

计算最短穿越

于海

一、直线穿越的表示

给定 n 维空间中的 m 个非空点集 S_1, \dots, S_m , 如果直线 L 与每个点集均有公共点, 则称 L 为这 m 个点集的一条直线穿越 (line transversal 或 stabbing line)。同样, 如果线段 K (本文中的线段均指闭线段, 且包括单点集) 与每个点集均有公共点, 则称 K 为这 m 个点集的一条线段穿越 (segment transversal 或 stabbing line segment)。穿越的概念是海莱定理 (Helly's theorem) 的延伸, 它的存在性等问题一直是计算几何中海莱类定理 (Helly-type theorems) 的理论 ([1]、[2]) 和算法 ([3]、[4]) 研究的热点。

在本文中, 我们主要讨论最短线段穿越的问题。

引理 1.1 S_1, \dots, S_m 为非空点集, L 为它们的一条直线穿越, 定义集合 $E = \{ \text{线段 } r \mid r \subset L, \text{ 且 } r \text{ 为线段穿越} \}$, 则 $E \neq \emptyset$ 。

证明 $\forall i=1, \dots, m$, 因 $L \cap S_i \neq \emptyset$, 可取 $u_i \in L \cap S_i$, 在 L 上取一条包含所有 u_i 的线段 t , 则有 $t \in E$, 从而 $E \neq \emptyset$ 。 \square

在引理 1.1 的条件下, 因 $E \neq \emptyset$, 可记 $K = \bigcap_{r \in E} r$ 。易知: 当 $\bigcap_{i=1}^m S_i$ 为单点集 $\{a\}$ 时, $K = \{a\}$; 当 $\bigcap_{i=1}^m S_i$ 为多点集时, $K = \emptyset$ 。下面考察 S_1, \dots, S_m 为非空凸闭点集且 $\bigcap_{i=1}^m S_i = \emptyset$ 的情况。

定理 1.1 S_1, \dots, S_m ($m \geq 2$) 为非空凸闭点集且 $\bigcap_{i=1}^m S_i = \emptyset$, L 为它们的一条直线穿越, 则 \exists 唯一线段穿越 $K \subset L$, 对 \forall 线段穿越 $t \subset L$, 有 $K \subset t$ 。

证明 首先证明存在性。记 L 的直线方程为 $y = ax + b$, 其中 a, b 为 n 维向量。 $\forall i=1, \dots, m$, $L \cap S_i$ 为凸闭点集, 从而为线段或射线或直线, 则 $\exists p_i, q_i \in \mathbb{R} \cup \{-\infty, +\infty\}$, $p_i \leq q_i$, t_i 为 L 上由 $x \in [p_i, q_i]$ 定义的部分, 且有 $L \cap S_i = t_i$ 。记 $p = \max\{p_1, \dots, p_m\}$, $q = \min\{q_1, \dots, q_m\}$, 则有: (1) $p \neq +\infty, q \neq -\infty$, 这是显然的; (2) $p \neq -\infty$, 否则便有 $p_1 = \dots = p_m = -\infty$, 从而 S_1, \dots, S_m 均包含 L 上由 $x \in [-\infty, q]$ 定义的部分, 这与 $\bigcap_{i=1}^m S_i = \emptyset$ 矛盾; (3) $q \neq +\infty$, 理由类似(2); (4) $q < p$, 否则, S_1, \dots, S_m 均包含 L 上由 $x \in [p, q]$ 定义的部分, 也与 $\bigcap_{i=1}^m S_i = \emptyset$ 矛盾。记 K 为 L 上由 $x \in [q, p]$ 定义的部分, 则易验证: (1) $K \cap t_i \neq \emptyset$, 从而 K 为一条线段穿越, 即 $K \in E$; (2) $\forall t \in E$, 设 t 为 L 上由 $x \in [f, g]$ 定义的部分, 因为 $t \cap t_i \neq \emptyset$ ($i=1, \dots, m$), 则有 $g \geq p, f \leq q$, 从而有 $K \subset t$ 。因此, K 即为所求线段。

下证唯一性。假设 K_1, K_2 是满足上述条件的两条线段穿越, 则有 $K_1 \subset K_2$ 且 $K_2 \subset K_1$, 亦即 $K_1 = K_2$ 。证毕。 \square

定义 1.1 处于一般位置的一组直线(**lines in general position**)指无三线共点、无两线平行的一组直线。

定义 1.2 **简单集(simple objects)**指存储空间只需 $O(1)$, 相互之间的求交、求公切线等基本操作也只需 $O(1)$ 时间的点集。

由定理 1.1 可知, 在 S_1, \dots, S_m 为非空凸点集且无公共点的情况下, 点集的直线穿越和该直线上的最短线段穿越是一一对应的。特殊的, 各个点集为如下三种最常被研究的情况:

(1) 处于一般位置的一组直线 (2) 无公共点的等半径圆 (3) 无公共点的全等凸多边形时, 这种一一对应关系均是成立的。因此, 为了得到线段穿越的描述, 我们只需得到直线穿越的描述即可。由定理 1.1 的证明过程可以看出, 如果这些点集为简单集, 则我们可以在 $O(m)$ 时间内找到特定直线穿越的相应最短线段穿越。

1、圆的直线穿越

[3]中采用了模式识别中常用的 Hough 变换来解决圆的直线穿越问题。如图 1.1, Hough 变换将把直线 L 映射为平面上的点 (ρ, θ) 。

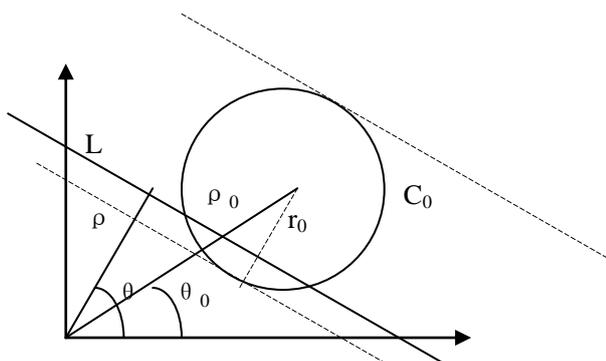


图 1.1 Hough 变换

直线 L 穿越圆 C_0 当且仅当下式成立:

$$\rho_0 \cos(\theta - \theta_0) - r_0 \leq \rho \leq \rho_0 \cos(\theta - \theta_0) + r_0$$

从而, 下式定义的点集实际上给出了平面中 n 个圆的所有直线穿越:

$$\begin{aligned} \max_{1 \leq i \leq n} f_i(\theta) &\leq \rho \leq \min_{1 \leq i \leq n} g_i(\theta), \\ f_i(\theta) &= \rho_i \cos(\theta - \theta_i) - r_i \\ g_i(\theta) &= \rho_i \cos(\theta - \theta_i) + r_i \end{aligned}$$

其中, $\max_{1 \leq i \leq n} f_i(\theta)$ 和 $\min_{1 \leq i \leq n} g_i(\theta)$ 均可以在 $O(n \log n)$ 时间内求出 (参见[3])。

2、线段的直线穿越

在[5]中, Edelsbrunner 等人采用了下述变换 T :

$$\begin{aligned} T: \quad p = (a, b) &\Rightarrow T_p: y = ax + b \\ l: y = kx + d &\Rightarrow T_l = (-k, d) \end{aligned}$$

可以看出，该变换将一条直线映为一点，将一点映为一条直线。同时，它还能保持点与直线之间的上下关系。

引理 1.2 p 与 l 的上下关系经 T 变换后仍然保持。

证明 不妨设 p 在 l 上方，从而有 $b > ka + d$ ，亦即 $d < a(-k) + b$ ，这便意味着 T_p 在 T_l 的上方。 \square

引理 1.3 所有过 (a,b) 点且不垂直的直线 l 对应的 T_l 组成直线 $y=ax+b$ 。

证明 设 l 的直线方程为 $y-b=k(x-a), k \in (-\infty, +\infty)$ ，从而 $T_l = (-k, b-ka)$ ，显然这些点组成直线 $y=ax+b$ 。 \square

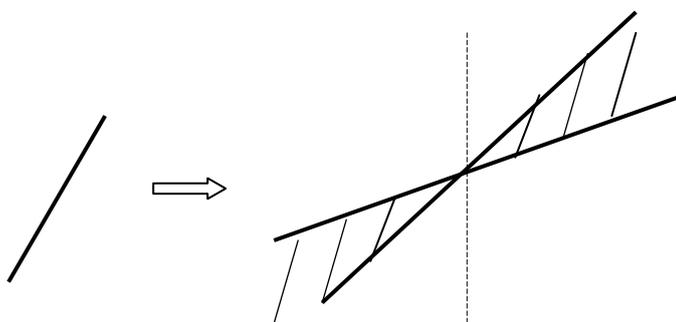


图 1.2 线段及其对应的双楔

在变换 T 的作用下，一条非垂直线段的两个端点将被映射为两条相交直线，它们所夹的两个不含垂直线的区域称为双楔 (double wedge)。由引理 1.2 和引理 1.3 知，双楔中的点所对应的直线正是所有穿越该线段的直线，因此， n 条线段所对应的 n 个双楔相交的区域中的点便对应着这些线段的所有直线穿越。该区域的边数为 $O(n)$ ，整个区域可以在 $O(n \log n)$ 时间内求出 (参见[5])。

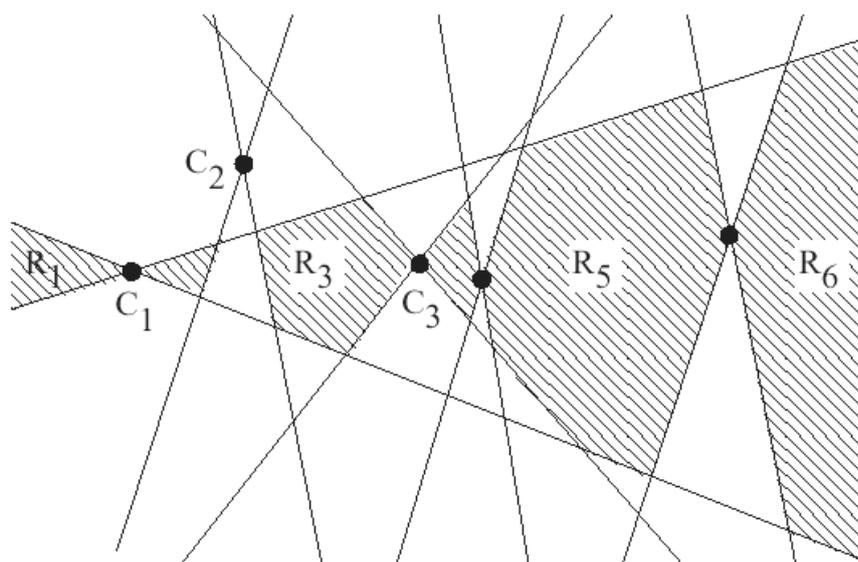


图 1.3 双楔求交所得的区域

3、直线的线段穿越

对任意非垂直的直线 L , 设其方程为 $y=ax+b$, 记

$$L^- = \{(x, y) \mid y \leq ax + b\}$$

$$L^+ = \{(x, y) \mid y \geq ax + b\}$$

称 L^- 为该直线的下半平面, L^+ 为该直线的上半平面。

以下考虑处于一般位置的 n 条直线 L_1, \dots, L_n , 其中无垂直线, 且它们的斜率 k_1, \dots, k_n 依次递增。定义如果 $a < b$, 则 $[a, b] = \{t \mid a \leq t \leq b\}$, 如果 $a > b$, 则 $[a, b] = \{t \mid t \leq b \cup t \geq a\}$,

记 $r_i = [k_i, k_{i+1}]$ ($i=1, \dots, n$, 设 $k_{n+1}=k_1$)。令

$$H_i^L = \{L_1^-, \dots, L_i^-, L_{i+1}^+, \dots, L_n^+\}$$

$$H_i^R = \{L_1^+, \dots, L_i^+, L_{i+1}^-, \dots, L_n^-\}$$

再记 A_i^L 为 H_i^L 中各半平面的交, A_i^R 为 H_i^R 中各半平面的交 (参见图 1.4)。[18]中给出了如下引理:

引理 1.4 $\forall i=1, \dots, n$, 若线段 s 具有斜率 $k \in r_i$, 则 s 是 L_1, \dots, L_n 的线段穿越当且仅当它的两端点分别落在 A_i^L 和 A_i^R 中。

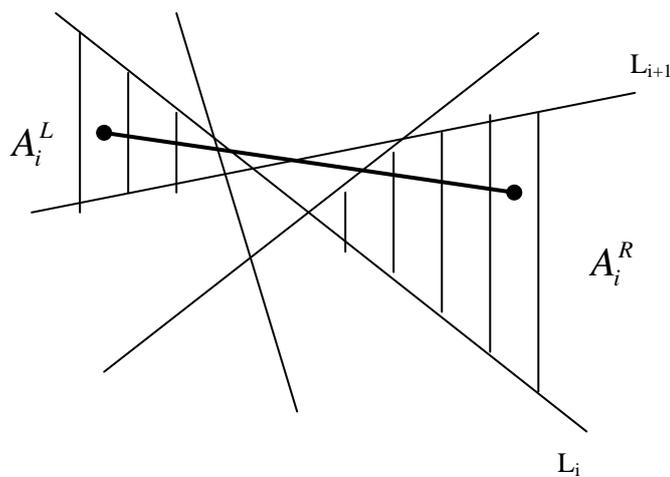


图 1.4 直线的线段穿越表示

二、主要算法的设计

为简化起见, 以下的讨论只针对处于一般位置的一组直线。

定义 2.1 直线的上包迹 (upper envelope) 和下包迹 (lower envelope) 统称为**正规包迹 (normal envelope)**

定义 2.2 设包迹 E_1 由直线 L_{11}, \dots, L_{1m} 各自决定的某半平面交获得, 包迹 E_2 由直线 H_{21}, \dots, H_{2n} 各自决定的某半平面交获得。若 L_{11}, \dots, L_{1m} 的斜率均小于 H_{21}, \dots, H_{2n} 的斜率, 则称 E_1 和 E_2 是**两个分离的包迹 (two separate envelopes)**, 并记为 $E_1 < E_2$; 若 L_{11}, \dots, L_{1m} 的斜率均小于一实数 k , 则记为 $E_1 < k$

引理 2.1 正规包迹 E_1, E_2 满足 $E_1 < E_2$, 则它们有且仅有一个交点

证明 易知正规包迹可以表示为单值连续函数 (这实际上是正规包迹的充要条件), 设两正规包迹各自对应的连续函数为 $F_1(x)$ 和 $F_2(x)$ 。由两包迹分离可知, $\exists k_1, k_2 \in R$ 满足

$E_1 < k_1 < k_2 < E_2$ 。取定 $x_0 \in R$, 令 $a = \frac{|F_2(x_0) - F_1(x_0)|}{k_2 - k_1}$, 从而

$$\begin{aligned} F_1(x_0 - a) &\geq F_1(x_0) - ak_1 \geq F_2(x_0) - ak_2 \geq F_2(x_0 - a) \\ F_1(x_0 + a) &\leq F_1(x_0) + ak_1 \leq F_2(x_0) + ak_2 \leq F_2(x_0 + a) \end{aligned}$$

由连续性知 $F_1(x)$ 和 $F_2(x)$ 必在区间 $[x_0 - a, x_0 + a]$ 内有一交点。下证交点的唯一性。否则设在 x_1, x_2 ($x_1 < x_2$) 处有两个不同的交点, 则

$$k_1 > \frac{F_1(x_2) - F_1(x_1)}{x_2 - x_1}, k_2 < \frac{F_2(x_2) - F_2(x_1)}{x_2 - x_1}$$

这与 $k_1 < k_2$ 矛盾。 \square

引理 2.2 H_i^L, H_i^R 是无限凸集, 且它们各自的两条无限边分别在直线 L_i, L_{i+1} 上 (当 $i=n$ 时, L_{i+1} 指 L_1)

证明 具体证明留给读者。 \square

由引理 2.2 知 A_i^L 和 A_i^R 是相离的 (最多只有一个公共点), 因为它们分别位于直线 L_i 和 L_{i+1} 交点的两边。从而再由引理 1.4 知, 斜率范围为 r_i 的所有线段穿越中最短者必定连接着凸集 A_i^L 和 A_i^R 的边界。根据这个结论, 我们可以很容易的说明[17]中给出的一条引理 ([17] 中给出的证明较为复杂)。

定理 2.1 n 条直线形成的凸壳分别截这 n 条直线得到 n 条线段, 则这 n 条线段的最短穿越也就是原来 n 条直线的最短穿越

根据上述讨论, 我们只需计算出 n 个斜率范围内各自的最短穿越线段, 便可最终得到所有穿越线段中的最短者。这里涉及到两个主要算法, 我们将分别说明。

在下面的算法中, 正规包迹均使用它的各边右端的顶点从左到右形成的序列来表示, 需要注意的是, 为了算法叙述的方便, 我们认为包迹最右边射线的右顶点 (称为无穷顶点) 也存在。这些顶点和包迹的边之间存在着——对应。另外, 我们规定正规包迹各边的方向为从左到右沿着该边的方向。

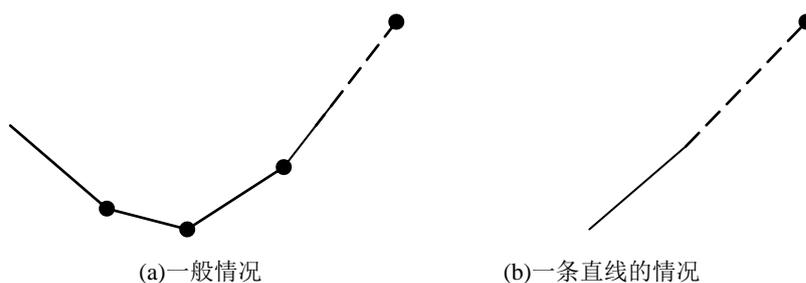


图 2.1 包迹的顶点表示, 注意最后一个顶点

(1) 计算 A_i^L 和 A_i^R 的边界 ∂A_i^L 和 ∂A_i^R

由引理 2.1 和引理 2.2, 合并两个分离的上包迹或者合并分离的上包迹与下包迹最关键的便是找出两包迹之间的唯一交点, 根据这个思想, 我们可以设计出有效的包迹合并算法。

算法 2.1 合并两个分离的上包迹

设 $E_1 < E_2$, 记 E_1 的顶点序列为 $\{V_1[i] \mid i=0, \dots, m-1\}$, 它们相应的边所在直线的序列为 $\{L_1[i] \mid i=0, \dots, m-1\}$; E_2 的顶点序列为 $\{V_2[j] \mid j=0, \dots, n-1\}$, 它们相应的边所在直线的序列为 $\{L_2[j] \mid j=0, \dots, n-1\}$

- (1) $x \leftarrow 0, y \leftarrow 0, V \leftarrow \emptyset, L \leftarrow \emptyset$
- (2) WHILE ($y < n-1$ && $V_2[y]$ 不在直线 $L_1[x]$ 上方)
 $y \leftarrow y+1$
- (3) WHILE ($x < m-1$ && $V_1[x]$ 在直线 $L_2[y]$ 上方)
 $V \leftarrow V \cup \{V_1[x]\}, L \leftarrow L \cup \{L_1[x]\}, x \leftarrow x+1$
- (4) IF ($V_2[y]$ 在直线 $L_1[x]$ 上方)
 GOTO (6)
- (5) GOTO (2)
- (6) 求出 $L_1[x]$ 和 $L_2[y]$ 的交点 $I, V \leftarrow V \cup \{I\}, L \leftarrow L \cup \{L_1[x]\}$
- (7) WHILE ($y < n$)
 $V \leftarrow V \cup \{V_2[y]\}, L \leftarrow L \cup \{L_2[y]\}, y \leftarrow y+1$

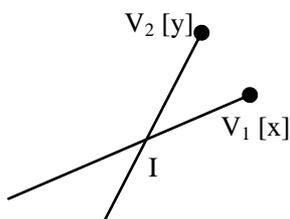


图 2.2 算法 2.1 跳至 (6) 时的情况

定理 2.2 算法 2.1 返回的 V, L 分别是 E_1, E_2 合并后的包迹 E 的顶点序列和相应的边所在直线的序列; 整个算法具有复杂度 $O(m+n)$

证明 引理 2.1 保证了经过有限步后必可从 (4) 跳到 (6), 我们只需证明 (6) 中求得的 I 即是 E_1, E_2 唯一的交点。跳到 (6) 时, 可以验证, $V_1[x]$ 在直线 $L_2[y]$ 上或其下方, $V_2[y]$ 在直线 $L_1[x]$ 上方。同时, $V_1[x-1]$ (若存在) 必在 $L_2[y]$ 上方 (否则算法便会在 $x-1$ 时跳到

(6)), 而 $V_2[y-1]$ (若存在) 必在 $L_1[x]$ 上或其下方 (否则算法便会在 $y-1$ 时跳到 (6))。从而包迹上这两点各自对应的两段必相交, 它们的交点便是两包迹唯一的交点。

显然, (2)、(3) 对 E_1 、 E_2 的每个顶点至多检查一次, 故而复杂度为 $O(m+n)$, 另外, (4)、(5) 最多被执行 $m+n$ 次, (7) 具有复杂度 $O(n)$, 其他语句复杂度为 $O(1)$, 从而整个算法的复杂度为 $O(m+n)$ 。□

定义 2.3 顺时针计 α 到 β 夹角为 θ , 则向量 α 到 β 的角度 η 定义为: 若 $0 \leq \theta \leq \pi$, 则 $\eta = \theta$, 否则 $\eta = 2\pi - \theta$

算法 2.2 合并分离的上包迹和下包迹

记 E_1 的顶点序列为 $\{V_1[i] \mid i=0, \dots, m-1\}$, 它们相应的边所在直线的序列为 $\{L_1[i] \mid i=0, \dots, m-1\}$, 相应的边所对应的方向向量序列为 $\{D_1[i] \mid i=0, \dots, m-1\}$; E_2 的顶点序列为 $\{V_2[j] \mid j=0, \dots, n-1\}$, 它们相应的边所在直线的序列为 $\{L_2[i] \mid i=0, \dots, n-1\}$, 相应的边所对应的方向向量序列为 $\{D_2[i] \mid i=0, \dots, n-1\}$

(1) $x_0 \leftarrow 0, y_0 \leftarrow m-1, x_1 \leftarrow 0, y_1 \leftarrow n-1$

(2) IF ($x_0 \neq y_0 \parallel x_1 \neq y_1$)

GOTO (8)

(3) $x \leftarrow \left\lfloor \frac{x_0 + y_0}{2} \right\rfloor, y \leftarrow \left\lfloor \frac{x_1 + y_1}{2} \right\rfloor$

(4) 以下 (4) 至 (7) 不妨设 $V_1[x]$ 在 $V_2[y]$ 的左侧 (即 $V_1[x]$ 的 x 轴坐标 $\leq V_2[y]$ 的 x 轴坐标), $\alpha \leftarrow D_1[x+1]$ 到 $V_1[x]V_2[y]$ 的角度, $\beta \leftarrow D_2[y]$ 到 $V_1[x]V_2[y]$ 的角度

(5) IF (E_1 是下包迹 && E_2 是上包迹)

$\alpha \leftarrow -\alpha, \beta \leftarrow -\beta$

(6) IF ($\alpha \geq 0$) {

IF ($\beta \geq 0$)

$y_1 \leftarrow y$;

ELSE IF ($|\beta| \leq \alpha$)

$y_0 \leftarrow x, y_1 \leftarrow y$;

ELSE

$x_0 \leftarrow x+1, x_1 \leftarrow y+1$;

}

ELSE {

IF ($\beta \leq 0$)

$x_0 \leftarrow x+1$;

ELSE IF ($|\beta| \leq \alpha$)

$y_1 \leftarrow y$;

ELSE

$x_0 \leftarrow x+1$;

}

(7) GOTO (2)

(8) 以下 (8) 至 (10) 不妨设 E_1 是上包迹, E_2 是下包迹。此时, $x_0=y_0$ 或 $x_1=y_1$ 。

这里设 $x_0=y_0$, 对 $x_1=y_1$ 的情况请读者根据算法的主要思想自行给出

```

WHILE ( $x_1 \neq y_1$ ) {
     $y \leftarrow \lfloor \frac{x_1 + y_1}{2} \rfloor$  ;
    IF ( $E_1 < E_2$ ) {
        IF ( $V_2[y]$ 在  $L_1[x_0]$ 下方)
             $x_1 \leftarrow y + 1$  ;
        ELSE
             $y_1 \leftarrow y$  ;
    }
    ELSE {
        IF ( $V_2[y]$ 在  $L_1[x_0]$ 上方)
             $x_1 \leftarrow y + 1$  ;
        ELSE
             $y_1 \leftarrow y$  ;
    }
}
    
```

(9) 计算 $L_1[x_0]$ 和 $L_2[x_1]$ 的交点 I 。

(10) IF ($E_1 < E_2$)

将 E_1 在 I 点右边的部分和 E_2 在 I 点右边的部分合并得到包迹 E ;

ELSE

将 E_1 在 I 点左边的部分和 E_2 在 I 点左边的部分合并得到包迹 E ;

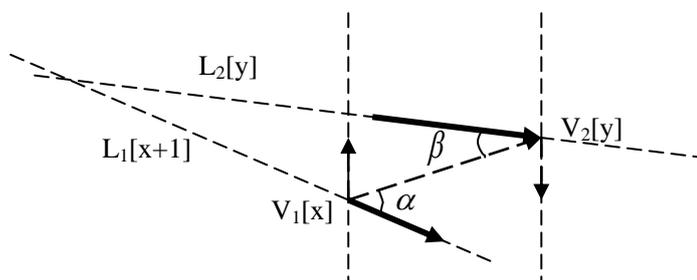


图 2.3 算法 2.2 中的对半消除

定理 2.3 算法 2.2 能找出分离的上包迹和下包迹合并后的包迹; 在第 (10) 步复杂度为 $O(1)$ 的条件下整个算法具有复杂度 $O(\log m + \log n)$

证明 算法的关键步骤是第 (6) 步的对半消除, 它每次淘汰某一包迹上不可能与另一包迹相交的部分。我们只针对其中的某一情况给出证明, 其余的情况留给读者验证。如图 2.3, 考虑 $\alpha < 0, 0 < \beta \leq |\alpha|$ 的情况。此时, 下包迹在直线 $L_2[y]$ 下方, 而上包迹在直线 $L_1[x+1]$

上方, 由连续性, $V_2[y]$ 左边的部分必定与上包迹相交。因此, $V_2[y]$ 右边的部分必不可能与上包迹相交 (因为两包迹有且仅有一个交点), 完全可以舍去, 这也正是 $y_1 \leftarrow y$ 所做的。

第 (5) 步实际上利用关于 x 轴的对称操作将 (E_1 是上包迹 && E_2 是下包迹) 转化为 (E_1 是下包迹 && E_2 是上包迹)。第 (8) 步是在已经得到交点 I 在某一包迹上所在的边后, 对另一包迹进行对半查找。第 (10) 步的正确性由引理 2.2 保证。

由于 (6)、(8) 实现了对半消除 (即每次都有一半的包迹顶点被淘汰), 从而在第 (10)

步复杂度为 $O(1)$ 的条件下整个算法具有复杂度 $O(\log m + \log n)$ 。 \square

如果采用链表结构来表示包迹，第 (10) 步完全可以在 $O(1)$ 时间内实现。

由算法 2.1，我们可以很容易的结合分治算法实现计算直线的上包迹，复杂度满足 $T(n) = 2T(n/2) + O(n)$ ，从而 $T(n) = O(n \log n)$ 。通过关于 x 轴的对称变换，直线的下包迹可以转化为求解上包迹。最后，结合算法 2.2，我们便可以求出各 ∂A_i^L 和 ∂A_i^R 。

(2) 计算 ∂A_i^L 和 ∂A_i^R 之间的最短距离

在[9]中，Edelsbrunner 已经给出了求解两凸多边形最短距离的复杂度为 $O(\log m + \log n)$ 的算法。该算法分为三步：(1) 初始化阶段 (Initial Phase)，(2) 对半消除阶段 (Binary Elimination) (3) 收尾阶段 (Final Phase)。在我们所关心的问题中，根据引理 2.2，第一步是不需要执行的；第二步中，他的算法对可能出现的各种情况都做了相应的讨论和处理，但是，在我们的问题中，有很多情况是根本不会出现的，从而完全可以对该步算法的相应部分作适当的删除和简化。

引理 2.3 s 是连接 ∂A_i^L 和 ∂A_i^R 各自某一顶点的线段，则[9]中定义的 $\alpha', \alpha'', \beta', \beta''$ 均在 $(0, \pi)$ 范围内。

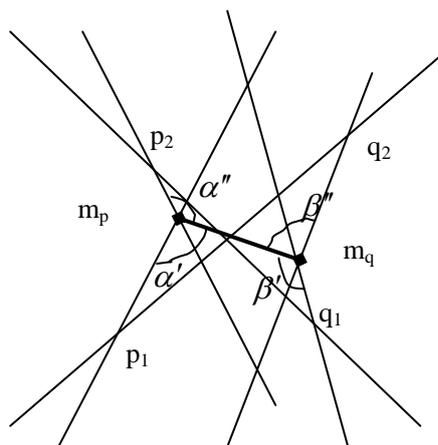


图 2.4 对 Edelsbrunner 算法的简化

根据上述引理，我们得到如下简化了的算法：

算法 2.3 简化对半消除 (Simplified Binary Elimination)

情况 1: 如果某一条链仅有一个顶点 (不妨设 $p_1=p_2$)。如果 $\beta' \geq \pi/2$ ，则令 $q_1=m_q$ ；如果 $\beta'' \geq \pi/2$ ，则令 $q_2=m_q$

情况 2: 如果某一条链仅有两个顶点 (不妨设 $p_2=m_p$)。如果 $\alpha' + \beta' > \pi$ ，则: 若 $\alpha' \geq \pi/2$ 便令 $p_1=p_2$ ，若 $\beta' \geq \pi/2$ 便令 $q_1=m_q$ ；如果 $\beta'' \geq \pi/2$ ，则令 $q_2=m_q$ ；如果 $\alpha' < \beta'' < \pi/2$ ，则: 若 m_q 在线段 p_1p_2 上的投影存在便令 $q_2=m_q$ ，否则便令 $p_2=p_1$

情况 3: 每条链都至少含三个顶点。如果 $\alpha' + \beta' > \pi$, 则: 若 $\alpha' \geq \pi/2$ 便令 $p_1=m_q$, 若 $\beta' \geq \pi/2$ 便令 $q_1=m_q$; 如果 $\alpha'' + \beta'' > \pi$, 则: 若 $\alpha'' \geq \pi/2$ 便令 $p_2=m_p$, 若 $\beta'' \geq \pi/2$ 便令 $q_2=m_q$

定理 2.4 算法 2.3 具有复杂度 $O(\log m + \log n)$

三、关于最低复杂度

在本节里, 我们将主要对一类问题的算法最低复杂度 (lower bounds) 作一总结, 并证明求 n 条直线上包迹问题的最低复杂度为 $\Omega(n \log n)$ 以及合并两个分离的上包迹的最低复杂度为 $\Omega(n)$, 从而说明了在第二节中给出的相应的算法是最优的。证明一个问题的最低复杂度一般有两种方法: 一种方法是给出某一实例, 证明求解该实例所需的复杂度, [9]中求解两多边形的最长距离、[10]中求解两多边形的最短顶点距离均采用了这样的方法; 另一种方法是, 已知问题 A 的最低复杂度, 同时问题 A 又能在线性时间内转化为问题 B, 则问题 B 至少具有和问题 A 同样的最低复杂度, 这一方法在[14]、[15]中被多次使用。我们将着力突出后一种, 即问题之间的相互转化。

问题 1 元素互异问题 (element uniqueness problem): 判断含 n 个实数的集合中各元素是否互异

问题 2 最小间距问题 (minimum gap problem): 求出实轴上 n 个点之间的最小间距

问题 3 求出由 n 条直线决定的凸壳

问题 4 求出由 n 条直线决定的包迹

问题 5 宽度问题 (width problem): 判断平面上由 n 个点组成的某个点集的宽度是否小于等于某一给定的值 w

问题 6 圆的穿越问题 (stabbing circles problem): 判断平面上 n 个等半径圆是否存在一条直线穿越

问题 7 最大间距问题 (max gap problem): 给定实轴上 n 点以及实数 d , 判断是否存在相邻两点, 它们之间的距离大于等于 d

问题 8 圆弧覆盖问题 (arc cover problem): 给定同一圆周上的 n 条开圆弧, 判断它们的并是否覆盖了整个圆周

问题 9 圆弧相交问题 (arc intersection problem): 给定同一圆周上的 n 条闭圆弧, 判断它们的交是否非空

问题 10 线段的穿越问题 (stabbing segments problem): 给定平面上的 n 条线段, 判断是否存在它们的一条直线穿越

问题 11 求出平面上 n 条直线的上包迹

问题 12 求出平面上 n 点的凸壳

问题 13 求出平面上 n 条线段的上包迹

引入记号 $A \rightarrow B$ 表示问题 A 可以在线性时间内转化为问题 B。在[14]中, Ching 和 Lee 证明了如下转化关系: (1) \rightarrow (2) \rightarrow (3); 在[15]中, D. Avis 和 J. M. Robert 则证明了如下转化关系: (5) \rightarrow (6), (7) \rightarrow (8) \rightarrow (9) \rightarrow (10), 其中, (5) \rightarrow (6), (8) \rightarrow (9)的转化是可逆的, 从而也有(6) \rightarrow (5), (9) \rightarrow (8)。另外, 由[19]中介绍的格雷厄姆方法可知(3) \rightarrow (4)。

定义 3.1 设 e 是凸多边形 P 上的一条边, v 是 P 的所有顶点中距离 e 所在直线最远的顶点, 则称 (e, v) 为一组对跖边点对 (antipodal edge-vertex pair), v 到 e 所在直线的距离称为该对跖边点对的宽度。

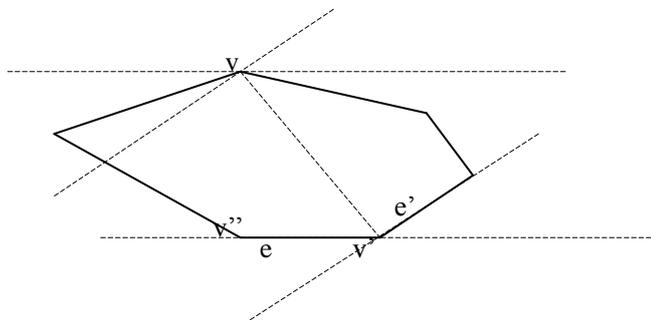


图 3.1 对跖边点对

引理 3.1 (1) 凸 n 多边形 n 条边决定的 n 组对跖边点对可在 $O(n)$ 时间内全部找到。

(2) 凸 n 多边形的宽度等于它的 n 条边决定的 n 组对跖边点对宽度中最小者。

证明 (1) 如图 3.1, 取定一条边 e , 离它最远的顶点 v 可在 $O(n)$ 时间内找到, 它们构成一组对跖边点对。设 e' 是 e 逆时针方向上的下一条边, 则 v' 逆时针到 v 这条链上所有的顶点必都处在过 v 且与 e' 平行的直线和 e' 之间 (因为由凸性, 这条链夹在过 v 与 e 平行的直线和 vv' 之间), 从而它们到 e' 的距离均不超过 v 到 e' 的距离, 所以离 e' 最远的顶点必可在 v 逆时针到 v' 这条链上找到。综上可知, 在找到一对对跖边点对后, 对凸多边形边界搜索一周便可找出所有的对跖边点对。

(2) 略。 □

易知 n 点点集的宽度等于该点集凸壳的宽度, 因此由引理 3.1 可知, 当找到了该点集的凸壳后, 其宽度便可在线性时间内求出, 从而得出结论: (5) \rightarrow (12)。

下面说明(11)和(12)之间的相互转化关系。我们需要使用第一节中提到的变换 T 。

定义 3.2 设凸多边形最左边的顶点为 P_1 (若同时有多个点均具有最小的 x 轴坐标, 则取处于最上方的点), 最右边的顶点为 P_2 (若同时有多个点均具有最大的 x 轴坐标, 则取处于最上方的点), 称 P_1 与 P_2 所连接的凸多边形上方的边界为该凸多边形的上盖 (roof)。

考虑 L_1, \dots, L_n 这 n 条直线 (不失一般性, 可设其中无垂直线), 在变换 T 的作用下, 它们分别映射为点 P_1, \dots, P_n 。设 L_1, \dots, L_n 的上包迹顶点从左到右依次记为 $V_i (i=1, \dots, r)$, 而 P_1, \dots, P_n 的凸壳上盖的顶点从右到左记为 $K_i (i=1, \dots, t+1)$, 边从右到左记为 $E_i (i=1, \dots, t)$, 它们所在的直线分别记为 $H_i (i=1, \dots, t)$ 。

引理 3.2 $r=t$, 且 $T(V_i)=H_i (i=1, \dots, t)$

证明 我们使用数学归纳法。当 $n=2$ 时, 因为 V_1 在 L_1, L_2 上, 由引理 1.2, $P_1=K_1, P_2=K_2$ 在直线 $T(V_1)$ 上, 即 $T(V_1)=H_1$ 。设 $n=k$ 时结论仍成立, 现考虑 $n=k+1$ 的情况。

不妨设 V_1 是直线 L_1 和 L_2 的交点, 其余通过 V_1 的直线斜率均在 L_1 和 L_2 之间。由引理 2.2, 这两条直线中有一条具有所有直线中最小的斜率, 设为 L_1 , 又注意到 L_1 是同斜率的直线中截距最大者, 从而 $P_1=K_1$, 下证 $P_2=K_2$ 。否则, 设 P_1P_2 所在的直线为 L , 易知 $T(V_1)=L$, 有两种情况: 若在 L 上方有一点 P_1 , 则 L_1 将在点 V_1 上方, 这与 V_1 是上包迹的一个顶点矛盾; 若在 P_1P_2 的延长线上有一点 P_1 , 则 L_1 同样通过点 V_1 且斜率比 L_2 大, 这与 L_2 的选取矛盾。

盾。从而我们有 $P_2=K_2$ 。再由引理 1.2, 便有 $T(V_1)=H_1$ 。

下面, 我们将所有斜率小于 L_2 斜率的直线去掉, 此时 $V_i(i=2, \dots, k+1)$ 将成为剩余直线的上包迹顶点, 而 P_2 将成为凸壳上盖最右边的顶点 (因为变换域中相应的去掉的点正是所有 P_2 右边的点)。由归纳法, 可知 $r-1=t-1$, 且 $T(V_i)=H_i (i=2, \dots, k+1)$ 。综上, 便知 $r=t$, 且 $T(V_i)=H_i (i=1, \dots, k+1)$, 从而原引理成立。 \square

对于 L_1, \dots, L_n 这 n 条直线, 首先通过变换 T 在 $O(n)$ 内将它们映射成 n 个点 P_1, \dots, P_n ; 计算出这 n 个点的凸壳后, 我们便可以在 $O(n)$ 内找到它的上盖; 根据所得的上盖我们便可以通过 T 的逆变换得到原来直线上包迹的各个顶点, 再由引理 2.2 便可在 $O(1)$ 内得到完整的上包迹。因此, 我们有(11) \rightarrow (12)。

仿照上面的讨论, 我们也可以很容易的得到(12) \rightarrow (11), 从而便有(11) \leftrightarrow (12)。

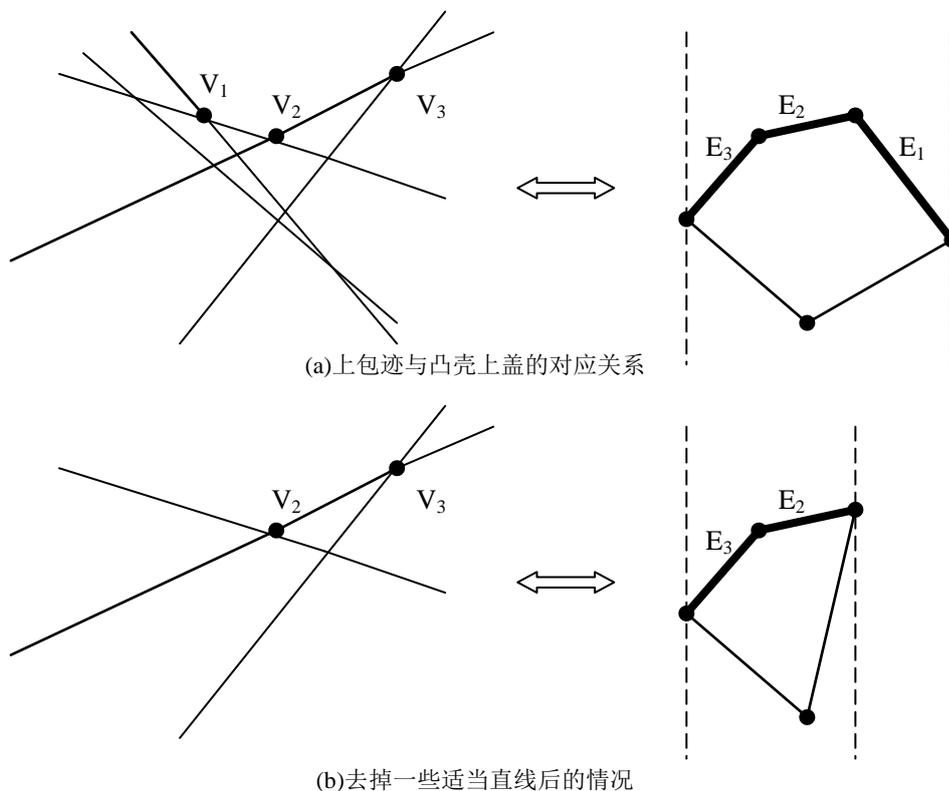


图 3.2 问题(11)与问题(12)的相互转化

最后我们说明(11) \rightarrow (13)。不妨设这 n 条直线不相互平行。在所有直线中, 若有垂直线, 我们可将所有线段旋转适当角度, 使各条直线均不垂直。然后, 找出斜率最小和最大的两条直线 l_1 和 l_2 (如果斜率一样, 取截距较大者, 即处于上方的直线)。求出 l_1 上与其他直线相交的所有交点中最左边的交点, 设它的 x 轴坐标为 a , 再求出 l_2 上与其他直线相交的所有交点中最右边的交点, 设它的 x 轴坐标为 b , 注意到 l_1 必和 l_2 相交, 从而有 $a \leq b$ 。最后计算出所有直线在 $[a, b]$ 范围内的线段部分。以上步骤均能在 $O(n)$ 内完成, 而这些线段的上包迹实际上是直线上包迹除去最外两条在 l_1, l_2 上的射线 (参见图 3.3), 从而在得到线段的上包迹后能在 $O(1)$ 内得到原先直线的上包迹。

总结上述的转化关系, 我们得到如下三条转化链:

(1) \rightarrow (2) \rightarrow (3) \rightarrow (4)

(6) \leftrightarrow (5) \rightarrow (12) \leftrightarrow (11) \rightarrow (13)

(7) → (8) ↔ (9) → (10)

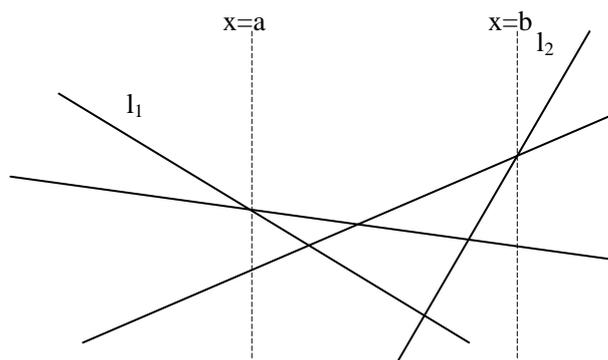


图 3.3 将问题(11)转化为问题(13)

其中, (1)、(5)、(7)已经在代数决策树 (algebraic decision tree) 的框架下被证明具有最低复杂度 $\Omega(n \log n)$, 从而以上列举的所有问题均具有最低复杂度 $\Omega(n \log n)$ 。同时, (4)、(13)、(10)均已经找到了 $O(n \log n)$ 的算法 (参见[16]、[12]、[4]), 从而所有问题也就均具有了 $O(n \log n)$ 的算法, 即最优算法。

推论 3.1 合并两分离的上包迹的算法具有最低复杂度 $\Omega(n)$

证明 设其最低复杂度为 $X(n)$ 。结合分治算法, 可以得到一种计算直线上包迹的算法, 设该算法的复杂度为 $T(n)$, 则有 $T(n)=2T(n/2)+X(n)$, 于是我们得到 $T(n) = X(n) + 2X(\frac{n}{2}) + \dots + 2^k X(\frac{n}{2^k})$ ($2^k \leq n < 2^{k+1}$)。假设 $\lim_{n \rightarrow \infty} \frac{X(n)}{n} = 0$, 则 $\forall \varepsilon > 0$, $\exists m \in \mathbb{R}, \forall t \geq m$, 总有 $\frac{X(t)}{t} < \varepsilon$, 从而 $\forall n > m$,

$$\begin{aligned} \frac{T(n)}{n \log n} &= \frac{\sum_{\frac{n}{2^i} \geq m} 2^i X(\frac{n}{2^i}) + \sum_{\frac{n}{2^i} < m} 2^i X(\frac{n}{2^i})}{n \log n} \\ &= \frac{\sum_{\frac{n}{2^i} \geq m} X(\frac{n}{2^i}) / \frac{n}{2^i}}{\log n} + \frac{\sum_{\frac{n}{2^i} < m} 2^i X(\frac{n}{2^i})}{n \log n} \\ &< \varepsilon + \frac{\sum_{\frac{n}{2^i} < m} 2^i X(\frac{n}{2^i})}{n \log n} \end{aligned}$$

在上式中, 令两边取 $n \rightarrow \infty$, 便有

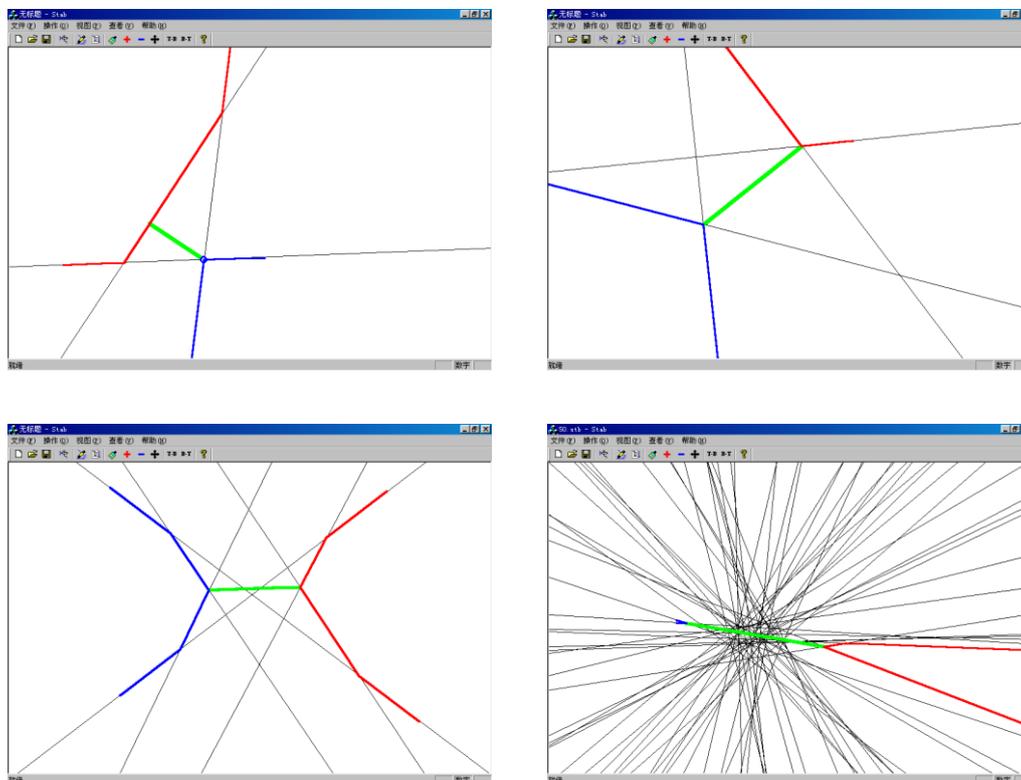
$$\lim_{n \rightarrow \infty} \frac{T(n)}{n \log n} \leq \varepsilon, \forall \varepsilon > 0$$

这与问题 (11) $\Omega(n \log n)$ 的复杂度矛盾。 □

另外, 能否将上述三条链连接起来从而形成一个环呢? 这将是一个有趣的问题。

四、几个计算实例

我们根据上述设计的几个重要算法，用 VC6.0 实现了一个计算直线的最短穿越的演示程序。下面是它运行的几个结果。



[参考文献]

- [1] B. Aronov, J. E. Goodman, R. Pollack, and R. Wenger *On the Helly number for hyperplane transversals to unit balls*
- [2] B. Aronov, J. E. Goodman, R. Pollack, and R. Wenger *A Helly-type theorem for hyperplane transversals to well-separated convex sets*
- [3] M. Atallah and C. Bajaj *Efficient algorithms for common transversals* Information Processing Letters 25(1987), 87-91
- [4] H. Edelsbrunner *Finding transversals for sets of simple geometric figures* Theoretical Computer Science 35(1985), 55-69
- [5] H. Edelsbrunner, H. A. Maurer, F. P. Preparata, A. L. Rosenberg, E. Welzl, and D. Wood *Stabbing line segments* BIT 22(1982), 274-281
- [6] F. P. Preparata *New parallel-sorting schemes* IEEE Trans. on Computers, Vol. C-27, No. 7, July 1978, 669-673
- [7] S. G. Akl *A constant-time parallel algorithm for computing convex hulls* BIT 22(1982), 130-134
- [8] M. H. Overmars *Dynamization of order decomposable set problems* Journal of Algorithms 2(1981), 245-260

- [9] H. Edelsbrunner *Computing the extreme distances between two convex polygons* Journal of Algorithms 6(1985), 213-224
- [10] M. McKenna and G. T. Toussaint *Finding the minimum vertex distance between two disjoint convex polygons in linear time* Comp. & Maths. with Appls., Vol. 11, No. 12(1985), 1227-1242
- [11] J. O'Rourke *An on-line algorithm for fitting straight lines between data ranges* Comm. ACM 24(1981), 574-578
- [12] J. Hershberger *Finding the upper envelope of n line segments in $O(n \log n)$ time* Information Processing Letters 33(1989), 169-174
- [13] M. J. Atallah *Computing the convex hull of line intersections* Journal of Algorithms 7(1986), 285-288
- [14] Y. T. Ching and D. T. Lee *Finding the diameter of a set of lines* Pattern Recognition, Vol. 18(1985), 249-255
- [15] D. Avis and J. M. Robert *Lower bounds for line stabbing* Information Processing Letters 33(1989), 59-62
- [16] M. Keil *A simple algorithm for determining the envelope of a set of lines* Technical Report 91-1, University of Saskatchewan (1991)
- [17] B. Bhattacharya and G. Toussaint *Computing shortest transversals*
- [18] B. Bhattacharya, J. Czyzowicz, P. Eged, I. Stojmenovic, G. Toussaint, and J. Urrutia *Computing shortest transversals of sets*
- [19] 周培德 *计算几何——算法分析与设计* 清华大学出版社, 2000