

3次元凸包

Computational Geometry #11

presentation : ir5

もくじ

- 11.0 概要
- 11.1 3次元凸包の複雑度
- 11.2 乱択逐次構成アルゴリズム
- 11.3 解析
- 11.4 凸包 & 半空間の交差
- 11.5 ボロノイ図

11.0

概要

11.0 概要

- d 次元凸包とは?
 - d 次元上に点集合が与えられるので, それらを被覆する凸集合の中で最小のものを求める問題
 - 数式で書くと, d 次元の点 $p_1, \dots, p_n \in \mathbb{R}^d$ が与えられるので, $\sum \lambda_i p_i$: where $\lambda_i \geq 0$ かつ $\sum \lambda_i = 1$ なる点の存在範囲を求める問題と同じである
- 応用例:
 - 一部の数理計画問題が解けるようになる
 - 複雑な図形の近似



11.0 概要

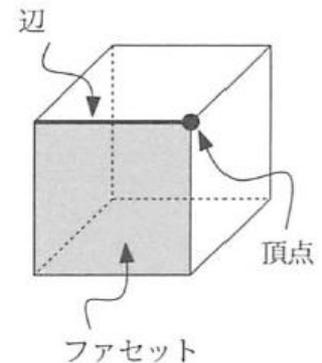
- 今回は $d = 3$ のみに対して考察をする
- 高次元の場合はより状況が複雑であるので考えない
 - $d = 3$ における考察が役に立つことがある, らしい?
 - よくわからん...

11.1

3次元凸包の 複雑度

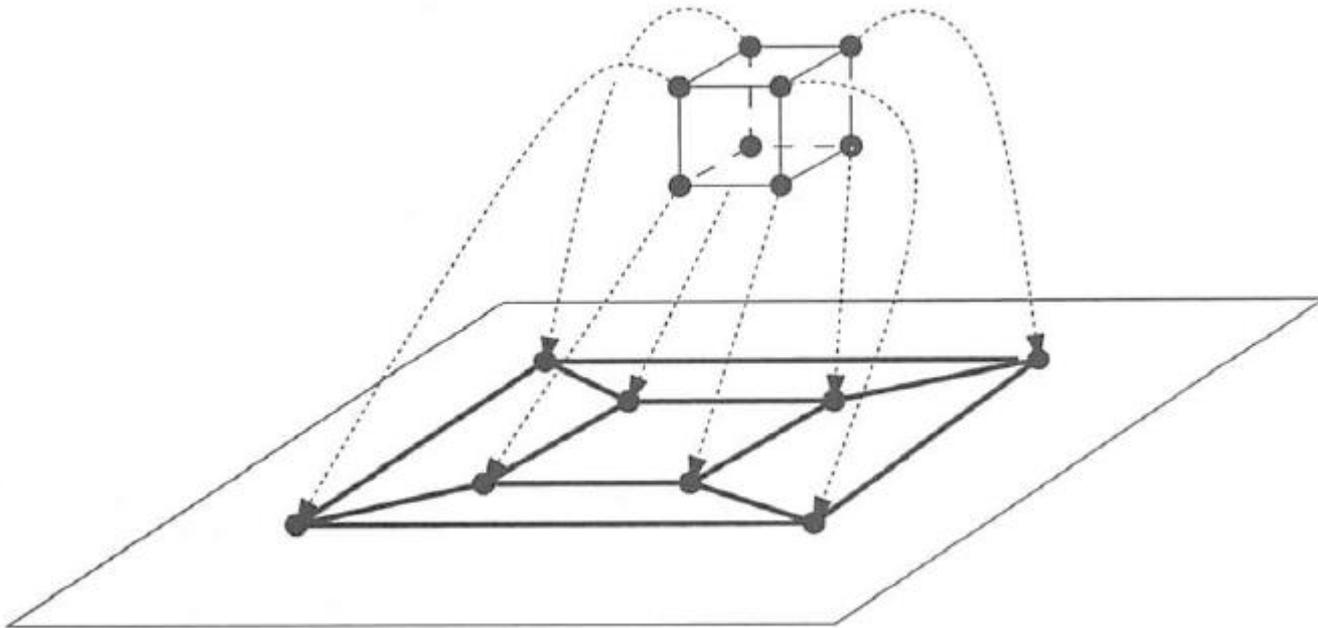
11.1 3次元凸法の複雑度

- 入出力は以下であるとする
 - 入力：3次元上の n 個の点 p_1, \dots, p_n
 - 出力：凸多面体 \mathcal{P}
- \mathcal{P} の ファセット を, 「 \mathcal{P} の境界上で同一平面上にある極大な部分集合」として定義する
 - 3次元の場合はただの面のことである
- \mathcal{P} の複雑度 (頂点数, 辺数, ファセット数の上界) はどの程度になるだろうか?



11.1 3次元凸法の複雑度

- P の接続関係をグラフと見なして平面展開すると, オイラーの公式と平面性によって以下のことが分かる
 - **Th.1** P の辺数の上界は $3n-6$, ファセット数の上界は $2n-4$ である.



11.2

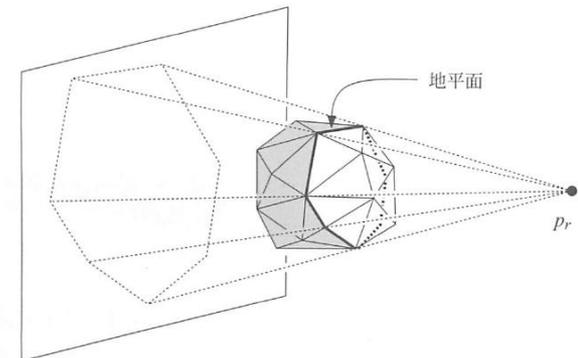
乱択逐次構成 アルゴリズム

11.2 乱択逐次構成アルゴリズム

- 実際にはアルゴリズムをつくっていくことを考える
 - 点集合 P に対し「 $CH(P) := P$ の凸包」とする.
- 与えられた点が全て同一平面上にあるときは2次元に帰着されるのでどうでもよいのでそうでないときを考える
- 点を適当に入れ替えて p_1, p_2, p_3, p_4 が4面体をなすようにする
 - $O(n)$
- 残りの点 p_5, \dots, p_n をランダムに入れ替えて, 順番に追加していくことを考える.
 - $P_r := \{p_1, \dots, p_r\}$ とする

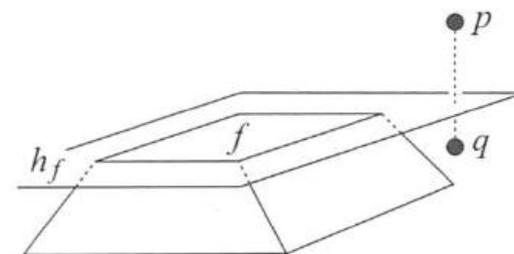
11.2 乱択逐次構成アルゴリズム

- p_r を追加するとき
 - p_r が $CH(P_{r-1})$ の内部or周上にあるならやることは特に無い
 - $CH(P_r) = CH(P_{r-1})$
 - そうでないときは変化が起こる
- p_r を視点として P_r を眺めることを考える. このとき, P_r は(シルエットだけ考えると)多角形に見える
 - この多角形は見える部分と見えない部分の境界である.
 - この境界を地平面と呼ぶことにする.



11.2 乱択逐次構成アルゴリズム

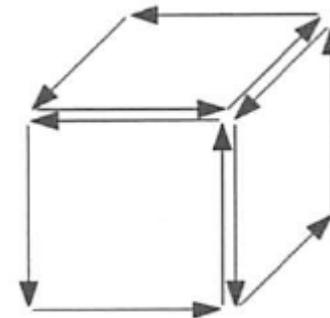
- p_r を追加する
- そもそも「見える」ってどういうことなのか
 - 点 p からある多面体のファセット f が見える
 $\Leftrightarrow f$ を含む平面を $h(f)$ としたとき, $h(f)$ の多面体を含まない側に p が含まれている
- P_{r-1} に p_r を追加したとき, 見えるファセットはすべてなくなり, 見えないファセットはすべて残り続ける.
- 逐次構成において地平面が重要な役割を果たす



f は p からは見えるが
 q からは見えない

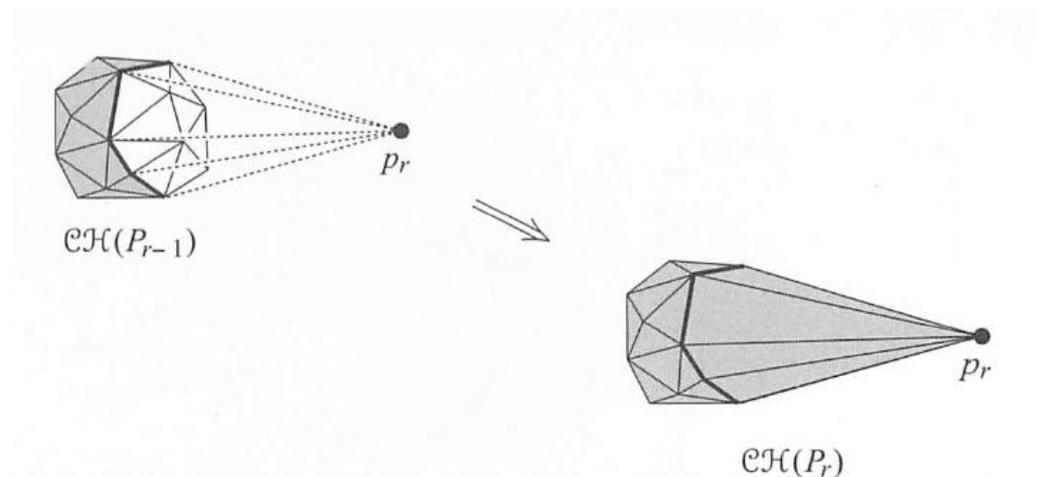
11.2 乱択逐次構成アルゴリズム

- そもそも凸包をどうやってデータ構造として表現するか?
 - 凸多面体は平面展開することができる
 - 凸多面体を2次元上の平面グラフと見なして、例の **2重連結辺リスト** の形で保持しておくことにする.
- 多面体の外側から見たときに、ファセットの点の順番が反時計回りになるように蓄えるものとする。(これによってファセットの向き付けがわかるようになって便利である)



11.2 乱択逐次構成アルゴリズム

- 2重連結辺リストがあれば更新はまあできそうである
 - 見えるファセットと見えないファセットの検出
 - → 見えるファセットの消去
 - → 新しいファセットの追加
 - とかで良い
- P_{r-1} の複雑度は $O(n)$ で、それが n 回ある
 - ので全体で $O(n^2)$ でできる

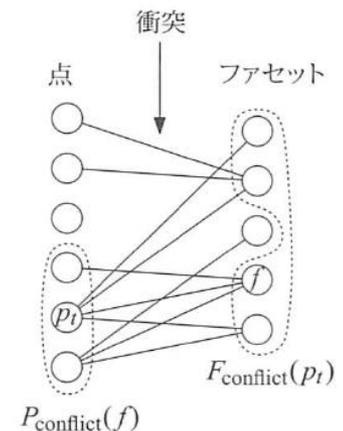


11.2 乱択逐次構成アルゴリズム

- $O(n^2)$ は遅いし恥ずかしいので速くしたい
 - 「衝突」という考えを導入する
- P_r のファセット f と $p_t (t > r)$ は, もし p_t から f が見えるとき, 衝突しているということにする.
- f と p_t が衝突しているとき, f と p_t が両方凸包に入ることはない.
 - $\therefore p_t$ が加えられると f は消滅してしまう.

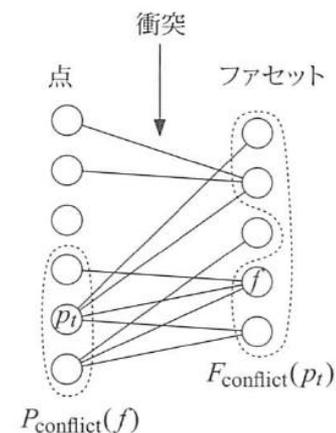
11.2 乱択逐次構成アルゴリズム

- 衝突の関係は2部グラフによって表現できる
 - これを衝突グラフと呼ぶことにし, G で表す.
 - 衝突グラフは隣接リスト形式で保持していくとする
 - $P_{\text{conflict}}(f), F_{\text{conflict}}(p_i)$ と表す
 - 衝突グラフの管理方法について考える
- G の初期化は $O(n)$ でできる
 - 4つの平面と $(n-4)$ 個の頂点について調べるだけ



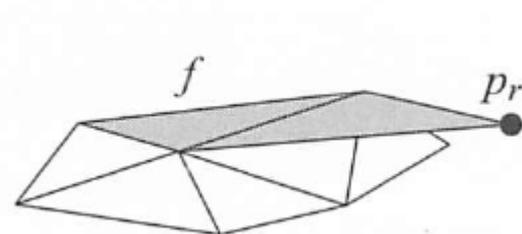
11.2 乱択逐次構成アルゴリズム

- p_r を追加する際に関して
 - 見えるファセットは衝突グラフによってすでに列挙されている
 - P_{r-1} の頂点を全て見ることなくファセットを更新することができる
 - 消えたファセットについては衝突グラフから削除する

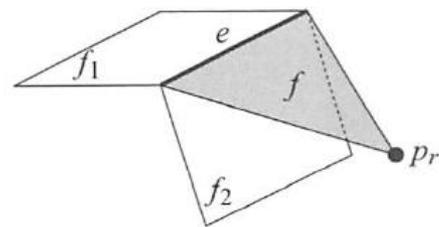


11.2 乱択逐次構成アルゴリズム

- p_r を追加した後について
 - 同一平面に含まれるものと統合されるものは変化無しなので無視して良い
 - そうでないものについては、見る必要がある。すべて3角形をなしていることに留意。



- 新しく追加された3角形のファセット f について考える
- p_r の反対の辺を e とする. もし p_t から f が見えるとすると, p_t から e を見る事が出来る.
- e に隣接していたファセットを f_1, f_2 とすると, p_t は f_1, f_2 の少なくとも1つと衝突していたはずである
 - これにより衝突先の候補を減らす事ができる



11.3

解析

11.3 解析

- さっきのアルゴリズムの性能あんまり自明じゃない感じがするけどなんかよさそうなのでは?
 - 解析する
- 次のことを証明する.

Lemma 3. アルゴリズムによって作られるファセットの個数の期待値は高々 $6n-20$ である.

11.3 解析

Proof.

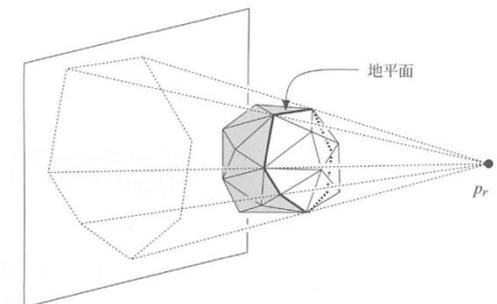
- アルゴリズムを逆順にシミュレートしていくということを考える.
- $CH(P_r)$ から点を取り除いた時に消えるファセットの個数を d_r とおく.
 - 作られるファセットの総数は $4 + \sum_{r=5 \rightarrow n} d_r$ である
- Th.1 により, r 頂点の多面体は高々 $3r-6$ 個の頂点しか持たない.
 - 頂点の順番はランダムに入れ替えているので, 1頂点の平均次数は $6-12/r$ で抑えられる.
- 最初の4頂点はfixされているがその影響とかさほどない
 - $E[d_r] \leq 6$ が得られる
- $\therefore E[\text{作られるファセット}] = E[4 + \sum_{r=5 \rightarrow n} d_r] \leq 6n-20$ となる. ■

11.3 解析

Lemma. 4 アルゴリズムは期待時間 $O(n \log n)$ で凸包を求める.

Proof.

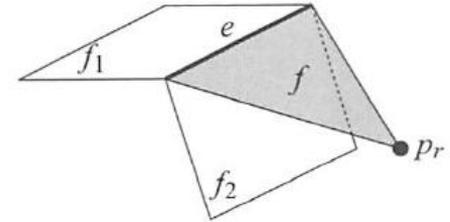
- 初期値の設定は $O(n)$ でできている
- 頂点 p_r を追加するときを考える. (追加されないときはどうでもいい)
 - $F_r := p_r$ を追加することによって消えるファセットの個数とする
- 見えるファセットの削除, 及び新しいファセットの追加, 地平面の検出は $O(F_r)$ ができる.
- Lemma 3 から, $E[\sum_{r=5 \rightarrow n} F_r] = O(n)$ である.



11.3 解析

Proof. (cont.)

- 問題は衝突グラフの更新である.
- 新しく出来たファセットに対する追加について:
 - \mathcal{L} を地平面の辺集合とする. $e \in \mathcal{L}$ に対して f_1, f_2 をそれに隣接するファセットとし, $P(e) := f_1$ を見ることのできる点の個数 + f_2 を見ることのできる点の個数 とおく.
 - 追加にかかる時間は $\sum_{e \in \mathcal{L}} P(e)$ である.
 - これの上界が全体を通じて $O(n \log n)$ であることを後で示す.
- 古いファセットの消去は, 新しく出来たファセットの追加に対する時間程度しかかからないので, 計算量に影響を与えない.
 - \therefore 実行時間は全体で $O(n \log n)$ となる. ■



11.3 解析

- さきほどの解析で一番ややこしそうなところを丸投げしてしまった
 - どうすんのか?
- コンフィギュレーション空間という枠組みを使う
- コンフィギュレーション空間とは
 - 一般的な乱択逐次構成アルゴリズムにおいて複雑度を見積もるための枠組み
 - ほとんどの乱択逐次アルゴリズムは証明方法がほとんど同じでそれらに対して同じような証明をするのは阿呆らしい → 公理的な枠組みが作られた...らしい.

11.3 解析

コンフィギュレーション空間

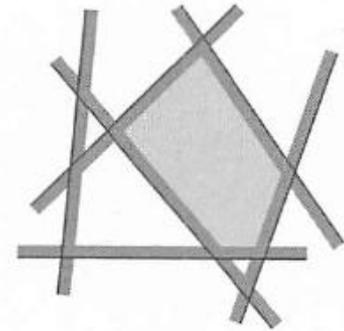
- 定義
 - 4つ組 (X, Π, D, K) として定義される.
 - X : 入力集合 (点集合など).
 - Π : コンフィギュレーションと呼ばれる何か. 問題に応じて柔軟に決められる.
 - $D: \Delta \in \Pi \rightarrow 2^X$ の写像.
 - $D(\Delta)$ の要素は Δ を 定義する という.
 - $K: \Delta \in \Pi \rightarrow 2^X$ の写像.
 - $K(\Delta)$ の要素は Δ を 拒否する という.
- コンフィギュレーション空間は以下の要件を満たさなければならない.
 - $d := \max(|D(\Delta)|; \Delta \in \Pi)$ は定数である.
 - $\forall \Delta \in \Pi, D(\Delta) \cap K(\Delta) = \emptyset$.

11.3 解析

コンフィギュレーション空間 (cont.)

- $S \subseteq X$ に対し, S の上でアクティブなコンフィギュレーションは次で定義されるものである.
 - $T(S) := \{\Delta \in \Pi; D(\Delta) \subseteq S \text{ かつ } K(\Delta) \cap S = \emptyset\}$
- 問題の目標は, X の上でアクティブなコンフィギュレーション $T(X)$ を求めることである.
 - $T(X) := \{\Delta \in \Pi; K(\Delta) \cap X = \emptyset\}$
- なんかよくわからないですね

11.3 解析



- 定義だけだと意味不明なので具体例をみる
- 例として、「入力に半平面がいっぱい与えられるのでその共通部分を求める」という問題を考えることにする.
- 入力 $X :=$ 半平面の集合 である.
- $T(X)$ がこの場合の目標 (=共通部分) になるように頑張って Π , $D(\Delta)$, $K(\Delta)$ を上手に定める.
 - $\Pi :=$ 入力 X の直線上の交点全て
 - $D(\Delta) :=$ 点 Δ を含む2本の直線
 - $K(\Delta) :=$ 点 Δ を含まない半平面全て
 - とおくと $T(X)$ は境界をなす交点全てになる.

11.3 解析

- 目標はこれを乱択逐次構成にて使うことである.
 - X をランダムに並べて x_1, \dots, x_n としたとする
 - $X_r := \{x_1, \dots, x_r\}$ として $\mathcal{T}(X_r)$ を更新していくことを考える
 - コンフィギュレーションの枠組みにより, $\mathcal{T}(X_{r-1})$ から $\mathcal{T}(X_r)$ に更新する際, 増加するコンフィギュレーション $\Delta \in \Pi$ は定義集合が増えることによるもののみであり, 減少する $\Delta \in \Pi$ は拒否集合が増えることによるもののみである.
 - $\mathcal{T}(S) = \{\Delta \in \Pi; D(\Delta) \subseteq S \text{ かつ } K(\Delta) \cap S = \emptyset\}$
- 実行時間について強力な上界が知られている.

11.3 解析

Th 9.14 $|T(X_r) - T(X_{r-1})|$ の期待値は高々 $(d/r) E[|T(X_r)|]$ である.

Th 9.15 $1 \leq r \leq n$ の段階において現れる全てのコンフィギュレーションについての和: $\sum_{\Delta} |K(\Delta)|$ は, 下の値で抑えられる.

$$\sum_{r=1 \rightarrow n} d^2((n-r)/r)(E[|T(X_r)|] / r)$$

11.3 解析

Th 9.14 $|\mathcal{T}(X_r) - \mathcal{T}(X_{r-1})|$ の期待値は高々 $(d/r) \mathbb{E}[|\mathcal{T}(X_r)|]$ である.

Proof.

- X_r から x_r を除くとコンフィギュレーションがいくら減少するかを考える.
- とりあえず X_r をある $X_r^* \subseteq X$ (s.t. $|X_r^*| = r$) に固定しておいて考える.
- X_r からランダムに選んだ要素 x_r を除くことによってあるコンフィギュレーション $\Delta \in \Pi$ が消えたとすると, それは $x_r \in D(\Delta)$ であったはずである.
- $\Delta \in \mathcal{T}(X_r), x \in D(\Delta)$ なる対 (Δ, x) は高々 $d \cdot |\mathcal{T}(X_r)|$ 個しかない

11.3 解析

Th 9.14 $|T(X_r) - T(X_{r-1})|$ の期待値は高々 $(d/r) E[|T(X_r)|]$ である.

Proof. (cont.)

- 従って次式が成り立つ.

$$\sum_{x \in X_r} \text{card}(\{\Delta \in \mathcal{T}(X_r) \mid x \in D(\Delta)\}) \leq d \cdot \text{card}(\mathcal{T}(X_r))$$

- よって X_r から消す要素 x_r をランダムにしたときの, 消えるコンフィギュレーションの個数の期待値は $(d/r)|T(X_r)|$ である.
- X_r を固定しないものとする, 期待値として $(d/r) E[|T(X_r)|]$ を得る.



11.3 解析

Th 9.15 $1 \leq r \leq n$ の段階において現れる全てのコンフィギュレーションについての和: $\sum_{\Delta} |\mathbf{K}(\Delta)|$ は, 下の値で抑えられる.

$$\sum_{r=1 \rightarrow n} d^{2((n-r)/r)} (\mathbf{E}[|\mathcal{T}(X_r)|] / r)$$

証明略... ■

11.3 解析

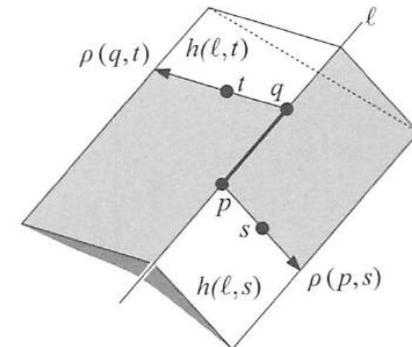
話を3次元凸包に戻す.

- 証明したいのは, $\sum_e P(e) = O(n \log n)$ であった. ここで, e は各段階における地平面の辺すべてについて和を取るものとする.
- コンフィギュレーションの枠組みを次のように定義する.

- 入力 $X :=$ 入力点 (そのまんま)
- $\Pi :=$ フラップの集合
 - フラップ: 凸包の辺を拡張したもので, 4つ組 (p, q, s, t) で表す.

- $D(\Delta) := \{p, q, s, t\}$

- $K(\Delta) :=$ して, 拒否集合 (killing set) $K(\Delta)$ はもっと可視化しにくい. p と q を通る直線を ℓ とする. 点 x が与えられたとき, ℓ を境界とする半平面のうち x を含む方を $h(\ell, x)$ という記号で表す. 2点 x, y が与えられたとき, x を始点とし y を通る半直線を $\rho(x, y)$ とする. 点 $x \in S$ が $K(\Delta)$ に含まれるための必要十分条件は, それが次の領域のうちの1つに含まれることである.
 - $h(\ell, s)$ と $h(\ell, t)$ によって定義される閉じた凸3次元楔形領域の外部.
 - $h(\ell, s)$ の内部であるが, $\rho(p, q)$ と $\rho(p, s)$ によって定義される閉じた2次元楔形領域の外部.
 - $h(\ell, t)$ の内部であるが, $\rho(q, t)$ と $\rho(q, p)$ によって定義される閉じた2次元楔形領域の外部.
 - 直線 ℓ の内部であるが, 線分 \overline{pq} の外部.
 - 半直線 $\rho(p, s)$ の内部であるが, 線分 \overline{ps} の外部.
 - 半直線 $\rho(q, t)$ の内部であるが, 線分 \overline{qt} の外部.



11.3 解析

次のことが成り立つように定義されていることがわかっているだけで十分だと思う

Lemma. 5 フラップ $\Delta=(p,q,s,t)$ が $T(S)$ に含まれる

\Leftrightarrow 辺 pq, qs, st が凸包 $CH(S)$ の辺で, 辺 pq, ps に接続するファセット f_1 及び辺 pq, qt に接続するファセット f_2 が存在する.

また, $CH(S)$ のファセット f_1 か f_2 が点 $x \in P$ から見えるとき, $x \in K(\Delta)$ である. ■

11.3 解析

Lemma.6 $\sum_e P(e) = O(n \log n)$

Proof.

- $CH(P_{r-1})$ 上で p_r を追加するときの地平面の辺 e について考える.
- $\Delta=(p,q,s,t)$ を $pq=e$ なる2つのフラップのうちの1つとする.
- Lemma. 5 より, $\Delta \in \mathcal{T}(P_{r-1})$ である. また, $P(e) \subseteq K(\Delta)$.
- $E[|\mathcal{T}(P_r)|] = 2 \cdot (CH(P_r) \text{の辺数}) \leq 6r-12$ である.
- Th 9.15 によって $\sum_{\Delta} |K(\Delta)|$ を抑えることを考えると次の式が成り立つ. ■

$$\sum_e \text{card}(P(e)) \leq \sum_{\Delta} \text{card}(K(\Delta)) \leq \sum_{r=1}^n 16 \binom{n-r}{r} \left(\frac{6r-12}{r} \right) \leq 96n \ln n$$

11.3 解析

以上で次の定理が得られた.

Th.7 3次元凸包は乱択アルゴリズムにより $O(n \log n)$ の期待時間で求められる.

11.4

凸包と 半空間の交差

11.4 凸包と半空間の交差

- 前回「双対変換」なんてのが出て来ました

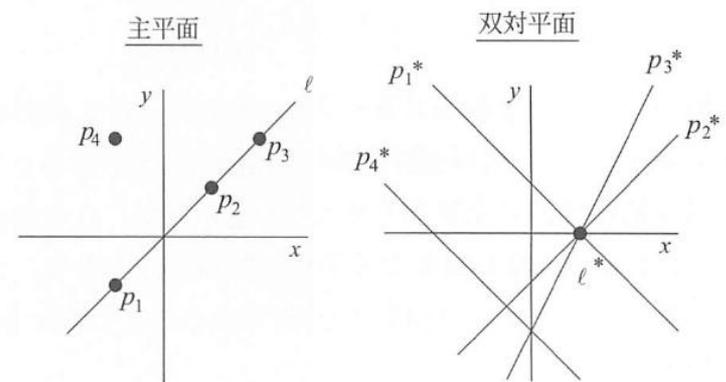
- 直線 $y=ax+b \leftrightarrow$ 点 $(a, -b)$ の変換.

- $p \rightarrow p^*, l \rightarrow l^*$ とするとき,

- $p \in l \Leftrightarrow l^* \in p^*$
- p が l より上にある $\Leftrightarrow l^*$ が p^* より上にある
- の2つが成り立つ.

- 3次元においても $z=ax+by+c \leftrightarrow$ 点 $(a, b, -c)$ などとすれば同様のことが成り立つ.

- ここでは双対平面(空間)上で凸包が何に対応するのかを調べる.



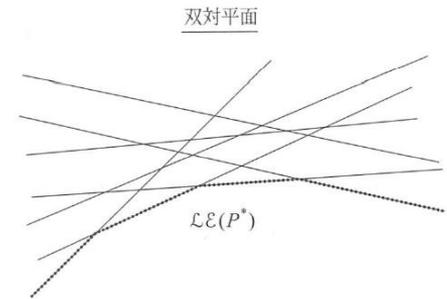
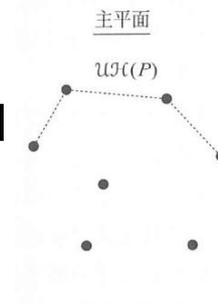
11.4 凸包と半空間の交差

- とりあえず2次元で考える
- めんどいので入力点の x 座標は pairwise distinct とする.
- 上部凸包に焦点を当てる.
- 「直線 l が P の上部凸包の境界をなしている

⇔ l が P の点を少なくとも2点は通る
かつ $\forall p \in P, l$ は p より上側にある

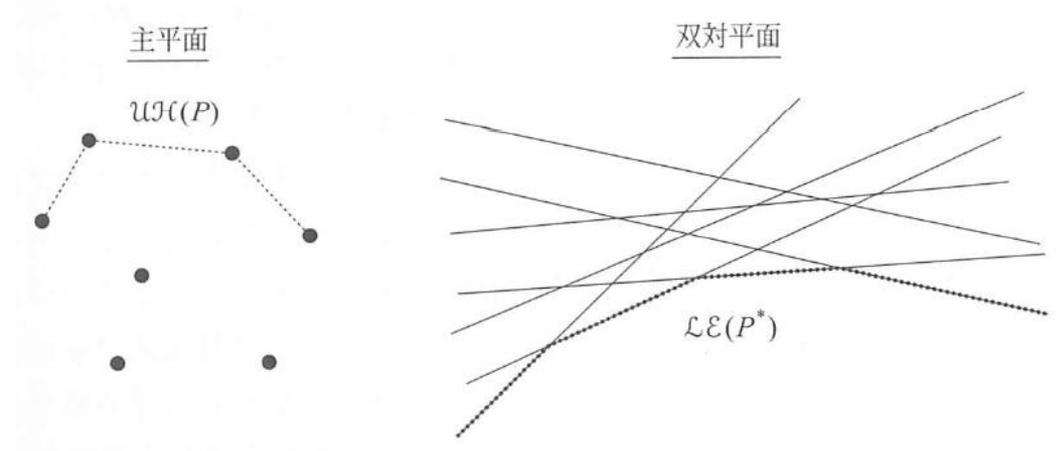
- である.
- これを双対の言葉で書くと:

⇔ 点 l^* が P^* の直線と少なくとも2回は交差する
かつ $\forall p^* \in P^*, l^*$ は p^* より下側にある



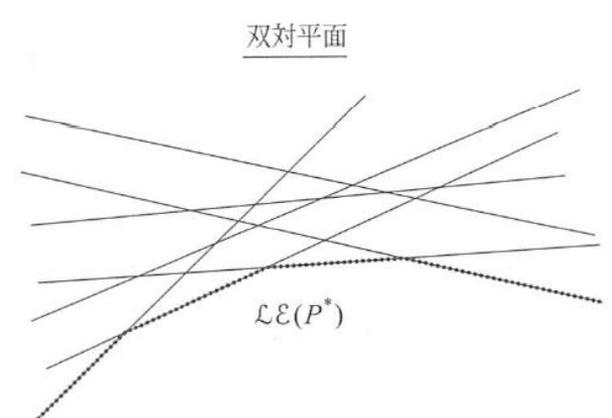
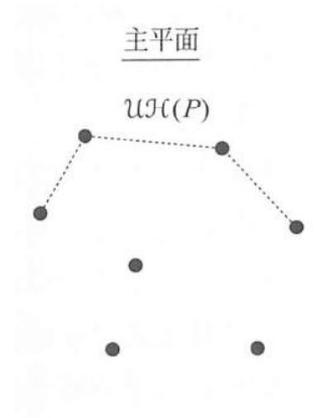
11.4 凸包と半空間の交差

- これは何を意味しているのか?
 - 主平面における上部凸包は, 双対平面の直線群における下部の直線チェーンに対応している
 - このチェーンを下側エンベロップと呼ぶ.
- 直線の傾きの大小を考えると, 「主平面における左→右の点」は「双対平面における右→左の直線」に対応している.
- 下部凸包と上側エンベロップの対応も同様に成り立つ.



11.4 凸包と半空間の交差

- これを利用して、平面上の半平面の共通部分を凸包によって求めることができる.
 - 上側の半平面を集合を H_+ , 下側の半平面の集合を H_- とする.
 - H_+^* の下部凸包を求めることによって $\cap H_+$ が求められる.
 - 同様にして $\cap H_-$ も求められる.
 - あとは $(\cap H_+) \cap (\cap H_-)$ を計算すればよい.
- 3次元でも同様である.



~ *END* ~