

# 면과 선분의 공간 관계 발견 알고리즘

홍 은지<sup>1</sup>, 정 소영, 유 석인

<sup>1</sup>성공회대학교 컴퓨터정보학부, 서울대학교 전산과학과

## An Algorithm to Find Spatial Relationship between an Area and a Line Segment

Eunji Hong<sup>1</sup>, SoYoung Jeong, and Suk I Yoo

<sup>1</sup>Dept. of Computer & Information Science of SungKongHoe Univ., Dept. of Computer Science of Seoul National Univ

### 요약

지리 정보 시스템(Geographic Information System, GIS)에서 가장 중요한 부분을 차지하는 것 중의 하나가 공간 데이터 모델(spatial data model)에서 정의된 각 공간 데이터들간의 공간 관계 연산자(spatial relational operator)의 효과적인 구현이다. 공간 데이터는 점(point), 선(line), 면(area)으로 표현될 수 있다. 이들 사이의 모든 공간 관계는 Disjoint, Touch, Cross, In, Overlap 의 다섯 가지 연산자로 표현 가능함이 알려져 있으며, 이들에 대한 실제적인 위상 관계를 표현하는 다양한 수학적 모델링 방법이 존재한다. 하지만, 실제 이들 공간 연산자들을 수학적 모델에 따라 그대로 구현하려고 하면, 컴퓨터 상에서는 표현할 수 없거나 많은 자원을 차지하는 데이터 구조를 필요로 한다거나, 또는 비효율적인 알고리즘으로 구현할 수 밖에 없는 현실적인 어려움에 봉착한다. 그 중에서도 구현하기 어려운 연산은 면과 선과의 관계, 면과 면과의 공간 관계를 찾아내는 공간 연산자이다. 본 논문에서는 선분의 양끝점을 이용하여 면과 선분(line segment)과의 관계를 찾아내는 알고리즘을 제안한다. 이 알고리즘을 사용하여 면과 선, 면과 면과의 관계를 찾아내는 공간 연산자를 효율적으로 구현할 수 있다.

### 1. 서론

지리 정보 시스템(Geographic Information System, GIS)에서 가장 중요한 부분을 차지하는 것 중의 하나가 공간 데이터 모델(spatial data model)에서 정의된 각 공간 데이터들간의 공간 관계 연산자(spatial relational operator)의 효과적인 구현이다. 예를 들어, 지리 정보 시스템을 이용하여 “서울시에서 한강이 지나가는 구의 이름을 모두 찾으라”는 명령이라든지, “종로구와 인접한 모든 구의 이름을 찾으라”는 명령 등은 모두 공간 관계 연산자를 이용한 응용 분야가 될 수 있다.

공간 데이터는 점(point), 선(line), 면(area)으로 표현될 수 있다. 점은 면적이나 길이를 가지지 않고 특정 위치(location)를 나타낸다. 선은 1차원으로 면적은 없고 길이만 가지며, 시작점과 끝점이 있다. 만약 선이 직선이라면, 시작점과 끝점만으로도 표현될 수 있지만, 복잡한 선은 선의 방향이 바뀔 때마다 중간 점들에 대한 정보를 가지고 있어야 한다. 따라서, 복잡한 선은 직선으로 표현되는 선분(line segment)들의 집합으로 표현된다. 면은 닫힌 공간으로 2차원이며 면적을 가진다. 면의 경계선에 대한 정보를 유지하여 면을 표현할 수 있다. 면의 경계선은 시작점과 끝점이 같고, 선분들로 표현된다.

공간 관계 연산자에 대한 엄밀한 개념 정립을 위해서 기하학적, 위상 수학적 접근 방법이 다양하게 적용되어 왔다. 그 결과 공간 데이터 모델사이의 모든 공간 관계는 Disjoint, Touch, Cross, In, Overlap 의 다섯 가지 연산자로 표현 가능함이 알려져 있으며[1] 그에 대한 실제적인 위상 관계를 표현하는 방법으로는 point-set theory 를 이용한 방법[2], calculus-based method 를 이용한 방법[3] 등 다양한 수학적 모델링 방법이 존재하고 있다.

GIS 분야에서 통일된 데이터 모델로 규격화하려는 움직임으로 건설된 OpenGIS[4]에서 정의된 공간 데이터 모델 역시 [1]에 언급된 기본적인 5개의 공간 연산자에 기반하여 공간 관계를 정의하고 있으며 그리 현 데이터들의 위상 기하학적 공간 관계를 수학적 방법으로 표현하기 위해

서 DE+9IM 모델링 방법을 제시하고 있다[5]. DE+9IM 방법은 기본적으로 공간 데이터의 내부(interior) 영역, 외부(exterior) 영역, 경계(boundary) 영역을 정의하고 그들 간의 관계를 행렬로 표현하여 각각의 공간 연산자들이 나타내는 정의에 대응시켜 표현하는 수학적 방법이긴 하지만, 실제 이들 공간 연산자들을 수학적 모델에 따라 그대로 구현하려고 하면 컴퓨터 상에서는 표현할 수 없거나 많은 자원을 차지하는 데이터 구조를 필요로 한다거나, 또는 비효율적인 알고리즘으로 구현할 수 밖에 없는 현실적인 어려움에 봉착한다. 그 중에서도 구현하기 어려운 연산은 면과 선과의 관계, 면과 면과의 관계를 찾아내는 공간 연산자이다.

앞에 기술한 바와 같이, 면과 선은 모두 선분들의 집합으로 이루어져 있으므로, 면과 선, 면과 면과의 관계를 찾아내는데 선분을 이용할 수 있다. 본 논문에서는 면과 선분과의 공간 관계를 찾아내는 알고리즘을 제안한다. 본 논문에서 제시하는 알고리즘의 중요 아이디어는 선분을 이루는 시작점과 끝점이 어떤 공간 관계를 가지는지에 따라 분류한 후, 각 경우마다 면과 선분이 가질 수 있는 공간 관계를 찾아내는 것이다. 이 알고리즘을 사용하여 면과 선, 면과 면과의 관계를 찾아내는 공간 연산자를 효율적으로 구현할 수 있다.

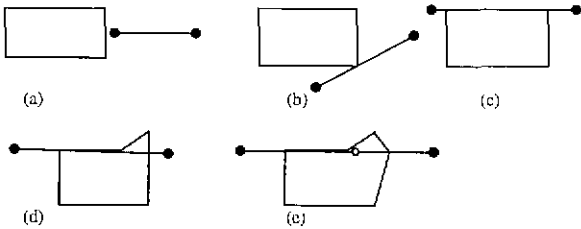
### 2. 면과 선분의 공간 관계 발견 알고리즘

본 논문에서 제시하는 알고리즘의 중요 아이디어는 선분을 이루는 시작점과 끝점이 면과 어떤 공간 관계를 가지는지에 따라 분류한 후 각 경우마다 면과 선분이 가질 수 있는 공간 관계를 찾아내는 것이다.

우선 점과 면 사이에는 Touch, Disjoint, In 의 세가지 관계성이 성립할 수 있다. 따라서, 선분의 양 끝점이 면과 가지는 공간 관계 가능성은  $3 \times 2 = 6$  가지이다. 선분과 면은 Overlap 할 수 없으므로, 선분과 면은 Touch, Disjoint, In, Cross 중 하나의 관계를 가진다. 선분의 양끝점과 면과의 공간 관계 분류에 따른 선분과 면과의 관계가 [표 1]에 있다. 지

면 관계상, 모든 관계를 모두 살펴볼 수는 없으므로, 대표적인 것만 몇 개 살펴본다.

양 끝점이 모두 Disjoint 한 경우를 살펴보자 이 경우 선분 L이 면 A에 In 되는 것은 길데 불가능하다 하지만, 나머지 세 공간 관계는 모두 가능하다. 예는 [그림 1]과 같다



[그림 1] 선분의 양끝점이 면에 Disjoint 할 경우

대부분의 경우는 A를 이루는 모든 선분에 대해 L이 어떤 관계인지를 검사해야지 A와 L이 어떤 공간 관계인지를 파악할 수 있다 예를 들어, A를 이루는 모든 선분이 L과 Disjoint하다면 A와 L은 Disjoint한다고 판단할 수 있다. 하지만, A의 선분 중 L과 Cross인 선분이 존재한다면, 남은 다른 A의 선분과 L과의 관계는 검사할 필요 없이 A와 L은 Cross한다고 판단할 수 있다 [그림 1]에서 (d)는 A를 구성하는 선분 중 오른쪽 선분이 L과 Cross하고 있으므로, A와 L은 Cross한다. 하지만, (e)에서는 L과 Cross하는 A의 선분이 하나도 없다 Touch하거나 in하는 선분들이 있을 뿐이다 단, L이 A의 내부를 통과하고 있다는 것을 알 수 있다. 이를 판단하기 위해서, L이 A의 선분과 Touch하는 경우, 그 교차점에서 L선에 있는 직전 또는 직후의 점((e)에서 °으로 표시된 곳)이 A의 내부에 있는지를 검사한다. 해당 점이 A의 내부에 있다는 것은 L이 A의 내부와 교집합이 있다는 것이고, 양끝점이 외부에 있으므로, L은 A와 Cross한다고 결론 내릴 수 있다 하지만, L과 Touch하는 선분이 A에 존재하지만, 그 교차점에서 L상에 있는 직전 또는 직후의 점이 A의 내부에 있지 않다면, (b)와 같이 L과 A는 Touch한다 (c)와 같이 L에 in되는 선분이 A에 존재한다면, L은 A에 Touch한다

이를 표로 나타낸 것이 [표 1]이다 테이블의 열은 선분 L의 양끝점이 A와 어떤 관계인지를 나타낸 것이고, 테이블의 행은 L과 A의 공간 관계 가능성을 나타낸 것이다 선분 L의 양끝점이 A와 Disjoint할 경우, L과 A가 Disjoint할 가능성이 있는데, L이 A의 모든 선분과 Disjoint해지거나 L과 A가 Disjoint하다는 것을 나타내고 있다

면과 선분의 공간 관계 판단 알고리즘은 [그림 2]와 같다 알고리즘의 입력은 면을 나타내는 A와 선분을 나타내는 L이다 A는 A를 이루는 선분들의 집합( $I_1, I_2, I_3, \dots, I_n$ )으로 이루어져 있으며, L은 시작점  $ps$ 와 끝점  $pe$ 로 이루어져 있다. 출력은 A와 L간의 공간 관계로서 Touch, Disjoint, In, Cross 중의 하나이다

Input Area A( $I_1, I_2, \dots, I_n$ )  
 Insegment L( $ps, pe$ )  
 Output A와 L과의 공간 관계  
 (In, Cross, Touch, Disjoint 중 하나)

foundIn <- False  
 foundTouch <- False  
 foundDisjoint <- False

if ( $ps$ 와  $pe$ 가 모두 A에 In하거나

```

두 점 중 하나는 In, 하나는 touch인 경우)
while q in (I1, ..., In)
    if (L이 q에 Cross 하면)
        return Cross
    else if (L이 q에 touch 하고, L과 q의 교차점의 직전
    또는 직후의 L 상의 점이 A에 Disjoint 하면)
        return Cross
    end if
end while
return In
else if (두 점 중 하나는 In, 하나는 Disjoint인 경우)
    return Cross
else if (두 점 중 하나는 Touch, 하나는 Disjoint인 경우)
    while q in (I1, ..., In)
        if (L이 q에 cross 하면)
            return Cross
        else if (L이 q에 Touch 하고, L과 q의 교차점의 직전
        또는 직후의 L 상의 점이 A에 in 하면)
            return Cross
        end if
    end while
    return Touch
else if (두 점 모두 A에 Touch 하는 경우)
    while q in (I1, ..., In)
        if (L이 q에 cross 하면)
            return Cross
        else if (L이 q에 Touch 하면)
            if (L과 q의 교차점의 직전 또는 직후의
            L 상의 점이 A에 in 하면)
                foundIn <- True
            else (L과 q의 교차점의 직전 또는 직후의
            L 상의 점이 A에 Disjoint 하면)
                foundDisjoint <- True
            end if
            if (foundDisjoint가 True 이고
            foundIn이 True 이면)
                return Cross
            end if
        end if
    end while
    if (foundIn이 True 이면)
        return In
    else
        return Touch
    end if
else // 두 점 모두 Disjoint인 경우
    while q in (I1, ..., In)
        if (L이 q에 cross 하면)
            return Cross
        else if (L이 q에 Touch 하면)
            if (L과 q의 교차점의 직전 또는 직후의 L 상의
            점이 A에 in 하면)
                return Cross
            else
                foundTouch <- True
            end if
        else if (q가 L에 In 하면)
            foundTouch <- True
        end if
    end while
    if (foundTouch가 True 이면)
        return Touch
    else
        return Disjoint
    end if
end if
    
```

[그림 2] 면과 선분의 공간 관계 발견 알고리즘

	선분(L)의 양 끝 점이 둘 다 A에 Disjoint	둘 다 A에 in	하나는 in, 하나는 Touch	하나는 Disjoint 하나는 in	하나는 Touch, 하나는 Disjoint	둘 다 Touch
A와 L이 Disjoint	가능	절대 불가	절대 불가	절대 불가	절대 불가	절대 불가
Disjoint 할 추가 조건(L이)	A의 모든 선분과 Disjoint					
Touch	가능	절대 불가	절대 불가	절대 불가	가능	가능
Touch 할 추가 조건(L이)	Disjoint와 cross가 아니면 성립				A와 cross 하지 않다면 성립	in도 Cross도 아니면 성립
Cross	가능	가능	가능	가능	가능	가능
Cross 할 추가 조건(L이)	Cross 하는 선분이 A에 존재한다 또는 Touch 하는 선분이 A에 존재한다면, 교차점의 직전 또는 직후 점이 A에 In 한다	Cross 하는 선분이 A에 존재한다. 또는 Touch 하는 선분이 A에 존재한다면, 교차점의 직전 또는 직후의 L 상의 점이 A와 Disjoint.	Cross 하는 선분이 A에 존재한다 또는 L과 Touch 하는 A의 선분과의 교차점 직전 또는 직후의 L 상의 점이 A와 Disjoint	성립	Cross 하는 선분이 A에 존재한다 또는 Touch 하는 선분이 A에 존재한다면, 교차점의 직전 또는 직후 점이 A에 In 한다	Cross 하는 선분이 A에 존재한다 또는 또는 Touch 하는 선분이 A에 존재한다면, 교차점의 직전 또는 직후 점이 A에 in 하는 경우도 있고 Disjoint 하는 절도 있다
In	절대 불가	가능	가능	절대 불가	절대 불가	가능
in 할 추가 조건(L이)		Cross가 아니면 성립	Cross가 아니면 성립			L과 Touch 하는 A의 선분의 교차점 직전(또는 직후)의 L 상의 점이 A에 In

[표 1] 선분의 양끝점과 면과의 공간 관계 분류에 따른 선분과 면과의 관계

3. 결론

본 논문에서는 면과 선분과의 공간 관계를 찾아내는 알고리즘을 개진하였다. 본 논문에서 제시하는 알고리즘의 중요 아이디어는 선분을 이루는 시작점과 끝점이 면과 어떤 공간 관계를 가지는지에 따라 분류한 후, 각 경우마다 면과 선분이 가질 수 있는 공간 관계를 찾아내는 것이다. 이 알고리즘을 사용하여 면과 선, 면과 면과의 관계를 찾아내는 공간 연산자를 효율적으로 구현할 수 있다. 서울대학교 전산학과 인공지능연구소에서는 본 논문에서 제안한 알고리즘에 따라 공간 연산자를 구현하였다.

참고 문헌

- 1 Clementini, Eliseo, Di Felice P, A Model for Representing Topological Relationships Between Complex Geometric Features in Spatial Databases, *Information Sciences* 90(1-4):121-136, 1996
- 2 Max J Egenhofer Robert D Franzosa Point-set topological spatial relations *International Journal of Geographical Information Systems*, 5(2) 161-174, 1991
- 3 Eliseo Clementini, Paolo Di Felice, Peter van Oosterom A small set of formal topological relationships suitable for end-user interaction *Third International Symposium on Large Spatial Databases, Lecture Notes in Computer Science no. 692, pages 277-295 Singapore, June 1993 Springer-*

Verlag.

- 4 Open GIS Consortium, *OpenGIS Specification Model*, 1997
- 5 Clementini E, Di Felice P, A Comparison of Methods for Representing Topological Relationships, *Information Sciences* 80, 1-34, 1994