평면상의 점들에 대한 조각적 이차 다항식 곡선 맞추기

(Fitting a Piecewise-quadratic Polynomial Curve to Points in the Plane)

김 재 훈 †
(Jae-Hoon Kim)

요 약 본 논문에서 우리는 평면상에 점들이 주어지는 경우에, 조각적 이차 다항식 곡선으로 맞추는 문제를 다룬다. 곡선은 이차 다항식 선분들로 이루어지고, 하나의 선분은 두 점 사이를 연결한다. 하지만 이 곡선은 점들의 부분집합만을 지나고, 지나지 못하는 점들에 대해서는 L^{∞}거리로 에러를 측정한다. 이 문제에 대해서 우리는 두 가지 최적화 문제를 생각한다. 첫째로 허용 가능한 에러의 범위가 주어지고, 곡선 선분의 개수를 줄이는 문제이고, 둘째로 선분의 개수가 주어지고, 에러를 줄이는 문제이다. 주어진 점들의 개수 n에 대해서, 우리는 첫번째 문제에 대한 $O(n^2)$ 알고리즘과 두번째 문제에 대한 $O(n^3)$ 알고리즘을 제안한다.

키워드 : 이차 다항식, 맞추기, L^{∞}거리, 최적화

Abstract In this paper, we study the problem to fit a piecewise-quadratic polynomial curve to points in the plane. The curve consists of quadratic polynomial segments and two points are connected by a segment. But it passes through a subset of points, and for the points not to be passed, the error between the curve and the points is estimated in L^{∞} metric. We consider two optimization problems for the above problem. One is to reduce the number of segments of the curve, given the allowed error, and the other is to reduce the error between the curve and the points, while the curve has the number of segments less than or equal to the given integer. For the number n of given points, we propose $O(n^2)$ algorithm for the former problem and $O(n^3)$ algorithm for the latter.

Key words : quadratic polynomial, fitting, L^{∞} metric, optimization

1. 서 론

본 논문에서는 평면상에 n개의 점 $p_i = (x_i, y_i)$, $x_1 < x_2 < \cdots < x_n$, 들의 집합 D가 주어질 때, 이 점들을 잘 맞추는 조각적 이차 다항식 곡선(piecewise-quadratic polynomial curve)을 찾는 문제를 생각한다. 이 문제는 CAD/CAM, CAGD(Computer Aided Geometric Design), 컴퓨터 그래픽스 등의 분야에서 자주 나타난다. 이전 연구들에서는 조각적 일차 다항식 곡선, 즉, 직선 선분들을 가지는 곡선만을 다루었다. 본 논문에서는 각 선분들이 이차 다항식으로 이루어진 곡선을 다룰 것이다.

조각적 이차 다항식 맞추기 곡선을 찾는 문제에서 우리는 두 가지 최적화 문제를 생각할 것이다. 첫째로 곡선이 지나는 점들의 개수를 줄이는 문제를 생각한다. 곡선의 각 선분은 주어진 점들을 지나는데 이 지나는 점들의 개수를 최소화하는 것이다. 이것은 조각적 다항식 곡선의 다항식 선분의 개수를 줄이는 것과 같다. 우리는 곡선이 지나지 못하는 점들과 곡선 사이의 에러를 L^{∞} 거리로 측정한다. 에러 범위 ε이 입력으로 주어지고, 선분의 개수를 최소화하면서 지나지 못하는 점들과 에러 범위 ε 안에서 지나는 곡선을 찾을 것이다. 이 문제를 min-# 문제라고 부를 것이다. 둘째로, 우리는 선분의 개수 k가 입력으로 주어지고, k보다 작거나 같은 개수

† 종신회원 : 부산외국어대학교 컴퓨터공학부 교수
jhoon@pufs.ac.kr
논문접수 : 2008년 6월 20일
심사완료 : 2008년 12월 10일

정보과학회논문지: 시스템 및 이론 제36권 제1호(2009.2)

의 선분을 가지는 곡선 중에 점들과 곡선 사이의 에러를 최소화 하는 곡선을 찾을 것이다. 이 문제를 min-ε 문제라고 부를 것이다.

1.1 관련연구

평면상에 주어진 점들을 조각적 다항식 곡선에 의해서 맞추는 문제는 CAD/CAM, 컴퓨터 그래픽스와 같은 응용분야에서 잘 알려진 문제이다. 하지만 계산기하학 분야에서는 알고리즘의 시간 복잡도 계산과 같은 분석을 통해 다루었고, 대부분은 조각적 일차 다항식 곡선, 다시 말해, 직선 선분들로 이루어진 일종의 경로들을 다루었다. 이 경로와 점 들 사이의 에러를 L^∞ 또는 L^2 거리로 측정하고, 이차원 또는 삼차원 공간에서의 문제들을 다루었다.

우선 [1]에서 이차원 평면 상에서 L^∞ 거리로 min-# 문제에 대해서 연구하였고 $O(n^2)$ 알고리즘을 제안하였다. 또한 min-ε 문제에 대해서는 $O(n^2 \log n)$알고리즘을 제안하였다. [2]에서는 직선 선분들이 y축에 평행한 직선과 한 점에서 만나는 경로를 단조경로라 하는데 이 특별한 경우에 대해서, min-#과 min-ε 문제에 대해서 각각 $O(n^{4/3+\delta})$ 알고리즘을 제안하였다.

이차원 평면의 L^2 거리에 대해서는 [3]에서 min-#과 min-ε 문제에 대해서 각각 $O(n^2 \log n)$과 $O(n^2 \log^2 n)$ 알고리즘을 제안하였고, 이후에 [4]에서 저자들은 시간을 각각 $O(n^2)$과 $O(n^2 \log n)$으로 줄였다.

삼차원 공간에서는, min-#과 min-ε 문제에 대해서 각각 $O(n^2 \log n)$과 $O(n^2 \log^3 n)$ 시간을 가지는 L^2 거리의 결과가 존재하고, L^∞ 거리에 대해서는 시간이 각각 $O(n^2)$과 $O(n^2 \log n)$으로 축소된다[5].

이상의 이전 연구들은 모두 일차 다항식 곡선들만을 다루었다. 실제 CAD/CAM, 컴퓨터 그래픽스등과 같은 응용분야에서 사용되는 곡선들은 대부분 이차 이상의 다항식 곡선들이 사용된다. 따라서 이차 이상의 다항식 곡선들로 점들을 맞추는 알고리즘을 분석하는 것이 의미 있는 일이며, 이 논문에서는 이차 다항식 곡선으로 점들을 맞추는 문제를 다룰 것이다.

2. 이차 다항식 곡선 맞추기

2.1 min-# 문제

우리는 먼저 min-# 문제를 다룰 것이다. 평면상의 점들의 집합 D와 점들과 곡선 간의 에러 ε이 입력으로 주어질 때, 조각적 이차 다항식 곡선 $p(x)$는 주어진 점들의 부분집합을 지나고 점들과의 에러가 ε보다 작거나 같아야 한다. 다시 말해서, 어떠한 부분집합 S⊆D가 존재해서 곡선 $p(x)$는 다음을 만족한다.

$$y_i = p(x_i),\ \forall p_i \in S, \tag{1}$$

$$|y_i - p(x_i)| \le \varepsilon,\ \forall p_i \in D-S. \tag{2}$$

문제 min-#는 위 식 (1)과 (2)를 만족하는 곡선들이 지나는 부분집합 S의 최소 크기를 찾는 것이다. 다시 말해서, 곡선의 최소 선분 개수를 찾는 것이다.

2.1.1 지름길 그래프 생성

우리는 min-# 문제를 풀기 위해서 그래프를 이용한다. 각 점들은 그래프의 노드에 대응하고, 두 점 사이를 연결하는 곡선의 선분이 두 점 사이의 점들을 에러 범위 안에서 지날 때, 이 두 점에 대응하는 노드들을 에지로 연결한다. 이렇게 만들어진 그래프를 지름길 그래프(shortcut graph)라고 부를 것이다. 우리는 문제 min-#를 풀기 위해서 지름길 그래프에서 점 p_1과 p_n에 대응되는 노드간에 최단 경로를 찾는다. 이 최단 경로의 길이가 선분의 최소 개수와 같다. 따라서 우리는 먼저 지름길 그래프를 어떻게 만들 것이지 생각할 것이다.

편의상으로, p_i를 주어진 점뿐만 아니라 대응되는 지름길 그래프의 노드를 나타낸다고 하자. 임의의 노드 p_i로부터 연결되는 에지들을 생각할 것이다. 점 p_i를 지나는 이차 다항식 곡선 선분을 $f(x)=ax^2+bx+c$라고 하자. 그러면 $y_i=f(x_i)$을 만족하므로, $f(x)=a(x^2-x_i^2)+b(x-x_i)+y_i$라고 할 수 있다. 노드 p_j로의 에지를 생각하는 경우에, 노드 p_i와 p_j 사이에 에지가 존재하기 위해서, 점 p_i와 p_j 사이의 모든 점 p_k에 대해서 다음을 만족한다.

$$(y_k - y_i) - \varepsilon \le a(x_k^2 - x_i^2) + b(x_k - x_i) \le (y_k - y_i) + \varepsilon \tag{3}$$

위 식 (3)에서 a와 b를 미지수로 생각할 때, (a,b) 평면상의 회랑(corridor)에 대응된다. 회랑이란 평행한 두 직선 사이의 공간을 말한다. 사이의 모든 점 p_k에 대해서 위 식을 만족해야 함으로, 위 식으로 만들어 지는 모든 회랑들의 교집합이 존재해야 한다.

위 부등식 (3)에 대응하는 회랑들의 교집합이 존재하는 지를 효율적으로 풀기 위해서, 회랑이 존재하는 (a,b) 평면의 쌍대(dual) 평면을 생각한다. 다시 말해서, (a,b) 평면상의 점과 직선을 각각 쌍대 평면상의 직선과 점으로 변환한다. 따라서, (a,b) 평면상의 평행한 직선들의 집합인 회랑은 쌍대 평면상의 수직 선분으로 변환된다. 구체적으로, (a,b) 평면상의 회랑의 테두리 직선들 $b=-(x_k+x_i)a+\dfrac{(y_k-y_i)\pm\varepsilon}{(x_k-x_i)}$은 쌍대 평면상의 점 $(\alpha,\beta)=\left(x_k+x_i, -\dfrac{y_k-y_i\pm\varepsilon}{x_k-x_i}\right)$로 변환된다. 따라서 회랑은 이 두 점들을 끝점으로 가지는 수직 선분으로 변환된다. 그리고 이 수직 선분들은 α좌표 값이 증가함을 알 수 있다.

또한 쌍대 변환에 의해서 (a, b)평면상의 점들은 직선

으로 변환되는데, 회랑들의 교집합에 속하는 점들은 쌍대 평면 (α, β)에서 직선에 대응되고 이 직선들은 회랑의 수직 선분을 모두 지나야 한다. 따라서 회랑의 교집합에 속하는 점들은 쌍대 평면의 수직 선분들을 모두 지나는 직선들에 대응된다.

다항식 곡선 $f(x)$는 끝점 p_j를 지나야 함으로, $a(x_j^2 - x_i^2) + b(x_j - x_i) - (y_j - y_i) = 0$라는 직선을 얻고, 이 직선은 위의 회랑들의 교집합을 지나야 한다. 우리는 이 조건을 쌍대 평면상의 조건으로 바꿀 수 있는데, 위 직선은 쌍대 평면상의 한 점으로 변환되고 이 점은 수직 선분들을 모두 지나는 직선과 만나야 한다. 여기서 우리는 점의 α 좌표 값이 수직 선분들의 α 좌표 값보다 큼에 주목한다.

따라서 우리는 쌍대 평면상의 수직 선분들이 주어지고, 이 선분들보다 큰 α 좌표 값을 가지는 점이 주어질 때, 수직 선분들 모두를 지나고 주어진 점을 지나는 적어도 하나의 직선이 존재하는지 테스트하는 문제를 생각할 수 있다. 이 직선이 존재하면 노드 p_i에서 p_j로 에지를 연결한다.

위의 설명을 종합해보면, 노드 p_i에서 출발하는 에지들의 존재를 테스트하기 위해서, 노드 p_j, $j > i$ 들을 순차적으로(incrementally) 고려한다. 노드 p_j로의 에지의 존재 여부가 막 결정되었다고 가정하자. 다음 노드 p_{j+1}로의 에지를 생각할 때, 곡선이 점 p_j에서 에러 범위 안에 지난다는 조건으로 만들어 지는 회랑이 추가되고, 이는 쌍대 평면에서 수직 선분이 추가됨을 말한다. 그리고 곡선이 점 p_{j+1}를 지난다는 조건의 직선은 쌍대 평면에서의 점에 대응되고 수직 선분들을 모두 지나고 이 점을 지나는 직선이 존재하는지 테스트함으로써, 노드 p_{j+1}로의 에지의 존재여부를 결정한다.

그럼 수직 선분들이 순차적으로 주어질 때, 점이 주어지면 선분들을 모두 지나고 점을 지나는 직선이 존재함을 알 수 있는 알고리즘을 설명한다. 우선 v_j, $j = i+1,\ldots,n$, 를 주어지는 수직 선분들이라고 하고, 선분의 위 끝점을 u_j, 아래 끝점을 l_j로 나타낸다고 하자. 우리는 점 u_j 들로 이루어진 upper support hull과 점 l_j 들로 이루어진 lower support hull 을 각각 deque Q_u와 Q_l로 유지한다. 또한 우리는 수직 선분 v_j들을 모두 지나는 직선 중 기울기가 가장 큰 것과 작은 것을 나타내는 두 직선 t와 t'을 유지한다(그림 1).

초기에 deque Q_u의 헤드에는 u_{i+2}, 꼬리에는 u_{i+1}, deque Q_l의 헤드에는 l_{i+2}, 꼬리에는 l_{i+1}가 저장된다. 또한 직선 t는 점 u_{i+2}와 l_{i+1}를 지나고 직선 t'는 점 u_{i+1}와 l_{i+2}를 지난다. 그러면 수직 선분 v_j가 주어진 경우를 생각해 보자. 우선 u_j가 직선 t'의 아래에 있거나, l_j가 직선 t의 위에 있는 경우는 t과 t'를 고치지 않는다. 이 경우가 아니면, 위 끝점 u_j가 직선 t의 아래에 있는 경우에, 우선 upper support hull을 고친다. deque Q_u의 헤드에 저장된 점이 헤드 바로 뒤에 저장된 점과 u_j를 연결한 선위에 있다면 헤드를 pop 한다. 이 과정을 반복한 후, deque Q_u에 점 u_j를 push 한다. 그리고 deque Q_l의 꼬리에 저장된 점과 u_j를 지나는 선을 생각해서, Q_l의 꼬리 바로 앞에 저장된 점이 이 선위에 있으면 Q_l의 꼬리에 저장된 점을 pop 한다. 이 과정을 반복한 후, Q_l의 꼬리에 저장된 점과 u_j를 지나는 선을 t로 지정한다. 아래 끝점 l_j가 직선 t'의 위에 있는 경우에는 위의 t

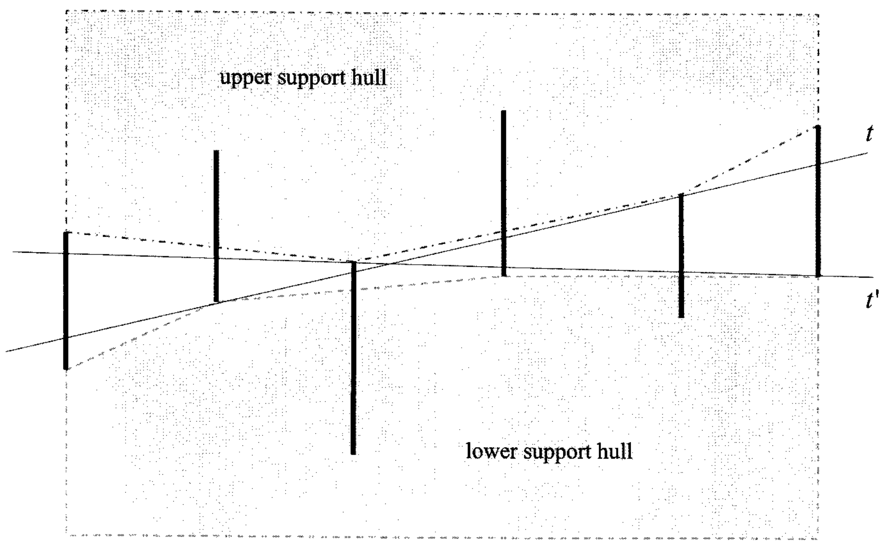


그림 1 평면상의 수직 선분들과 직선 t와 t'

를 고치는 경우와 비슷하게 t'를 고칠 수 있다.

위와 같이 직선 t와 t'를 유지해 가면서, 점이 주어지고 이 점이 t의 아래에 있고 t'의 위에 있다면, 수직 선분들을 모두 지나고 이 점을 지나는 직선이 존재함을 나타낸다. 따라서 노드 p_i로부터 연결되는 에지의 존재여부를 결정하기 위해서 점들 v_j, $j = i+1, ..., n$을 순차적으로 고려하면서, 위와 같이 직선 t와 t'를 유지하고 주어진 점이 두 직선 사이에 있는지 테스트하는데 총 $O(n)$ 시간이면 충분하다. 따라서 모든 노드 로부터 연결되는 에지의 존재여부를 결정하기 위해서는 총 $O(n^2)$ 시간이 걸릴 것이다.

보조정리 1. 평면상의 n개의 점들이 주어지는 경우에, 지름길 그래프를 만드는데 $O(n^2)$이면 충분하다.

2.1.2 min-# 알고리즘

우리는 이전 2.1.1 절에서 지름길 그래프를 생성하는 방법에 대해서 설명하였다. 이 절에서는 지름길 그래프를 이용하여 min-# 문제를 해결할 것이다. 따라서 min-# 문제를 해결하는 알고리즘을 다음과 같이 종합적으로 정리할 수 있다.

1) 곡선 선분이 두 점을 지나고 사이의 점들을 에러 범위 안에서 지나면 두 점에 대응하는 노드간에 에지를 연결하여 생성하는 지름길 그래프를 만든다.

1-1) 지름길 그래프의 임의의 에지의 존재여부를 결정하기 위한 문제를 평면상의 회랑들의 교집합과 한 직선의 교차여부를 결정하는 문제로 치환한다.

1-2) 1-1)의 문제를 쌍대 평면으로 변환하여 수직 선분들을 모두 지나는 직선과 한 점의 교차여부를 결정하는 문제를 푼다.

2) 지름길 그래프의 노드 p_1에서 p_n으로 최단 경로의 길이를 구한다.

2.1.1절에서 우리는 1-2)의 문제를 $O(n^2)$ 시간에 풀 수 있음을 보였다. 따라서 지름길 그래프를 $O(n^2)$ 시간에 생성할 수 있다. 결과적으로 다음 정리에서 min-# 문제를 푸는 시간 복잡도를 구할 수 있다.

정리 1. 평면상의 n개의 점들이 주어지는 경우에, 문제 min-# 를 $O(n^2)$ 시간에 풀 수 있다.

증명. 보조정리 1에 의해서 우리는 $O(n^2)$ 시간에 지름길 그래프를 만들 수 있다. 이 지름길 그래프에서 단순히 너비우선 탐색(breadth-first search)을 함으로서, 노드 p_1에서 p_n으로 최단 경로를 구할 수 있다. 따라서 총 시간은 $O(n^2)$으로 충분하다.

2.2 min-ε 문제

이 절에서는 양의 정수 $k>0$가 주어지고, 선분의 개수가 k보다 작거나 같은 조각적 이차 다항식 곡선에 대해서, 점들과의 에러가 최소가 되는 곡선을 찾는 문제를 다룬다.

우리는 점 p_i와 p_j 사이에 에러 상수 ε_{ij}를 정의한다. 에러 상수 ε_{ij}는 다음과 같이 정의된다.

$$\varepsilon_{ij} = \min_{f} \max_{i<k<j} |y_k - f(x_k)|.$$

여기서, 최소값은 점 p_i와 p_j를 지나는 임의의 이차 다항식 $f(x)$에 대해서 구한다. 다시 말해서, 에러 상수 ε_{ij}는 점 p_i와 p_j를 지나는 이차 다항식 $f(x)$가 가질 수 있는 에러의 최소값을 말한다. 우리는 각각의 점 p_i와 p_j 쌍에 대해서 에러 상수 ε_{ij}를 계산할 것이다. 이차 다항식 $f(x)$는 점 p_i와 p_j를 지나므로, $f(x)$는 다음과 같이 하나의 미지수 a의 함수로 쓸 수 있다.

$$f(x)= ax^2 + g(a)x + h(a)$$

여기서, g와 h는 a의 일차 함수이다. 따라서, 에러 상수는 $\varepsilon_{ij} = \min_{a} \max_{i<k<j} |y_k - f(x_k)|$와 같이 바꿔 쓸 수 있다. 이것은 다음과 같이 변수 a, b와 $2(j-i-1)$개의 제약(constraint)들로 이루어진 선형계획법(LP, Linear Programming)으로 나타낼 수 있다.

$$\begin{aligned} &\min\ b \\ s.t.\quad & y_k - f(x_k) \le b, \\ & -(y_k - f(x_k)) \le b,\ \ \forall k, i<k<j. \end{aligned}$$

이 LP 문제는 $O(n)$ 시간에 풀 수 있다는 것이 알려져 있다[6]. 따라서 우리는 에러 상수 ε_{ij}를 $O(n)$ 시간에 계산할 수 있고, 모든 점 p_i와 p_j 쌍에 대해서, ε_{ij}를 계산하는데 총 $O(n^3)$ 시간이면 충분하다.

그러면, 우리는 min-ε 문제의 답이 되는 최소의 에러 ε의 후보(candidate)들로 ε_{ij}들을 가짐을 알 수 있다. 따라서, 2.1절에서 풀었던 min-# 문제의 알고리즘을 사용해서 모든 ε_{ij}에 대해서 이진 검색(binary search)을 수행한다. 다시 말해서, ε_{ij}을 크기 순으로 정렬시킨 후, 이것을 $\varepsilon_1 \le \cdots \le \varepsilon_m$으로 나타낸다고 하자. 중간의 ε_l를 선택해서 min-# 문제를 풀어서 선분의 개수가 k 보다 작거나 같으면, $\varepsilon_l \ge \varepsilon$을 알 수 있고, 선분의 개수가 k 보다 크면, $\varepsilon_l \le \varepsilon$ 임을 알 수 있다. 따라서, 이진 검색으로 답을 찾을 수 있고, 총 $O(n^2 \log n)$ 시간이면 충분하다.

정리 2. 평면상의 n개의 점들이 주어지는 경우에, 문제 min-ε 을 $O(n^3)$ 시간에 풀 수 있다.

3. 결 론

우리는 본 논문에서 평면에 점들이 주어지고, 조각적 이차 다항식 곡선으로 맞추는 문제를 연구하였다. 특별히 두 가지 최적화 문제를 다루었는데, 첫번째로 에러의 범위 ε이 주어질 때, 점들과 곡선의 L^∞ 거리가 ε보다 작거나 같으면서 곡선의 최소 선분을 구하는 min-# 문제이고, 두번째로 곡선의 선분의 개수 $k>0$가 주어질

때, 점들과 곡선의 L^{∞} 거리의 최소값을 찾는 min-ε 문제이다. 이 문제들은 CAD/CAM, 컴퓨터 그래픽스 등에 응용될 수 있는데, min-# 문제는 최소 선분의 곡선을 찾아서, 곡선의 저장 공간을 최소화 할 수 있고, min-ε 문제는 점들을 잘 맞추는 근사 곡선을 찾을 수 있다.

우리는 min-#와 min-ε 문제에 대해서, 각각의 알고리즘을 제안하였고, 시간 복잡도 분석을 통해서 각각 $O(n^2)$과 $O(n^3)$ 시간을 가짐을 보였다.

참 고 문 헌

[1] S.L. Hakimi and E.F. Schmeichel. "Fitting polygonal functions to a set of points in the plane," *Computer Vision, Graphics, and Image Processing*, Vol.53, 132-136, 1991.

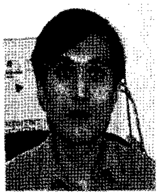
[2] K.R. Varadarajan. "Approximating monotone polygonal curves using the uniform metric," In Proc. of 12th ACM Symposium on Computational Geometry, 311-318, 1996.

[3] H. Imai and M. Iri. "Computational-geometric methods for polygonal approximations of a curve," *Computer Vision, Graphics, and Image Processing*, Vol.36, 31-41, 1986.

[4] W.S. Chan and F. Chin. "Approximation of polygonal curves with minimum number of line segments or minimum error," *Int. J. of Computational Geometry and Applications*, Vol.6(1), 59-77, 1996.

[5] G. Barequet, D.Z. Chen, O. Daescu, M.T. Goodrich, and J. Snoeyink. "Efficiently approximating polygonal paths in three and higher dimensions," In Proc. of 14th ACM Symposium on Computational Geometry, 317-326, 1998.

[6] F.P. Preparata and M.I. Shamos. "Computational geometry: An introduction," Springer-Verlag, 1985.



김 재 훈

1990년 3월~1994년 2월 서강대학교 수학과(학사). 1994년 3월~1996년 2월 KAIST 수학과(석사). 1996년 9월~2003년 2월 KAIST 전산학과(박사). 2003년 3월~현재 부산외국어대학교 컴퓨터공학부 조교수. 관심분야는 온라인 알고리즘, 스케줄링, 계산기하학