線分族と三角形族の中での直交探索

5L－4

徳山 豪
日本ＩＢＭ東京基礎研究所[1]

1. **要旨.** 空間内の三角形または線分の集合に対して、軸方向の直交領域と交わる図形を全て列挙する問題（直交探索）に対して、効率の良いアルゴリズムを考える。もっとも興味深いのは、探索の効率が、データの個数以外の様々な幾何学的パラメターに依存している事である。

2. **領域探索問題.** d 次元空間 ($d \geq 2$) 内の n 個の対象物をデータ構造に蓄え、与えられた空間の領域（質問領域）と交わるものを効率良く探索、列挙出来るようにする問題は領域探索問題と言われており、計算幾何学の基本問題の一つである [7, 8]。対象物が点であり、質問領域が軸方向の直方体である場合は、非常に多くの研究成果があり、$O(n \log^{d-1} n)$ のデータ空間を用いれば $O(\log^{d-1} n)$ 時間で探索が行なえる [5]。また、質問領域を単体に広げる（単体探索）と、$O(n)$ のデータ空間を用いて $O(n^{1-1/d})$ 時間で探索が行なえる [6] が、これ以上の効率化はほとんど出来ない事が知られている [3]。

一方、対象物が点でない時は、CAD 等で多くの応用があるにも関わらずあまり研究されていない。その理由の一つは、例えば対象物が平面上の直線の集合であり、質問領域が長方形であるときに、双対変換によって、上に書いた単体探索と理論的に同等であることが示せるからである。従って、単体探索が難しいことを考慮すると、点集合以外のデータを扱う領域探索は難しいだろうと考えられていた。

3. **線分と三角形の直交探索.** この論文では、対象物が d 次元空間内の線分あるいは三角形であり、質問領域が軸方向の直方体である場合（直交探索）を考える。直線が線分の特殊な場合であることを考えると、上に挙げた双対変換による言い替えを行なうと、この問題は少なくとも単体探索よりも難しいはずである。しかしながら、我々は、線分や三角形の直交探索は理論的にほとんどの場合単体探索より易しいことを示す。

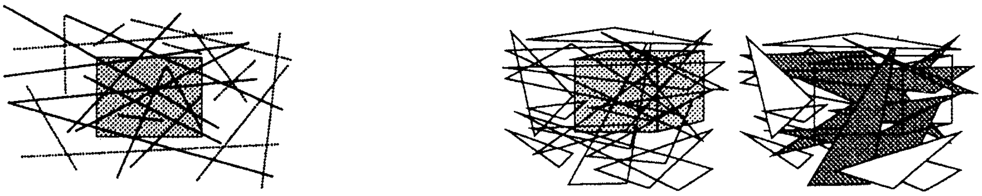

図 1: 直交探索 （２次元の線分族、３次元の三角形族)

二次元の n 本の線分の族の場合を考えよう。これらの線分の内が互いに交わる線分対が K 対あるとする。直線族の場合には $K = n(n-1)/2$ だが、普通の CAD などの入力では K はそれほど多くない。我々のデータ構造は K が小さい時はとても効率が良い。

Theorem 1 $O(m)$ ($m \geq n \log n$) のデータ空間を用いて、$O((K/m)^{1/2} \log^{3.5} n)$ 時間で二次元線分の直交探索が行なえる。

高次元の場合には、K として、軸方向の２次元部分空間への線分たちの射影の交差数を用いる。すると、

Theorem 2 $O(m)$ ($m \geq n \log^{d-1} n$) のデータ空間を用いて、$\tilde{O}((K/m)^{1/2})$ 時間で d 次元線分の直交探索が行なえる。ただし、$\tilde{O}$ は対数部分を除いた複雑度の表示。

同様に、d 次元空間の n 個の三角形に対しては、三角形の三次元部分空間への射影の交差対の数を M、辺たちの射影の交差対の数を K とすると、

[1]Orthogonal Queries in Segments and Triangles
Takeshi Tokuyama, IBM Japan Ltd. Tokyo Research Laboratory
1623-14, Shimo-Tsuruma, Yamato-shi, Kanagawa, 242, Japan.

Theorem 3 $O(m)$ $(m \geq n \log^{d-1} n)$ のデータ空間を用いて、$\tilde{O}((K/m)^{1/2} + M^{1/2}/m^{1/3})$ 時間で d 次元の三角形の直交探索が行なえる。

4. 手法. 二次元の線分族の時の手法を簡単に紹介する。ある線分が長方形の質問領域 W と交差するのは、その線分の端点の一つが W に入るか、又は線分が W の辺と交わるかのどちらかである。前者は端点の集合に対する直交領域探索のデータ構造を用意しておけば、探索が出来る。したがって、W の辺と交わる線分を探索すれば良い。これは ray-shooting もしくは ray-stabbing query と呼ばれるものになる [1]。

我々の場合は W の辺は垂直もしくは水平方向なので、場合分けして、水平方向の辺のみを考えよう。線分たちを y 軸に投影すると、軸上の区間の集合が得られる。これらを区間木というデータ構造 [7] に蓄えると、各区間は幾つかの基本区間の和で表される。各基本区間に対応する部分を元の２次元平面に持ちあげると、２本の水平直線に区切られた帯の内部で、端点をそれらの水平直線上に持つ線分の集合が得られる。これをハンモックと呼ぶ。

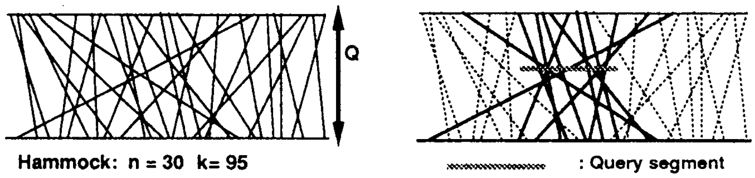

図 2: ハンモックでの探索

Lemma 4 ハンモックに対して直交探索が出来れば、一般の線分の族に対する直交探索に対する $O(\log n)$ 倍の大きさのデータ構造が作れる。

次に、ハンモックを幾つかの小さいが密な部分に（グラフ分解の手法を用いて）分割する。そして、各々の密な部分に一般の場合の理論的な ray-stabbing query の構造を持たせる。すると、

Lemma 5 $O(m)$ $(m \geq n)$ のデータ空間を用いて、$O((K/m)^{1/2} \log^{2.5} n)$ 時間でハンモックでの直交探索が行なえる。

この補題から定理１が導かれる。定理２、定理３はスタンダードな高次元化の手法で得ることが出来る。上の補題は動的計算幾何学 [2] の意味では、ある種のリスト探索構造 [4] のデータ空間と探索時間のトレードオフを示している。

参考文献

[1] P. Agarwal, Ray Shooting and Other Applications of Spanning Trees with Low Stabbing Number, *Proc. 5th ACM Computational Geometry Symposium,* (1989), 315-325.

[2] M. Atallah, Some Dynamic Computational Geometry Problems, *Computers and Mathematics with Applications,* **11** (1985), 1171-1181.

[3] B. Chazelle, Lower Bounds on the Complexity of Polytope Range Searching, *J. Amer. Math. Sci.,***2** (1989) 637-666.

[4] J. Driscoll, N. Sarnak, D. Slator, and R. Tarjan, Making Data Structure Persistent, *J. Computer and Systems Sci.***38** (1989), 86-124.

[5] G. Lueker, A Data Structure for Orthogonal Range Queries, *Proc. 19th IEEE FOCS* (1978), 28-34.

[6] J. Matoušek, Range Searching with Efficient Hierarchical Cuttings, *Proc. 8th ACM Computational Geometry Symposium* (1992), 276-285.

[7] F. Preparata and M. Shamos, *Computational Geometry, an Introduction*, 2nd edition, Springer-Verlag (1988).

[8] F. Yao, Computational Geometry, in Handbook of Theoretical Computer Science A, van Leeuwen ed. Elsevier (1991).