ans}} \oplus B_{7\varepsilon\delta}$ 包含 w_j，因此 c_{ans} 必定是一个正确答案。如果 c_{ans} 是在步骤 3(b) 中返回的，则它等于 $c_{\gamma,\,\xi}$。根据步骤 3 的前提条件，w_j 投影到 ξ 和 $d(w_j,\ \xi) \le 2\varepsilon\delta$。因此，根据引理 13，$c_{\gamma,\,\xi}$ 是 $(11\varepsilon\delta)$-线段查询的正确答案。

还需证明：如果 $\mathcal{G}_1$ 中存在一个与 $w_j w_{j+1}$ 相交的单元，那么查询过程要么会发现 $\min_{\tau_i \in T} d_F(\sigma,\tau_i) > \delta$，要么会返回一个单元。根据假设，$\mathcal{G}_1$ 中包含一个与 $w_j w_{j+1}$ 相交的单元，因此 $\text{aff}(w_j w_{j+1})$ 到 $\mathcal{G}_1$，的某些网格顶点的距离不超过 $\varepsilon\delta$，所以 D_{anp} 必须返回一个网格顶点 x，其与 $\text{aff}(w_j w_{j+1})$ 的距离不超过 $(1+\varepsilon)\varepsilon\delta$。因此，第 2 步不可能返回 null。如果第 2 步对 (κ,δ)-近似最近邻查询的回答是“否”，那么必有 $\min_{\tau_i \in T} d_F(\sigma,\tau_i) > \delta$。剩下的可能性是查询过程继续执行第 3 步。

假设第 3(a) 步适用。在这种情况下，$w_j \in \gamma$，根据引理 15，γ 到某个输入顶点 $v_{i,a}$ 的距离不超过 $(1+6\varepsilon)\delta$。因此，w_j 到 $G(v_{i,a} \oplus B_\delta) \subset \mathcal{G}_1$ 中某个单元的距离不超过 $7\varepsilon\delta$。因此，第 3(a) 步将成功返回 $G(w_j \oplus B_{7\varepsilon\delta})$ 中的一个。

如果步骤 3(b) 适用，则将返回 $c_{\gamma,\xi}$。

假设步骤 3(c) 适用。令 ξ 为 ℓ_{j+1} 中包含 w_{j+1} 在 ℓ_{j+1} 中投影的规范线段。根据引理 13(ii)，$c_{\gamma,\xi} \oplus B_{5\varepsilon\delta}$ 与 $w_{j+1}w_j$ 相交。由于 $c_{\gamma,\xi}$ 属于 $\mathcal{G}_1$，，它与某个输入顶点 $v_{i,a}$ 的距离不超过 δ，这意味着 $c_{\gamma,\xi} \oplus B_{5\varepsilon\delta}$ 被 $G(v_{i,a} \oplus B_{(1+6\varepsilon)\delta}) \subset \mathcal{G}_3$ 中的单元覆盖。因此，步骤 3(c) 将成功找到与 $w_j w_{j+1}$ 相交的单元 $\hat{\gamma} \in \mathcal{G}_3$。所以，步骤 3(a) 或 3(b) 对 $\hat{\gamma}$ 适用，查询过程将返回一个单元。◀