

## 공차를 고려한 다각형 영역의 내외부 판별 알고리즘

정연찬\*, 박준철\*\*

### Tolerance-based Point Classification Algorithm for a Polygonal Region

Chung, Y. C.\* and Park, J. C.\*\*

#### ABSTRACT

This paper details a robust and efficient algorithm for point classification with respect to a polygon in 2D real number domain. The concept of tolerance makes this algorithm robust and consistent. It enables to define 'on-boundary', which can be interpreted as either 'in-' or 'out-' side region, and to manage rounding errors in floating point computation. Also the tolerance is used as a measure of reliability of point classifications. The proposed algorithm is based on a ray-intersection technique known as the most efficient, in which intersections between a ray originating from a given test point and the boundary of a region are counted. An odd number of intersections indicates that the point is inside region. For practical examples the algorithm is most efficient because most edges of the polygon region are processed by simple bit operations.

**Key words** : Point classification, Ray-intersection method, Clipping, On-tolerance

#### 1. 서 론

'영역의 내외부 판별' 문제는 어떤 점이 닫힌 영역의 내부에 속하는지, 외부에 속하는지 판별하는 문제이며, 컴퓨터 그래픽스, CAD/CAM, 지리 정보 시스템 등의 다양한 분야에서 활용되는 기본적인 계산 기하(computational geometry) 문제다<sup>[1-3]</sup>. 본 연구에서는 2차원 실수 영역에서 정의되는 영역의 내외부 판별 문제를 다루고자 한다.

많은 기존의 연구는 주로 영역의 내부와 외부, 두 영역으로 검사점을 분류해 왔다. 본 논문에서는 '공차' 개념을 도입해서 내부, 외부, 경계의 세 영역으로 검사점을 분류하는 방법을 제시하고자 한다. 제시하는 방법은 다른 어떤 방법보다 강건한(robust) 알고리즘을 보장하는데, 이것은 공차 개념을 통해 얻어진다. 그리고, 일반적으로 가장 빠른 판별 방법인 직선 교차 방법을 쓰고 있기 때문에 계산(computing)의 효율성도 보장하

는 방법이다.

#### 1.1 관련 연구

기존의 연구는 크게 직선 교차 방법(ray-intersection or parity-count method), 각도 합계 방법(angle-summation or winding number method), 삼각형 팬 방법(triangle fan method), 격자 방법(grid method) 등의 네 가지로 나눌 수 있다<sup>[4-11]</sup>.

직선 교차 방법은 검사점을 시작점으로 하는 검사 직선(test ray)과 검사 영역과의 교차(crossing) 회수를 헤아리는 방법이다. 교차 회수가 홀수면 영역의 내부로, 짝수면 외부로 판별한다<sup>[4,9,11]</sup>.

각도 합계 방법은 판별 영역이 다각형 영역인 경우에 가능한 방법이다. 이 방법은 다각형의 모서리를 일관성 있는 방향으로 따라가면서, 검사점과 다각형의 모서리로 삼각형을 구성하고, 그 삼각형에서 검사점에서의 각들을 합한다. 각도 합이  $2\pi$ 의 배수이면 검사점은 내부 점이고, 영(zero)이면 외부 점이다<sup>[7,10]</sup>.

삼각형 팬 방법은 다각형을 하나의 꼭지점을 기준으로 방사형으로 뻗어나오는 삼각형들의 집합으로 보고, 모든 삼각형과 검사점의 내외부 판별결과를 합산

\*중신회원, 큐빅테크 큐빅기술연구소

\*\*정회원, 큐빅테크 큐빅기술연구소

- 논문투고일: 2001. 7. 6

- 심사완료일: 2001. 9. 10

하는 방법이다. 삼각형 팬 방법에서는 내부의 공유 모서리에 검사점이 걸칠 경우에 대한 특별한 처리가 요구되는데, 여기에서 오류가 개입될 가능성이 있다<sup>[7]</sup>.

격자 방법은 다각형을 감싸는 관리상자(majorizing box) 내부의 등간격 격자를 내부(fully inside)/외부(fully outside)/보류(intermediate)로 선행처리 한 후 이를 참조하는 방법이다. 격자 방법은 간단하고 강건한(robust) 방법이지만 실수 연산을 필요로 하는 분야에서는 적용하기 곤란하며, 그래픽스 분야에서 픽셀(pixel) 단위의 연산에 효과적인 방법이다<sup>[7]</sup>.

실수 영역에서 정의되는 다각형 영역의 내외부 판별에 사용할 수 있는 방법 가운데, 계산 시간이 가장 빠른 방법은 직선 교차 방법이다. 그러나, 하나의 다각형 영역에 대해서 여러 개의 점을 판별하는 경우에는 스와스 기법(swath method)이 가장 효율적인 것으로 평가되고 있다. 이 방법은 직선 교차 방법의 변형으로 주어진 다각형 영역에 대해서 미리 필요한 계산을 해두는 방식이다<sup>[3,7]</sup>.

### 1.2 직선 교차 방법의 문제점

직선 교차 방법은 개념적으로 명확한 방법이지만 실제 구현에서는 일관성 있게 교차 회수를 구하기가 쉽지 않다. 영역의 경계선과 검사 직선의 교점 계산을 통해서 교차 회수를 구하기 때문인데, 검사 직선이 다각형의 꼭지점을 지나거나 다각형의 모서리와 일치하는 경우에 교점의 정의가 불명확하기 때문이다.

Fig. 1의  $T_1$ 은 꼭지점이 검사 직선에 접촉하는 경우이며,  $T_2$ 는 꼭지점에서 만나는 경우이고,  $T_3$ 는 검사 직선이 모서리와 일치하는 경우이다. 특히, 이런 경우에 실수 연산의 라운드 오프 오류의 영향이 커지면서 교점 계산은 더욱 불안하게 된다. 또, 교점의 정의가 불명확하기 때문에 다각형을 이루는 경계의 시작점이 바뀌거나 방향이 바뀌는 경우에 서로 다른 판별 결과를 낳기도 한다. 가장 쉬운 접근 방법은 검사 직선의 방향

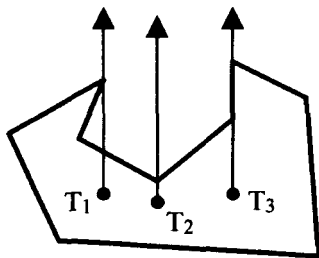


Fig. 1. Special cases of ray intersection.

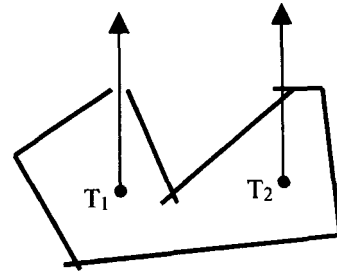


Fig. 2. Incomplete polygonal region.

을 바꾸어서 특수한 경우를 회피하는 방법이지만 라운드 오프 오류로 인해 완전한 회피를 보장하는 방향을 찾는 것도 쉽지 않다<sup>[9]</sup>. 이 문제를 해결하려는 다양한 시도들이 있었는데 실수 연산의 라운드 오프 오류를 제거하려는 연구들과 영역 경계의 방향 혹은 시작점과 무관하게 일관성 있는 판별 결과를 보장하는 방법에 관한 연구가 대부분이다<sup>[4,3,12]</sup>.

직선 교차 방법의 또 다른 문제는 영역의 정의가 불명확한 경우이다. Fig. 2와 같이 경계를 구성하는 모서리의 시작점과 끝점이 일치하지 않거나 단순 다각형(simple polygon)이 아닌 경우가 있다. Hui<sup>[8]</sup>는 모서리의 시작과 끝에 공차 영역(tolerance zone)을 적용해서 Fig. 2의  $T_1$ 과  $T_2$ 와 같은 경우에도 교점이 한번씩 계산되도록 하였으며, 별도의 거리 비교를 통해서 검사점이 영역의 경계에 있는 경우를 판별하였다.

### 1.3 본 논문의 목적

본 연구에서는 영역 내외부 판별 문제에 공차 개념의 적용을 통해 실수 연산의 라운드 오프 오류를 제어하고자 한다. 공차 개념을 통해 영역을 내부와 외부 및 경계의 세 가지로 구분하고, 영역의 내외부를 일관성 있게 판별하고자 한다. 실제로 경계로 분류되는 점들은 필요에 따라 내부 혹은 외부로 재 해석할 수도 있으며, 영역 판별의 신뢰성 평가의 잣대로 활용될 수도 있다.

영역을 내부와 외부 및 경계로 분류하는 개념은 몇몇 논문<sup>[1,13]</sup>에서 소개되었다. 그러나 2차원 실수 영역에서 정의되는 다각형 영역의 내외부 판별 문제에 공차의 개념을 적용한 실제 알고리즘을 제시한 논문은 발견하지 못했다.

본 연구에서 제안하는 방법은 직선교차 방법을 확장한 알고리즘이며, 전체적인 방법은 다각형 영역을 이루는 꼭지점을 분류하고, 분류된 꼭지점의 코드에 따라 인접한 꼭지점이 이루는 모서리의 교차 여부를 판

단하게 된다. 다루는 문제의 정의와 용어는 2장에서 설명하고, 제안하는 방법의 구체적인 내용을 3장과 4장에서 설명하겠다. 끝으로 5장에서 이 논문의 결론을 맺는다.

### 2. 문제 및 용어의 정의

본 연구에서 다루는 문제는 다각형 P와 검사점 T가 2차원 실수 공간에서 주어지고, 공차 e가 주어질 때, P와 e에 의해서 정의되는 '내부', '외부', '경계'의 세 영역 중 어디에 T가 존재하는지 결정하는 것이다. 이때, 다각형 P는 꼬임을 허용하는 임의의 형상이다. 기존의 연구에서 꼬임이 없는 단순 다각형(simple polygon) 영역을 가정하는 경우가 많은데, 실수 영역의 실제 문제들은 엄밀한 의미의 단순 다각형이 아닌 경우가 많기 때문에 꼬임을 허용하는 것이 타당하다. 재단곡면(trimmed surface)의 정의역 경계 곡선(domain boundary curve)의 경우에 정의에 의하면 꼬임이 없는 단순 곡선이지만<sup>[14,15]</sup>, 실제 데이터는 조금씩 꼬여 있거나 Fig. 2와 같이 시작과 끝점이 일치하지 않는 곡선의 조합으로 구성된다. 특히, 재단 곡면을 처리하는 CAD/CAM 시스템마다 허용하는 공차가 다르기 때문에 공차를 적용해도 단순 곡선으로 해석할 수 없는 경우가 많다. Fig. 2와 같은 모서리가 떨어진 경우에도 모서리를 연결하여 닫힌 영역을 구성하면 꼬임이 있는 영역으로 해석할 수 있다.

설명을 위해 다각형 P는 n개의 꼭지점으로 구성되며,  $V_1, V_2, \dots, V_n$ 으로 표현한다. 이때,  $V_1$ 과  $V_n$ 은 서로 다른 점이다. 그리고,  $V_{i-1}$ 과  $V_i$ 를 잇는 직선을 모서리(edge)라고 부르기로 한다. 그리고, 검사점 T를 시작점으로 하는 반직선을 검사 직선(test ray) R이라고 하자. 검사 직선은 계산의 효율성을 위해 통상 x-축 혹은 y-축과 평행한 직선을 채택하는 것이 좋는데, 본

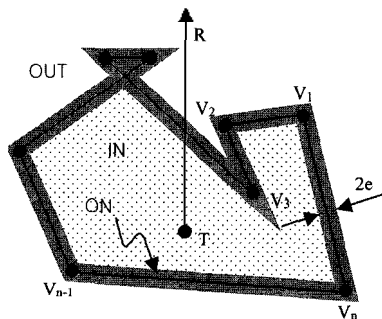


Fig. 3. Problem definition.

연구에서는 y-축과 평행한 직선을 채택했다. Fig. 3에 본 연구에서 다루는 문제를 도식적으로 나타내었다. 이때, 경계 영역의 폭은 공차의 두 배인 2e이다.

### 3. 꼭지점 분류

교차 회수를 구하기 위해 먼저 다각형을 이루는 꼭지점( $V_1, \dots, V_n$ )을 9개의 코드로 분류한다. 9개의 분류 코드는 각각의 꼭지점이 검사점 T를 기준으로 어떤 영역에 있는지 나타내며, Fig. 4는 9개의 분할 영역과 각 영역에 해당하는 분류 코드를 보여주고 있다. Fig. 4에서 CT에 해당하는 영역은 검사점 T를 중심으로 공차 e만큼의 상하좌우 폭을 가지는 사각 영역임을 알 수 있다. 아홉 개의 코드 값들은 비트(bit) 연산을 위해서 비트로 표현되어 있다.

꼭지점 분류의 과정은 그래픽스에서 흔히 사용하는 클리핑(clipping) 알고리즘인 'Cohen-Sutherland 알고리즘'<sup>[16]</sup>과 유사하다. C로 구현된 프로그래밍 코드가

LU (1010)	UP (1000)	RU (1001)	$y = T_y + e$
LT (0010)	CT (0000)	RT (0001)	
LD (0110)	DN (0100)	RD (0101)	$y = T_y - e$
$x = T_x - e$		$x = T_x + e$	

Fig. 4. Outcodes for vertex classification.

```

unsigned int VertexClassification(V, T, e)
{
    unsigned int outcode = CT;

    if (V.x > T.x + e) outcode |= RT;
    else if (V.x < T.x - e) outcode |= LT;

    if (V.y > T.y + e) outcode |= UP;
    else if (V.y < T.y - e) outcode |= DN;

    if (outcode == CT) {
        double d = Distance(V, T);
        if (d <= e) return CT;
        else if (V.y > T.y) return UP;
        else return DN;
    }

    return outcode;
}
    
```

Code 1. Vertex classification.

Code 1에 나타나 있는데, 꼭지점이 CT(중앙) 구간에 속하는 경우에만 다르게 처리됨을 알 수 있다. 이 함수에서 CT로 판정되면 검사점은 다각형의 꼭지점 위에 놓여 있다. Code 1의 함수에서 입력 인자 V는 다각형의 꼭지점에 해당하고, T는 검사점이며, e는 공차이다.

다각형을 이루는 하나의 꼭지점은 항상 한 모서리의 끝점이며 다음 모서리의 시작점이다. 그러나, 본 연구에서 제안하는 방법은 특정 꼭지점의 분류는 한 번만 필요하며, 여러 번 행하더라도 항상 같은 결과 값을 얻을 수 있다.

#### 4. 교차 회수 헤아리기

다각형과 검사 직선의 교차 회수는 다각형을 이루는 모서리 각각의 교차 회수를 모두 더하는 방식으로 구한다. 하나의 모서리는 두개의 꼭지점으로 구성되므로 모서리와 검사 직선과의 교차 여부는 3장에서 설명한 꼭지점 분류 코드를 이용해서 쉽게 판단할 수 있다. 두 꼭지점의 분류 코드가 같고 그 값이 CT(중앙)가 아니라면 그 모서리는 검사 직선과 교차하지 않는다는 것을 쉽게 알 수 있다. 그 값이 서로 다르더라도 L(좌), R(우), D(하)와 같은 성분이 동일하게 들어 있는 경우에도 교차하지 않는다는 것을 쉽게 알 수 있다. Fig. 5에서 (a)는 분류 코드가 같은 예이다. (b)는 동일한 성분 L이 들어 있어서 해당 모서리가 검사 직선의 왼쪽에 있음을 알 수 있다. 또, (c)의 경우는 동일한 성분 D가 있어서 아래쪽을 지나는 모서리임을 알 수 있다.

두 꼭지점의 분류 코드 값이 서로 다르고 UP(상) 성분이 동일하게 들어 있는 경우에는 교차에 해당하지만, Fig. 1과 같이 특수한 경우에 인접한 모서리간에 교차 횟수가 중복적으로 계산될 수 있다. 그러나, 모서리의 한 꼭지점이 엄밀히 검사 직선의 왼쪽에 있고 다른 한 꼭지점은 검사 직선 위에 있거나 오른쪽에 있을 때만 교차라고 정의하면 특수한 경우를 모두 고려할

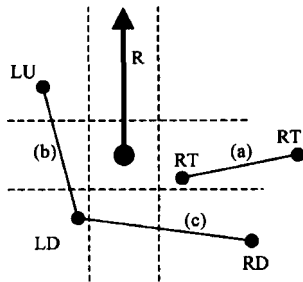


Fig. 5. Non-crossings.

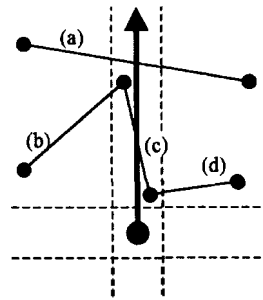


Fig. 6. Crossings (a, b) and non-crossings (c, d).

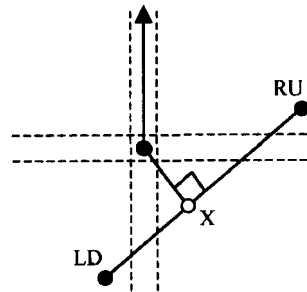


Fig. 7. Test crossing.

수 있다<sup>2)</sup>. 즉, 한 꼭지점의 분류 코드가 LU(좌상)이고, 다른 한 꼭지점의 분류 코드는 RU(우상) 혹은 UP(상)인 경우에만 그 모서리가 교차하는 것으로 판단한다. Fig. 6에서 (a)와 (b)는 한쪽 꼭지점이 LU이기 때문에 교차로 판단하지만, (c)와 (d)의 경우는 두 꼭지점이 모두 LU가 아니므로 교차로 판단하지 않는다. 이러한 방법을 적용하면 Fig. 6의 (b), (c), (d)가 서로 인접한 모서리이며, 꼭지점이 검사 직선 위에 있지만 교차 횟수는 한번으로 헤아려 진다는 것을 알 수 있다.

Fig. 7과 같이 두 꼭지점의 분류 코드 값이 동일하지 않으면서 어느 한쪽은 UP성분을 갖지 않는 경우에는 별도의 제산을 해야만 실제 교차 여부를 판단할 수 있다. 이러한 경우에는 검사점 T와 가장 가까운 모서리 위에 있는 '최단 거리 점'을 구한다. 최단 거리 점은 대부분의 경우 Fig. 7의 점 X와 같이 T에서 모서리에 수선을 내린 점이다. 그러나 수선의 발이 모서리(무한 직선이 아닌 선분) 위에 있지 않은 경우에는 모서리의 양 끝점 가운데 T에서 가까운 점이 최단 거리점이다. T에서 얻어진 점까지의 거리가 공차 e보다 짧으면 검사점 T는 다각형 P의 '경계'에 놓이는 것으로 판정한다. T에서의 거리가 공차 e보다 큰 경우에는 최단 거리 점의 y값이 T의 y값 보다 큰 경우에만 교차

```

#define UD (UP|DN) // 1100
#define LR (L|R) // 0011

int CountingCrossings (P, T, e)
{
    count = 0
    sCode = VertexClassification (Vn, T, e);
    if (sCode == CT)
        return -1;

    for (int i = 1; i <= n; i++) {
        eCode = VertexClassification (Vi, T, e);
        if (eCode == CT)
            return -1;

        aCode = sCode & eCode;
        oCode = sCode | eCode;
        if (aCode == CT) {
            if ((oCode == UD) || (oCode == LR))
                return -1; // ON
            else {
                X = NearestPt(Vi, Vj, T);
                if (Distance(mp, T) <= e)
                    return -1; // ON
                else if (mp.y > T.y
                    && (oCode & LT) == LT)
                    count++;
            }
        }
        else if (aCode == UP) {
            if ((oCode & LT) == LT)
                count++;
        }
        sCode = eCode;
    }
    return count;
}

```

Code 2. Counting crossings.

로 판정한다. 따라서 Fig. 7의 경우는 교차가 아닌 것으로 판정된다.

Code 2은 모서리와 검사 직선과의 교차 여부 판별 방법을 다각형의 모든 모서리에 적용해서 다각형 P와 검사 직선 R과의 교차 회수를 헤아리는 프로그램의 개략적인 예를 보여주고 있다. 특히 Code 2는 교차의 다양한 경우들이 비트로 표현된 꼭지점 분류 코드의 비트 연산(OR와 AND)을 통해서 쉽게 판별됨을 보여주고 있다.

## 5. 결 론

본 연구에서는 2차원 실수영역에 정의된 다각형 영역의 내부와 외부 및 경계 영역을 판별하는 안정적이고 효율적인 알고리즘을 제시하였다. 기존의 방법들은 영역을 내부와 외부로 구분하였기 때문에 영역의 경계 혹은 영역을 이루는 꼭지점에 가까운 점들의 판별에 일관성 없거나 안정성이 떨어지는 경우가 많았다. 그러나 제안하는 방법은 '공차'의 개념을 통해 '경계' 영역을 정의할 수 있었으며, 실수 연산의 라운드 오프 오

류를 제어할 수 있었다. '경계' 영역은 용도에 따라 '내부' 혹은 '외부'로 재해석 할 수 있으며 영역 판별의 신뢰성 평가의 잣대로 활용할 수도 있다. 제안된 방법은 영역 판별 방법 가운데 가장 효율적인 방법으로 알려진 직선 교차 방법을 확장한 알고리즘이며, 직선 교차 방법과 같은 계산 복잡도를 가진다. 그리고 실제 예제들의 경우 대부분의 다각형 모서리들이 간단한 비트 연산으로 걸러지므로 한 점을 판별하는 방법으로는 계산 효율을 극대화시켰다고 볼 수 있다.

추후 연구 과제로는 주어진 다각형 영역을 여러 번 판별해야 하는 경우 최적화하는 방법이 필요하며, 공차가 라운드 오프 오류와 비슷하거나 작은 경우의 특성을 분석할 필요가 있다. 또, 영역이 다각형이 아닌 곡선인 경우를 고려해 볼 수 있겠다.

## 참고문헌

1. Mortenson, M. E., *Geometric Modeling*, John Wiley & Sons, 1985.
2. O'Rourke, Joseph., *Computational Geometry in C (2<sup>nd</sup> ed.)*, Cambridge University Press, 1994.
3. Preparata, F. P. and Shamos, M. I., *Computational Geometry-An Introduction*, Springer-Verlag, New York, 1985.
4. Burton, F. W., Kollias, V. J. and Kollias, J. G., Consistency in point-in-polygon tests, *The Computer Journal*, Vol. 27, No. 4, pp. 375-376, 1984.
5. Feito, F., Torres, J. C. and Urena, A., Orientation, simplicity and inclusion test for planar polygons, *Computer and Graphics*, Vol. 19, No. 4, pp. 595-600, 1995.
6. Haines, E., Point in Polygon Strategies, *Graphic Gems*, Academic Press, pp. 24-35, 1994.
7. Huang, C. W. and Shih, T. -Y., On the complexity of point-in-polygon algorithms, *Computers and Geosciences*, Vol. 23, No. 1, pp. 109-118, 1997.
8. Hui, K. C., A robust point inclusion algorithm for regions bounded by parametric curve segments, *Computer-Aided Design*, Vol. 29, No. 11, pp. 771-778, 1997.
9. Taloy, G., Point in polygon test, *Survey Review* 1994, Vol. 32, No. 254, pp. 479-484.
10. Weiler, K., An Incremental Angle Point in Polygon Test, *Graphic Gems*, Academic Press, pp. 16-23, 1994.
11. Wu, H., Gong, J., Li, D. and Shi, W., An algebraic algorithm for point inclusion query, *Computers & Graphics*, Vol. 24, pp. 517-522, 2000.
12. Yamaguchi, F., Niizeki, M. and Fukunaga, H., Two robust point-in-polygon tests based on the  $4 \times 4$  determinant method. *Proceedings of ASME Design Technical Conference*, 16-th Design Automation Conference, pp. 89-95, 1990.

13. Fang, S., Bruderlin, B. and Zhu, X., Robustness in Solid Modeling: a tolerance-based intuitionistic approach, *Computer-Aided Design*, Vol. 25, No. 9, 1993, pp.567-576.
14. Choi, B. K. and Robert, B., Jerard, *Sculptured Surface Machining - Theory and Applications*, Kluwer academic publishers, 1998.
15. *The Initial Graphics Exchange Specification (IGES) Version 5.2*, IGES/PDES Organization, 1993.
16. Foley, J. Van, D. A., Feiner, S. and Hughes, J., *Computer Graphics: Principles and Practice (2<sup>nd</sup> ed.)*, Addison Wesley, 1990.

---

### 정 연 찬

1989년 한양대학교 산업공학과 학사  
 1991년 KAIST 산업공학과 석사  
 1996년 KAIST 산업공학과 박사  
 1998년 DaimlerChrysler 연구원  
 1991년. 현재 큐빅테크 연구소 부소장  
 관심분야: 가공경로생성, 형상모델링, 컴퓨터 그래픽스, 시스템 통합

---

### 박 준 철

1993년 서울대학교 산업공학과 학사  
 1995년 KAIST 산업공학과 석사  
 1995년. 현재 큐빅테크 연구소 선임연구원  
 관심분야: 형상모델링, 솔리드모델링, 데이터 교환

---