

## An Efficient Method for Minimum Distance Problem between Shapes Composed of Circular Arcs and Lines

金 鍾 敏\*·金 民 煥\*\*  
(Jong-Min Kim·Min-Hwan Kim)

**Abstract**—Generally, to get the minimum distance between two arbitrary shapes that are composed of circular arcs and lines, we must calculate distances for all the possible pairs of the components from two given shapes. In this paper, we propose an efficient method for the minimum distance problem between two shapes by using their structural features after extracting the reduced component lists which are essential to calculate the minimum distance considering the relationship of shape location. Even though the reduced component lists may contain all the components of the shapes in the worst case, in the average we can reduce the required computation much by using the reduced component lists. This method may be effectively applied to calculating the minimum distance between two shapes which are generated by the CAD tool, like in the nesting system.

**Key Words** : Nonconvex Polygon(비볼록 다각형), Convex Vertex(볼록 꼭지점), The Minimum Distance Problem(최소 거리 문제), Euclidean Distance (유클리디안 거리), The Nesting System(네스팅 시스템)

### 1. 서 론

Computational geometry[1, 2]는 여러가지 기하학적 문제(geometry problem)에 대해 그 문제를 해결하고자 하는 알고리즘의 설계, 분석 그리고 계산이 복잡도(Computational Complexity) 등을 다루는 분야이다. Computational geometry에서 다루는 여러가지 문제중에서 거리를 다루는 문제는 많은 관심과 연구의 대상이 되어 왔다[3-8]. 이 분야에서의 연구 대상을 살펴보면, 2차원 평면위에 있는 점들 중에서 최소거리를 갖는 두 점을 찾는 문제[4], 두 다각형의 꼭지점간의 최대거리를 구하는 문제[8], 두 다각형간의 최소거리를 구하는 문제[6, 7]등을 예로 들 수 있다.

두 다각형간의 최소거리문제는 거리계산의 대상이 되는 다각형의 종류에 따라 두가지의 경우

로 나누어 연구되었는데, 두 볼록 다각형(convex polygon)간의 경우와 볼록 다각형과 비볼록 다각형(nonconvex polygon)간의 경우이다. 각각 n개의 직선으로 이루어진 두 개의 볼록 다각형인 경우의 연산시간은  $O(\log n)$ 으로 증명되었으며[7], n개의 직선으로 이루어진 볼록 다각형과 m개의 직선으로 이루어진 비볼록 다각형인 경우에는  $O(n+m)$ 으로 증명되었다[7]. 그런데 실제의 응용분야에서 다루는 대상은 주로 비볼록 다각형이거나 원호(circular arc)를 포함하고 있는 비볼록 형상의 도형이다. 예를 들어, nesting 시스템[10-12]에서 다루는 도형은 보통 CAD 시스템을 이용하여 작성된 것으로서 일반적으로 비볼록형상을 가지며, 직선뿐만 아니라 원호도 포함되어 있다. 따라서 직선으로 구성된 볼록 다각형에 한정시켜 진행된 앞에서의 연구결과는 다양한 응용분야에 적용할 수 없다. 이에 따라 본 논문에서는 직선과 원호로 구성된 도형간의 최소거리문제(minimum distance problem)을 다루고자 한다.

두 도형사이의 최소거리를 구하는 간단한 한

\*正 會 員 : 釜山大 大學院 컴퓨터工學科 博士課程

\*\*正 會 員 : 釜山大 工大 컴퓨터工學科 副教授·工博

接受日字 : 1993年 6月 30日

1 次修正 : 1993年 12月 14日

2 次修正 : 1994年 3月 7日

방법으로는 도형을 구성하는 직선, 원호, 꼭지점을 모두 고려하여 일일이 계산해 주는 방법[12]을 생각할 수 있다. 이러한 경우에 각각  $m, n$ 개의 성분으로 구성된 도형의 경우, 그 연산시간은  $O(mn)$ 이 되며 모든 구성요소들을 고려하여 계산하게 되므로 비효율적이다. 따라서, 본 논문에서는 두 도형간의 최소거리계산에 반드시 고려해야 할 제한된 구성성분(직선, 원호)을 추출하고, 이들간의 거리를 기하학적인 특성을 이용하여 효율적으로 구할 수 있는 계산방법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 사용하는 용어에 대해 정의한다. 3장에서는 두 도형간의 거리를 효율적으로 계산하기 위해 필요한 정보들을 얻기 위한 전처리(pre-processing)과정과 두 도형간에 최소거리를 효율적으로 계산하기 위해 거리계산에 필요한 축소된 성분들의 리스트를 구하는 방법에 대해 기술한다. 또한 제한된 리스트의 기하학적인 특성을 이용하여 최소거리 계산에 불필요한 계산경우를 제거하는 과정과 구성요소의 대응관계에 따른 거리계산 방법에 대해 설명한다. 4장에서는 실험과 검토에 대해 기술하며, 5장에서 결론을 맺는다.

## 2. 정 의

이장에서는 본 논문에서 사용하는 기본적인 용어에 대해 정의한다(그림 1).

- 1) 원호와 직선으로 이루어진 2차원 형상을 도형이라 하며, 도형을 구성하는 원호와 직선을 성분이라 한다.
- 2) 도형을 이루는 성분과 성분이 만나는 점을 꼭지점이라 한다.
- 3) 어떤 도형을 이루는 성분상의 모든 점  $P(x, y)$ 의 집합을  $S$ 라 할 때,  $x$ 좌표값이  $\max\{x|P(x, y) \in S\}$ 이거나  $\min\{x|P(x, y) \in S\}$ 인 점들을 극단점(extreme point)이라 한다. 아울러  $y$ 좌표값이  $\max\{y|P(x, y) \in S\}$ 이거나  $\min\{y|P(x, y) \in S\}$ 인 점들도 극단점이라 한다.
- 4) 어떤 도형의 모든 극단점들을 포함하는 최소의 직각 사각형을 *MinMax*사각형이라 하고, 이것의 네 변중에서 네 직선  $x = \min\{x|P(x, y) \in S\}$ ,  $x = \max\{x|P(x, y) \in S\}$ ,  $y = \min\{y|P(x, y) \in S\}$ ,  $y = \max\{y|P(x, y) \in S\}$ 에 일치되는 변을 각각 *MinX*, *MaxX*, *MinY*, *MaxY*라 한다.

- 5) *MinMax*사각형을 이루는 변 *MaxX*, *MaxY*, *MinX*, *MinY*의 수평 및 수직 연장선을 각각 *MaxX*, *MaxY*, *MinX*, *MinY*의 연장선이라 한다.
- 6) 두 도형을 이루는 모든 점들의 집합을 각각  $S_1, S_2$ 라고 할 때,  $\min\{d(P_1, P_2) | P_1 \in S_1, P_2 \in S_2\}$ 를 두 도형간의 최소거리라 한다. 여기에서  $d(P_1, P_2)$ 는 두 점  $P_1, P_2$ 간의 유클리안(Euclidean)거리를 나타낸다.

## 3. 원호와 직선으로 구성된 두 도형간의 효율적인 최소거리 계산방법

원호와 직선으로 구성된 두 도형간의 최소거리를 효율적으로 구하기 위해 다음과 같은 단계를 거친다. 먼저, 주어진 도형의 형상 데이터를 보다 쉽게 다룰 수 있는 자료구조로 변형할 필요가 있으며, 이와 같은 전처리 과정을 거침으로써 최소거리계산에 필요한 여러 정보들을 얻을 수 있다. 다음에는 전처리 과정이 끝난 도형간의 위치관계에 따라, 도형을 이루는 전체 구성성분 중 최소거리계산시 고려되어야 할 제한된 구성성분들의 리스트를 구한다. 끝으로 이 리스트 성분들이 2차원적으로 상호 연결된 상태 즉 기하학적인 특성을 고려하여, 최소거리계산에 고려하지 않아도 되는 계산 경우를 제거함으로써 효율적으로 최소거리를 구할 수 있다. 본 장에서는 각 단계별 처리내용과 각 경우에 따른 거리 계산방법에 대해 설명한다.

### 3.1 전처리(PREPROCESSING) 과정

CAD시스템이나 수작업에 의해 작성된 도형이

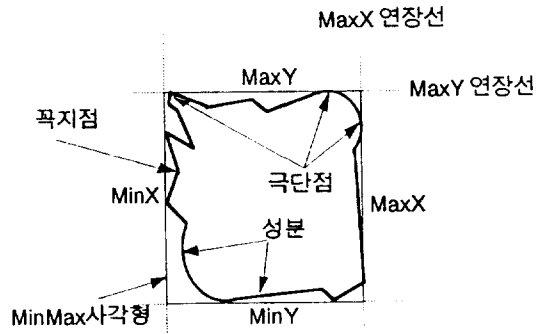


그림 1 용어 정의  
Fig. 1 Terminology definition

단순히 성분들(원호와 직선)이 나열된 리스트 형태로 표현되어 있으면, 도형의 형상을 고려한 여러가지 분석 및 연산에 부적합하다. 따라서 컴퓨터를 이용하여 유용한 분석 및 연산을 하기 위해서는 도형은 각 성분들의 연결관계 및 방위(orientation)등을 잘 나타낼 수 있는 구조로 표현되어야 한다. 본 논문에서는 원호와 직선 성분으로 구성된 도형간의 최소거리를 계산하기 위해 다음과 같은 전처리 과정을 거쳐 도형의 기하학적인 특성이 잘 나타나도록 하였다.

- 1) 도형을 이루는 각 성분들을 일정한 방향(반시계 방향)의 연결로 재배열한다. 이와같이 성분들의 재배열된 리스트를 그 도형의 성분 리스트라 한다.
- 2) 두 성분이 만나는 각도에 따라 각 꼭지점의 종류(오목, 볼록)를 결정한다.
- 3) 원호의 경우에는 원호의 종류(오목, 볼록)를 결정한다.
- 4) 도형의 극단점들을 구한다.
- 5) 도형의 *MinMax*사각형을 구한다.

### 3.2 도형간의 위치관계를 이용한 축소된 성분 리스트의 추출

두 도형간의 최소거리는, 실제로 두 도형의 마주보는 쪽에 위치한 성분들에 대해서만 거리를 계산하여 구할 수 있다. 여기에서 마주보는 쪽의 성분들을 효과적으로 정의해야 효율적인 연산결과를 얻을 수 있다. 본 절에서는 *MinMax*사각형과 극단점을 이용하여 마주보는 쪽의 성분리스트를 구하는 방법을 제안한다.

2차원 평면상에 놓여 있는 두 도형의 *MinMax*사각형의 상호 위치관계에 대한 대표적인 경우를 살펴보면 그림 2와 같다. 여기에서 (a)(b)의 경우는 상호간에 겹쳐진 영역이 없는 것으로서, *MinMax*사각형의 각 변을 연장선 방향 즉, x와 y축 방향에서 바라 볼 때, (a)는 변의 일부 또는 전체가 서로 마주 보고 있는 부분이 있는 경우이고 (b)는 없는 경우이다. 그림 2의 (c)(d)의 경우는 두 도형의 *MinMax*사각형간에 서로 중첩되는 부분이 있는 것으로서, (c)는 일부만이 중첩된 경우이고 (d)는 어느 한 쪽이 완전히 포함된 경우이다. 여기에서 서로 마주보는 쪽의 성분으로서, (a)-(c)의 경우에는 대각선 2-4와 2'-4'을 경계로 하여 4-1-2변 쪽의 성분과 2'-3'-4'변 쪽의 성분을 선택하고, (d)의 경우에는 두 도형의 전체 성분을 선택함으로써 보다 축소

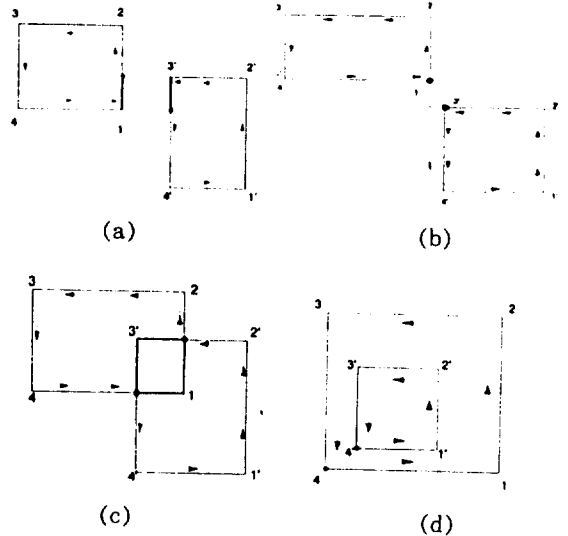


그림 2 두 *MinMax*사각형의 대표적인 위치관계  
**Fig. 2** Typical relationship of location between two *MinMax* rectangles

된 성분리스트를 구할 수 있다. 그러나 자세히 살펴보면 보다 축소된 성분리스트를 구할 수 있다.

그림 2에서와 같이 두 *MinMax*사각형이 놓여 있는 경우에, 서로 마주보는 변의 일부와 중첩된 변의 일부를 제외한 네 변에 위치한 극단점들을 반시계 방향 순서대로 나열한 리스트를 극단점 리스트라 하자. 이때 각 *MinMax*사각형에서 극단점들을 찾기 위해 반시계 방향으로 탐색하기 시작하는 점은, 그림 2의 (a),(c)에서는 굵은 선으로 표시된 부분과 같이 마주보는 변이나 중첩된 변을 벗어나는 점을 시작점으로 하고, 그림 2의 (b),(d) 경우에는 *MinMax*사각형의 네 꼭지점들 중에서 상호간에 제일 가까운 위치에 있는 꼭지점을 시작점으로 한다. 그림 2에서는 이러한 시작점을 굵은 검은 점으로 표시했다.

두 도형간에 최소거리를 계산하기 위해 반드시 필요한 성분들을 테스트 성분리스트라 하자. 두 도형의 테스트 성분리스트는 다음과 같이 두 가지 경우로 나누어 구할 수 있다.

(경우 1) 한 도형의 *MinMax* 사각형이 다른 도형의 *MinMax*사각형을 완전히 포함하지 않을 때

어떤 도형의 극단점리스트에서 마지막 극단점을 Tail 극단점, 처음 극단점을 Head 극단점이라 하자. 그러면 두 도형이 그림 2의 (a),(b),(c)와

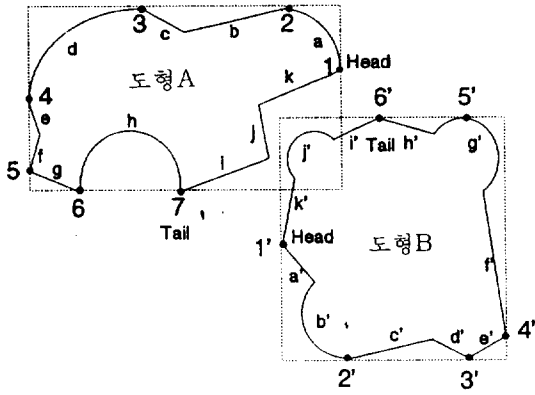


그림 3 극단점 리스트와 테스트 성분리스트의 예  
**Fig. 3** An example of the extreme point lists and the test Component lists

같은 위치관계로 놓여 있는 경우, Tail 극단점을 포함하는 성분부터 시작하여 Head 극단점을 포함하는 성분까지 반시계 방향으로 연결된 성분리스트가 테스트 성분리스트가 된다.

예를 들어, 그림 3에서와 같은 두 도형 A, B에서 극단점리스트는 각각 (1,2,3,4,5,6,7)과 (1',2',3',4',5',6')이다. 여기에서 1과 1'는 Head 극단점이고, 7과 6'는 Tail 극단점이다. 따라서 Tail 극단점에서 시작하여 Head 극단점까지의 성분리스트인 (i,j,k)와 (i',j',k')가 각각 도형 A,B의 테스트 성분리스트이다.

(경우2) 한 도형의 *MinMax*사각형이 다른 도형의 *MinMax*사각형을 완전히 포함할 때

그림 2의 (d)와 같이 *MinMax*사각형이 포함되는 경우에는 그림 4의 예에서와 같이 외부 도형(도형 A)의 일부 성분리스트가 내부 도형(도형 B)을 둘러싸고 있는 것을 알 수 있다. 따라서 이러한 외부 도형의 일부 성분리스트를 구하여 내부도형의 성분들과 최소 거리를 계산하면, 보다 효율적으로 두 도형간의 최소거리를 구할 수 있다. 이때 이러한 외부 도형의 일부 성분리스트와 내부 도형의 전체 성분리스트를 각각 그 도형의 테스트 성분리스트라 한다. 외부 도형의 테스트 성분리스트는 다음과 같은 방법으로 구한다.

먼저 외부 도형의 극단점리스트에서 이웃한 두 극단점사이의 성분리스트를 부분 성분리스트라 하자. 그러면 테스트 성분리스트는 부분 성분리스트중의 하나이다. 그림 5의 경우에는 ①,②,

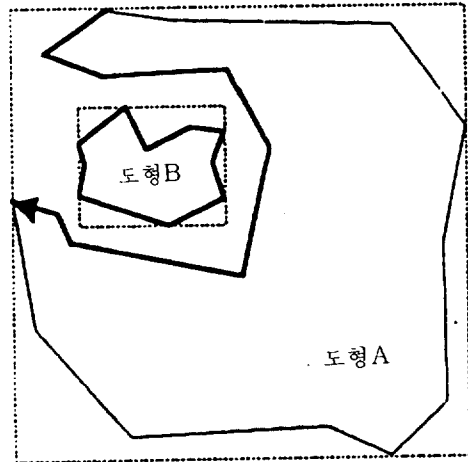


그림 4 포함의 예  
**Fig. 4** An example of the containment

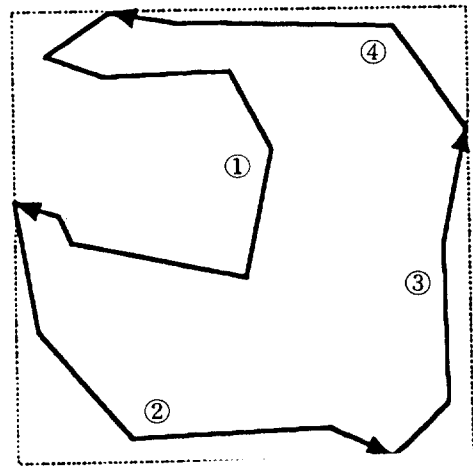


그림 5 부분 성분리스트  
**Fig. 5** Part component lists

③,④와 같은 네 개의 부분 성분리스트가 있다.

여기에서 부분 성분리스트에 대한 *MinMax*사각형을 부분 *MinMax*사각형(그림 6)이라 할 때, 테스트 성분리스트에 대한 부분 *MinMax*사각형을 반드시 내부 도형에 대한 *MinMax*사각형을 포함해야 한다. 따라서 부분 *MinMax*사각형이 내부 도형을 포함하지 못하는 부분 성분리스트는 테스트 성분리스트를 구하기 위한 고려대상에서 제외될 수 있다. 그런데 내부 도형을 포함하는 부분 *MinMax*사각형은 두 개 이상이

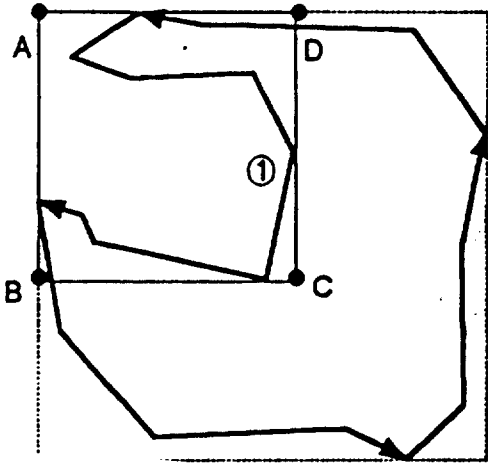


그림 6 부분 *MinMax*사각형  
Fig. 6 Part *MinMax* rectangle

존재할 수 있는데, 이들에 대한 부분 성분리스트를 대상 성분리스트라 하자. 예를 들어, 그림 7에서는 ①, ②, ③, ④가 부분 성분리스트이다. 이 중에서 내부 도형의 *MinMax*사각형을 포함하는 외부 도형의 부분 *MinMax*사각형은 사각형 ABCD, EFGH와 같다. 따라서 ②, ④ 부분 성분리스트는 고려대상에서 제외되고 ①, ③ 부분 성분리스트는 대상 성분리스트이다.

여기에서 그림 8에서와 같이 대상 성분리스트와 *MinMax*사각형의 일부 변으로 이루어지는 도형을 부분도형이라 정의하면, 이것을 이용하여 테스트 성분리스트를 결정할 수 있다. 왜냐하면 테스트 성분리스트에 대한 부분도형만이 내부 도형을 포함할 수 있기 때문이다.

두 도형간의 포함관계는, 포함되는 도형의 *MinMax*사각형이 정의되어 있는 경우에, 포함되는 도형의 임의의 한 *MinMax*연장선과 포함하는 부분 도형이 만나는 교점의 수를 이용하여 쉽게 구할 수 있다. 왜냐하면 두 도형이 포함관계를 만족하는 경우 그림 9에서와 같이 한쪽 방향의 연장선 위에 반드시 교점이  $(2n+1)$ 개 존재해야 하기 때문이다. 여기에서  $n$ 은  $n \geq 0$ 인 정수이다. 이때 한 *MinMax*연장선과 원호가 접하는 경우(그림 9의 교점 a)는 그 교점을 두개로 취급하며, 한 *MinMax*연장선과 꼭지점이 만난 교점중에서, 꼭지점을 이루는 두 성분이 *MinMax* 연장선을 통과하여 진행될때(그림 9의 교점 a')는 한 개로, *MinMax*연장선에서 되돌아 갈 때(그림 9의 교점 b')는 두 개로 취급한다. 그림 8에

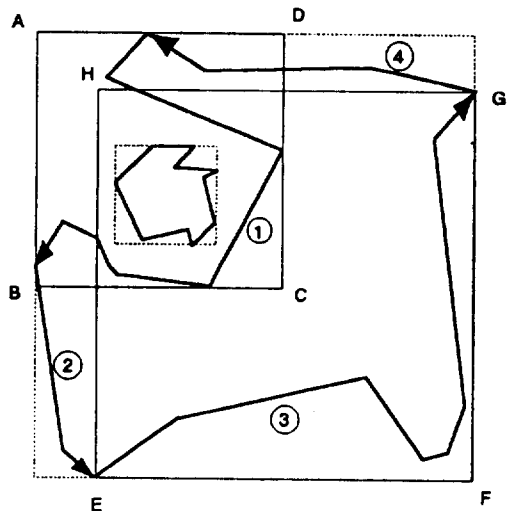


그림 7 부분 성분리스트와 부분 *MinMax*사각형  
Fig. 7 Part component lists and Part *MinMax* rectangles

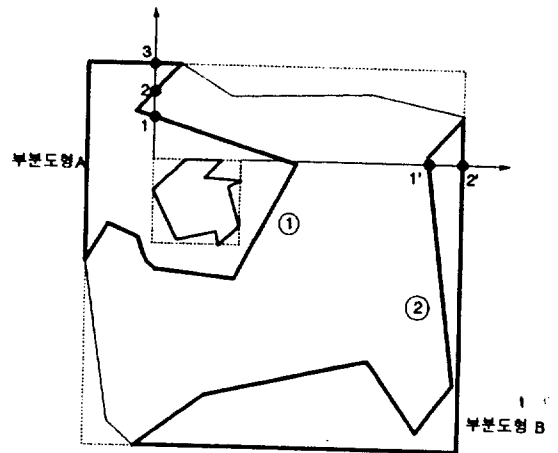


그림 8 테스트 성분 리스트의 결정 방법  
Fig. 8 Determination method for test component lists

서 부분도형 A의 경우, 내부 도형의 *MinMax*의 한 방향의 연장선위에 그 교점(1,2,3)이 세 개 존재하므로 내부 도형을 포함하는 부분도형이 되고, 부분도형 B의 경우에는 *MinMax*의 한 방향의 연장선위에 그 교점(1',2')이 두 개 존재하므로, 내부 도형을 포함하지 않음을 알 수 있다.

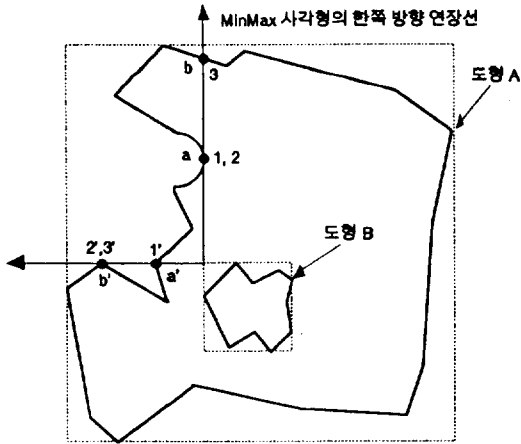


그림 9 포함여부 검사 방법  
Fig. 9 Method for containment test

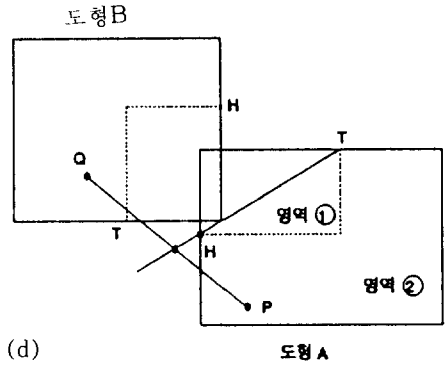
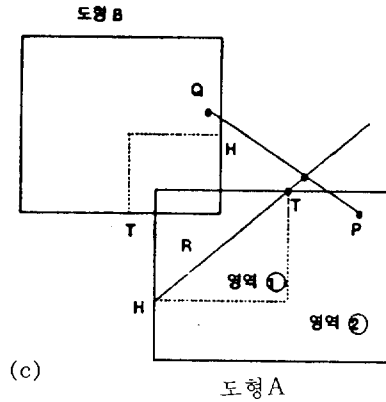
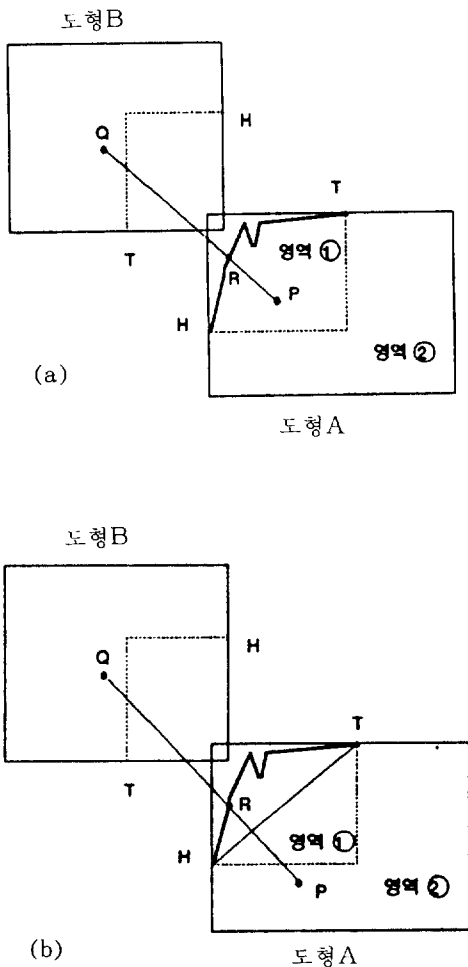


그림 10 테스트 성분리스트의 효율성 증명  
Fig. 10 Proof of the efficiency for test component lists

[정리 1] 두 도형간의 최소거리는 테스트 성분 리스트만을 고려하여 계산할 수 있다. (증명)[정리 1]은 다음과 같은 두 가지 경우로 나누어 증명하고자 한다.

(경우 1)한 도형의 *MinMax*사각형이 다른 도형의 *MinMax*사각형을 완전히 포함하지 않을 때 어떤 도형 A의 테스트 성분리스트에 포함되어 있지 않는 성분상의 한 점 P가 상대도형 B와 최소거리를 이룬다고 가정하자. 이때, 점 P는 그림 10의 (a),(b),(c),(d)에서와 같이 Head 극단점 H와 Tail 극단점 T에 의한 *MinMax*사각형 영역 ①에 놓일 수도 있고 또는 다른 영역 ②에 놓일 수도 있다. 점 P와 최소거리를 이루는 도형 B에서의 한점 Q가 있다고 하자.

(i) 점 P가 영역 ①에 있는 경우, 선분 PQ는 반드시 테스트 성분리스트의 적어도 한 성분과 교점(R)을 갖는다(그림 10의 (a)), 따라서, 이 교점과 점 Q간의 거리가 선분 PQ의 길이 보다 작게 되므로 가정에 모순된다.

- (ii) 점 P가 영역 ②에 있고, 선분 PQ가 선분 TH와 교점을 갖는 경우에는 (i)에서와 마찬가지로 테스트 성분리스트의 성분과 교점(R)이 생기므로 가정에 모순된다.(그림 10의 (b)).
  - (iii) 점 P가 영역 ②에 있고, 선분 PQ가 선분 TH의 점 T방향으로의 연장선과 만나는 경우(그림 10의 (c)), 선분 TQ의 길이가 선분 PQ의 길이보다 작다. 즉, 점 T와 점 Q의 X좌표의 차이의 절대값을  $|T_x - Q_x|$ 라고 표기할 때,  $|T_x - Q_x| < |P_x - Q_x|$ 이고  $|T_y - Q_y| \leq |P_y - Q_y|$ 이므로 선분 TQ의 길이가 선분 PQ의 길이보다 작게 되어 가정에 모순된다.
  - (iv) 점 P가 영역 ②에 있고 선분 PQ가 선분 TH의 점 H방향으로의 연장선과 만나는 경우(그림 10의 (d)),  $|H_x - Q_x| \leq |P_x - Q_x|$ 이고  $|H_y - Q_y| < |P_y - Q_y|$ 이므로 선분 HQ의 길이가 선분 PQ의 길이보다 작게되어 가정에 모순된다.
- 따라서 도형 A의 테스트 성분리스트 성분들에 포함된 점들만이 도형 B와 최소거리를 구하는데 적용될 수 있다. 도형 B의 테스트 성분리스트에 대해서도 마찬가지로 증명될 수 있으며, *MinMax*사각형이 다른 위치관계를 가질 때도 유사하게 증명할 수 있다.

(경우 2) 한 도형의 *MinMax*사각형이 다른 도형의 *MinMax*사각형을 완전히 포함할 때

내부 도형과 외부 도형간의 최소거리를 계산하는데 있어, 두 도형의 테스트 성분리스트만이 실제 고려대상이 되는 것은 자명하다. ■

### 3.3 기하학적인 특성을 이용한 계산경우의 축소

테스트 성분리스트를 이루는 모든 성분(직선, 원호, 꼭지점)을 고려해, [12]에서와 같이 거리 계산에 고려되어야 할 세가지 대응관계-꼭지점 대 직선 및 원호, 직선 대 원호, 원호 대 원호-에 따라 거리계산을 함으로써 최소거리를 구할 수 있다. 그러나 이러한 방법은 실제 최소거리 결정에 영향을 주지 않는 즉 고려하지 않아도 되는 성분들에 대해 불필요한 계산을 하는 단점이 있다. 기하학적인 특성을 이용하여, 테스트 성분리스트중에서 최소거리계산에 반드시 필요한 제한된 성분들에 대해서만 거리를 계산하면 불필요한 계산경우를 없앨 수 있으므로, 보다 효율적으로 최소거리를 구할 수 있다.

최소거리는 3.2절에서 구한 두 도형의 테스트 성분리스트를 이용하여 각 리스트를 구성하는 요소(꼭지점, 직선, 원호)간의 대응관계에 따라

계산을 하되, 모든 구성요소에 대해 계산을 하는 것이 아니라 다음에 기술하는 세가지 정리의 경우에 대해서만 계산함으로써 효율적으로 구할 수 있다.

[정리 2] 꼭지점 대 직선 또는 꼭지점 대 원호간의 거리계산 경우, 오목 꼭지점은 고려하지 않고 볼록 꼭지점에 대해서만 계산을 한다.

(증명) 그림 11에서 오목 꼭지점 P가 도형 B와 최소거리를 이루는 점을 Q라 하자. Q를 중심으로 하여 반지름 PQ의 원을 그리고 점 P에서 접선 L을 그려보면, P가 오목점이기 때문에 P와 만나는 두 성분 중 적어도 하나는 접선 L에 의해 나누어지는 두 반평면 영역중에서, 점 Q가 있는 반평면 영역에 존재해야 한다. 이때, 이러한 성분은 반드시 원 내부를 통과하게 되므로 점 Q와의 거리가 선분 PQ의 길이보다 작은 점이 성분상에 존재한다. 따라서 오목 꼭지점 P는 최소거리계산에서 고려할 필요가 없다. ■

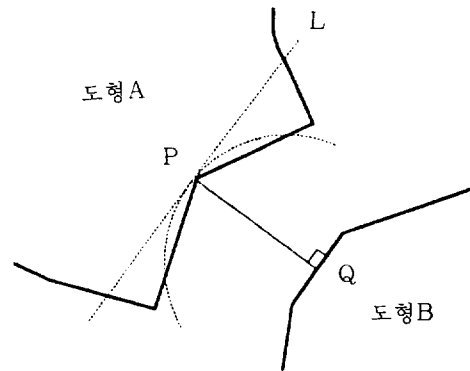


그림 11 오목꼭지점과 최소거리와의 관계

Fig. 11 Relation between the concave vertex and the minimum distance

[정리 3] 직선 대 원호간의 최소거리계산의 경우, 볼록원호만 직선과의 거리를 계산하면 된다.

(증명) 오목원호의 경우, 그림 12의 (a)에서와 같이 원호의 양 끝점 S와 E를 연결한 선과 원호 내부에 상대방 도형의 성분이 들어와 있는 경우와 (b)와 같이 외부에 있는 경우가 있다. (a)의 경우에는 [정리 2]에 의한 볼록꼭지점과 원호와의 거리계산에 의해 오목원호와 도형 A의 직선들과의 거리계산은 이미 모두 고려되었다. (b)의 경우에도 마찬가지로 오목원호의 양 끝점에서 도형 A와의 거리계산이 처리된다. 따라서 오목

원호와 직선간의 거리계산을 할 필요가 없다. 그러나 그림 13에서와 같이 직선과 볼록원호간에는 보다 짧은 거리를 갖는 경우가 있을 수 있으므로 계산을 따로 해 주어야 한다. ■

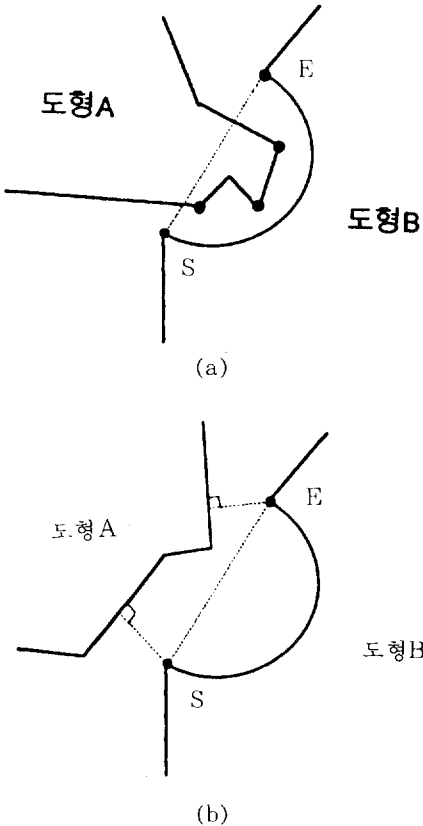


그림 12 오목원호와 직선과의 거리계산 예  
**Fig. 12** An example of distance calculation between a concave circular arc and a line

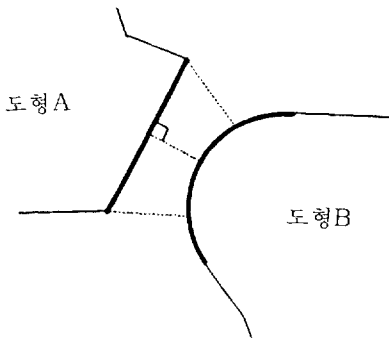


그림 13 볼록원호와 직선과의 거리계산의 필요성  
**Fig. 13** Necessity of distance calculation between a convex circular arc and a line

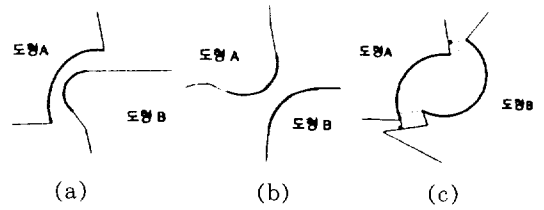


그림 14 원호 대 원호간의 거리계산의 필요성  
**Fig. 14** Necessity of distance calculation between two circular arcs

[정리 4] 원호와 원호간의 거리계산의 경우에는 오목원호 대 볼록원호, 볼록원호 대 볼록원호의 경우에 대해 계산해 주어야 하나 오목원호 대 오목원호간의 거리계산은 고려할 필요없다. (증명)여기에서는 [정리 2]와 [정리 3]에 의해 구해지는 거리보다 짧은 거리를 갖는 것이 원호와 원호간에 존재하는 예를 보임으로써 필요성을 증명하고자 한다. 그림 14의 (a)와 (b)의 경우에는 점선으로 표기된 부분과 같이 보다 짧은 거리를 갖는 경우가 존재할 수 있으므로 반드시 고려해야 한다. 그러나 (c)의 경우와 같이 오목원호간에는 점선으로 표기된 부분과 같은 거리가 이미 계산되어 고려되었으므로 더 이상 필요하지 않다. ■

### 3.4 테스트 성분리스트 구성 성분간의 대응관계에 따른 거리계산 방법

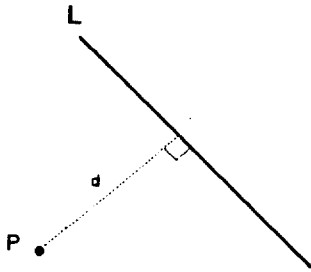
두 도형간의 최소거리를 계산하기 위해서는 두 도형의 테스트 성분리스트 구성 성분간의 대응관계중에서 3.3절의 [정리 2], [정리 3], [정리 4]에 해당하는 대응관계만 고려하면 된다. 본 절에서는 각 대응관계에 대한 거리계산 방법에 대해 설명한다.

#### 3.4.1 꼭지점 대 직선 및 원호간의 거리계산

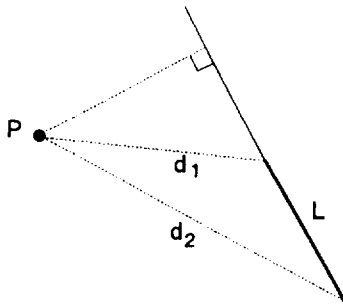
꼭지점 대 직선간의 거리계산은 그림 15의 (a)에서와 같이 꼭지점 P에서 직선 L에 내린 수선의 발의 길이(d)를 거리로 계산하되, 그림 15의 (b)와 같이 수선의 발이 직선에 포함되지 않을 경우는 꼭지점과 직선의 두 끝점간의 거리( $d_1, d_2$ )중 작은 값을 취한다.

꼭지점 대 원호간의 거리계산은 원호가 오목, 볼록인 경우로 나누어 계산할 수 있다. 꼭지점과 볼록원호인 경우는, 꼭지점이 볼록원호의 중심(O)와 시작점(S), 끝점(E)를 연결한 직선이 이





(a)



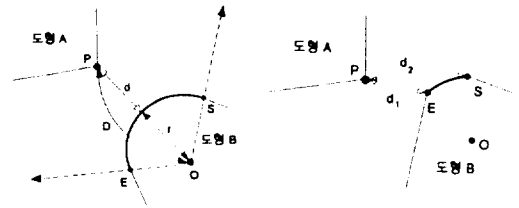
(b)

그림 15 꼭지점과 직선사이의 거리계산  
**Fig. 15** Distance calculation between a vertex and a line

루는 영역에 포함된 경우(그림 16의 (a))와 포함되지 않는 경우(그림 16의 (b))로 나누어 계산한다. 꼭지점과 오목원호인 경우는 꼭지점이 원호와 선분 SO, EO가 이루는 부채꼴(OSE)의 영역에 포함된 경우(그림 16의 (c))와 포함되지 않는 경우(그림 16의 (d))로 나누어 계산한다.

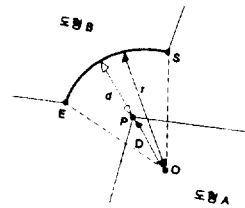
3.4.2 직선 대 원호간의 거리계산 방법

직선 대 볼록 원호간의 거리계산은 볼록원호의 중심점(O)에서 직선에 내린 수선의 발을 이용하여 구한다. 볼록원호의 중심점에서 직선에 내린 수선과 원호와 교점이 있고, 수선의 발이 직선에 포함될 경우, 그림 17과 같이 구할 수 있다. 그 외의 경우에는 직선과 볼록원호와의 거리는 계산하지 않아도 된다(이와같은 경우, 직선과 볼록원호와의 거리는 직선의 양끝점과 볼록원호와의 거리가 된다. 그런데 직선의 양 끝점이 볼록 꼭지점인 경우에 3.4.1절에서와 같이 꼭지점과 원호와의 거리계산에서 고려되므로 직선과 원호간의 거리계산에서 다시 계산할 필요가 없다).

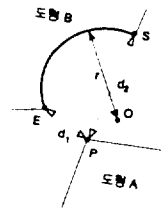


(a)

(b)



(c)



(d)

- (a) distance  $d = D - r$
- (b) distance  $d = \min\{d_1, d_2\}$
- (c) distance  $d = r - D$
- (d) distance  $d = \min\{d_1, d_2\}$

그림 16 꼭지점과 원호간의 거리계산  
**Fig. 16** Distance calculation between a vertex and a circular arc

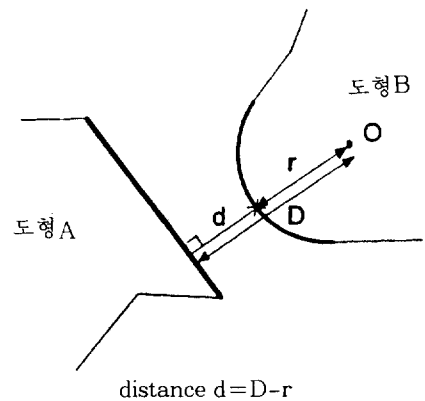


그림 17 직선과 원호간의 거리계산  
**Fig. 17** Distance calculation between a line and a circular arc

3.4.3 원호대 원호간의 거리계산 방법

원호대 원호간의 거리계산은 볼록원호와 볼록원호, 볼록원호와 오목원호의 경우로 나누어 계산한다. 볼록원호와 볼록원호의 경우는, 두 원호의 중심점을 연결한 선분  $O_aO_b$ 와 두 원호와의 교점(그림 18의(a)에서  $l_1, l_2$ )이 있는 경우에 대해 그림 18의 (a)같이 거리를 계산하며, 교점이 없는 경우는 계산할 필요가 없다(이와같은 경우에는, 두 볼록원호간의 거리는 꼭지점과 볼록원호의 경우에서 계산되므로 원호 대 원호의 경우에서 고려할 필요가 없다.)

볼록원호와 오목원호간의 거리계산의 경우는, 볼록원호의 중심( $O_a$ )이 오목원호와 그 중심( $O_b$ )이 이루는 부채꼴 영역에 포함되고, 두 원호의 중심을 연결한 직선과 두 원호와 각각 교점( $l_1, l_2$ )이 있는 경우, 그림 18의 (b)와 같이 거리를 구할 수 있다. 이외의 경우에는, 원호의 양 끝점 중 볼록 꼭지점에 의해 원호와의 거리가 3.4.1절에서 구해진다.

4. 실험 및 검토

원호와 직선으로 구성된 도형간에 최소거리를 구할 때, 도형의 모든 성분을 고려하여 계산하는 방법과 본 논문에서 제안하는 방법 즉 도형의 위치관계를 이용하여 테스트 성분리스트를 구하고, 불필요한 계산의 경우를 제거하여 최소거리를 구하는 방법에 대해 실험을 통해 성능비교를 하였다.

본 실험에서는 Cadam[9]이라는 CAD tool로 만들어진 형상 데이터를 실험 데이터로 이용하

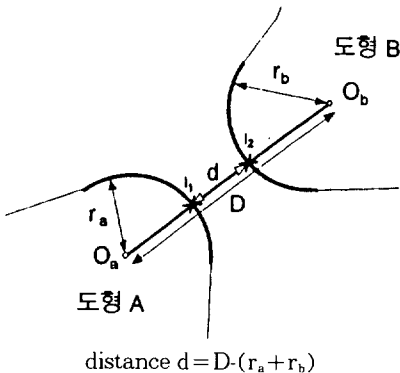


그림 18(a) 볼록원호와 볼록원호간의 거리계산  
Fig. 18(a) Distance calculation between two convex circular arc

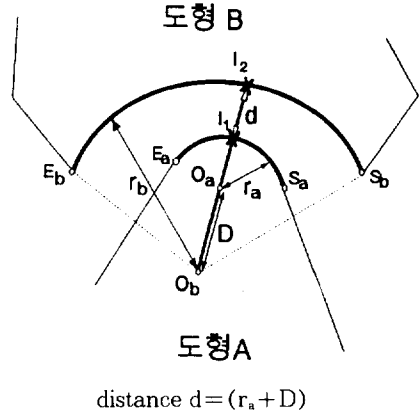


그림 18(b) 볼록원호와 오목원호간의 거리계산  
Fig. 18(b) Distance calculation between a convex circular arc and a concave circular arc

|   |          |          |          |          |
|---|----------|----------|----------|----------|
| L | 220.9999 | 386.0000 | 263.0000 | 386.0000 |
| a | 0        |          |          |          |
| L | 379.0000 | 15.0000  | 469.0000 | 15.0000  |
| a | 0        |          |          |          |
| A | 354.0000 | 45.0000  | 379.0000 | 45.0000  |
|   | 70.0000  |          |          |          |
| a | 0        |          |          |          |
| A | 130.0000 | 45.0000  | 130.0000 | 70.0000  |
|   | 105.0000 | 45.0000  |          |          |
| a | 0        |          |          |          |

그림 19 Cadam데이터의 예  
Fig. 19 An example of Cadam data

였으며, 본 논문에서 제안하는 방법을 적용하여 X-window 시스템에서 개발한 부품배치 프로그램(nesting program)을 이용하여 실험 데이터를 판재위에 배치하였다. 판재위에 부품을 배치할 때는 절단가공을 위한 공간으로서 부품간에 일정한 거리를 유지시켜 주어야 하는데, 이를 위해 최소거리 계산방법이 이용되었다. 실험 데이터로 이용된 Cadam 데이터(그림 19)의 구조를 살펴보면, 직선과 원호로 구성되어 있으며 이 두 성분을 조합하여 자유로운 형상의 부품을 만든다. L과 A는 각각 직선과 원호를 나타내고, a 0는 한 성분이 끝난 것을 표시한다.

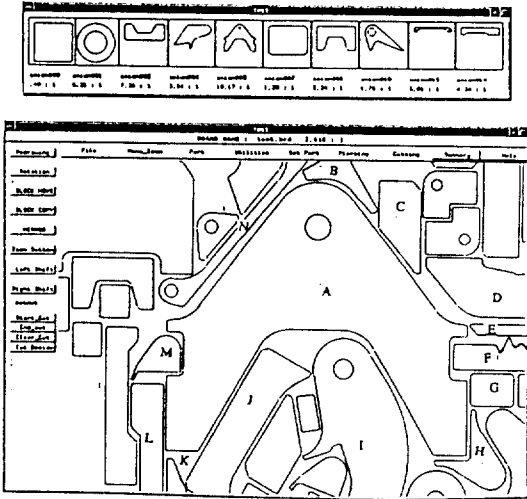


그림 20 부품배치 프로그램을 이용한 부품의 배치 결과  
**Fig. 20** Result of disposing parts using the nesting program

성능평가를 위해 도형의 모든 구성성분을 고려한 방법[12]과 본 논문에서 제안하는 방법에 대해 최소거리를 구하기 위한 계산량을 비교하였다. 이때 계산량은 구성성분간의 대응관계(직선 대 꼭지점, 원호 대 직선, 원호 대 원호)에 관계없이 모두 1회 연산으로 간주하였다.

그림 20은 Cadam 데이터를 부품배치 프로그램(nesting program)이용하여 판재위에 배치한 결과이며, 이때 도형 A와 주위의 도형 B~N과의 최소거리 계산량을 최소거리를 구하는 방법에 따라 표 1, 표 2, 표 3에 각각 나타내었다.

표 1은 도형 A와 주위의 도형 B~N과의 최소거리를 구할 때, 도형을 구성하는 모든 성분들을 고려하는 방법의 계산량을 나타내며, 표 2는 테스트 성분리스트만 고려하되 기하학적인 특성을 고려하지 않고 테스트 성분리스트간의 모든 계산경우를 고려한 방법이다. 표 3은 테스트 성분리스트만 고려하되, [정리 2], [정리 3], [정리 4]에서의 같이 불필요한 계산의 경우를 제거했을 때의 계산량을 나타낸다. 표 1, 표 2, 표 3에서 보는 바와 같이 본 논문에서 제시한 방법을 부품배치 프로그램에 적용하여 실험한 결과, 도형간의 최소거리 계산량을 매우 줄일 수 있음을 알 수 있다.

본 논문에서 제시한 방법은, 최악의 경우에 그

표 1 도형의 모든 구성요소를 고려한 방법  
**Table 1** A method of considering all the components of shape  
 최소거리 계산량=11.835

| FIG*  | COMP* | VERTEX | LINE* | ARC* | FIG.  | COMP. | VERTEX | LINE | ARC | QTY*  |
|-------|-------|--------|-------|------|-------|-------|--------|------|-----|-------|
| A     | 28    | 28     | 19    | 9    | B     | 8     | 8      | 7    | 1   | 539   |
| A     | 28    | 28     | 19    | 9    | C     | 6     | 6      | 6    | 0   | 390   |
| A     | 28    | 28     | 19    | 9    | D     | 11    | 11     | 8    | 3   | 772   |
| A     | 28    | 28     | 19    | 9    | E     | 6     | 6      | 6    | 0   | 390   |
| A     | 28    | 28     | 19    | 9    | F     | 46    | 46     | 26   | 20  | 3370  |
| A     | 28    | 28     | 19    | 9    | G     | 8     | 8      | 4    | 4   | 596   |
| A     | 28    | 28     | 19    | 9    | H     | 13    | 13     | 7    | 6   | 959   |
| A     | 28    | 28     | 19    | 9    | I     | 17    | 17     | 11   | 6   | 1219  |
| A     | 28    | 28     | 19    | 9    | J     | 12    | 12     | 10   | 2   | 818   |
| A     | 28    | 28     | 19    | 9    | K     | 9     | 9      | 8    | 1   | 604   |
| A     | 28    | 28     | 19    | 9    | L     | 12    | 12     | 8    | 4   | 856   |
| A     | 28    | 28     | 19    | 9    | M     | 10    | 10     | 6    | 4   | 726   |
| A     | 28    | 28     | 19    | 9    | N     | 8     | 8      | 4    | 4   | 596   |
| TOTAL | 364   | 364    | 247   | 117  | TOTAL | 166   | 166    | 111  | 55  | 11835 |

- a : 도형
- b : 도형을 구성하는 성분의 갯수(원호+직선)
- c : 꼭지점의 갯수(오목꼭지점+볼록꼭지점)
- d : 직선의 갯수
- e : 원호는 갯수(오목원호+볼록원호)
- f : 최소거리 계산량

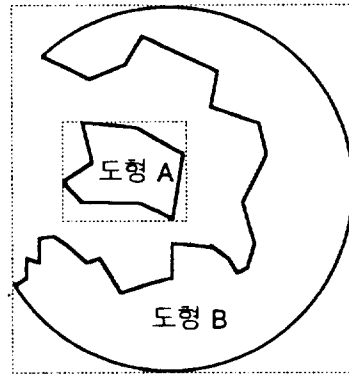


그림 21 최악의 경우 예  
**Fig. 21** An example of the worst case

그림 21과 같이, 두 도형을 구성하는 모든 성분이 테스트 성분리스트가 되어, 도형의 모든 성분을 고려하는 방법과 동일한 계산량을 수행하게 되며, 연산 차수는  $O(nm)$ 이 된다.

그러나 앞의 실험에서 보았듯이, 두 도형의 형상과 도형간의 위치관계를 볼 때, 테스트 성분리스트의 구성성분 갯수는 평균적으로 도형의 전

표 2 테스트 성분리스트의 모든 구성요소를 고려 하되 불필요한 계산의 경우를 제거하지 않은 방법

Table 2 A method of considering the test components list without eliminating unnecessary calculation

최소거리 계산량=1.560

| FIG.  | COMP. | VERTEX | LINE | ARC | FIG.  | COMP. | VERTEX | LINE | ARC | QTY  |
|-------|-------|--------|------|-----|-------|-------|--------|------|-----|------|
| A     | 7     | 8      | 4    | 3   | B     | 7     | 8      | 6    | 1   | 136  |
| A     | 4     | 5      | 2    | 2   | C     | 5     | 6      | 5    | 0   | 59   |
| A     | 4     | 5      | 2    | 2   | D     | 4     | 5      | 3    | 1   | 50   |
| A     | 5     | 4      | 3    | 2   | E     | 2     | 3      | 2    | 0   | 27   |
| A     | 6     | 7      | 4    | 2   | F     | 41    | 42     | 22   | 19  | 697  |
| A     | 6     | 7      | 4    | 2   | G     | 3     | 4      | 1    | 2   | 59   |
| A     | 2     | 3      | 2    | 0   | H     | 3     | 4      | 2    | 1   | 19   |
| A     | 5     | 6      | 3    | 2   | I     | 14    | 15     | 9    | 5   | 202  |
| A     | 6     | 7      | 4    | 2   | J     | 10    | 11     | 8    | 2   | 164  |
| A     | 6     | 7      | 4    | 2   | K     | 1     | 2      | 1    | 0   | 21   |
| A     | 1     | 2      | 1    | 0   | L     | 1     | 2      | 1    | 0   | 4    |
| A     | 6     | 7      | 4    | 2   | M     | 5     | 6      | 3    | 2   | 89   |
| A     | 4     | 3      | 2    | 2   | N     | 3     | 4      | 2    | 1   | 33   |
| TOTAL | 62    | 71     | 39   | 23  | TOTAL | 99    | 112    | 66   | 34  | 1560 |

표 3 본 논문에서 제안한 방법

Table 3 The proposed method in this paper

최소거리 계산량=709

| FIG   | COMP. | CV* | LINE | CCA* | CVA* | FIG   | COMP. | CV | LINE | CCA | CVA | QTY |
|-------|-------|-----|------|------|------|-------|-------|----|------|-----|-----|-----|
| A     | 7     | 4   | 4    | 2    | 1    | B     | 7     | 6  | 6    | 1   | 0   | 76  |
| A     | 4     | 2   | 2    | 1    | 1    | C     | 5     | 6  | 5    | 0   | 0   | 39  |
| A     | 4     | 2   | 2    | 1    | 1    | D     | 4     | 3  | 3    | 0   | 1   | 27  |
| A     | 5     | 3   | 3    | 1    | 1    | E     | 2     | 3  | 2    | 0   | 0   | 23  |
| A     | 6     | 3   | 4    | 2    | 0    | F     | 41    | 16 | 22   | 9   | 10  | 279 |
| A     | 6     | 3   | 4    | 2    | 0    | G     | 3     | 4  | 1    | 0   | 2   | 45  |
| A     | 2     | 3   | 2    | 0    | 0    | H     | 3     | 2  | 2    | 1   | 0   | 13  |
| A     | 5     | 2   | 3    | 2    | 0    | I     | 14    | 5  | 9    | 3   | 2   | 63  |
| A     | 6     | 3   | 4    | 2    | 0    | J     | 10    | 7  | 8    | 2   | 0   | 72  |
| A     | 6     | 3   | 4    | 2    | 0    | K     | 1     | 2  | 1    | 0   | 0   | 15  |
| A     | 1     | 2   | 1    | 0    | 0    | L     | 1     | 2  | 1    | 0   | 0   | 4   |
| A     | 6     | 3   | 4    | 2    | 0    | M     | 5     | 2  | 3    | 1   | 1   | 33  |
| A     | 4     | 2   | 2    | 1    | 1    | N     | 3     | 2  | 2    | 0   | 1   | 20  |
| TOTAL | 62    | 35  | 39   | 18   | 5    | TOTAL | 99    | 60 | 65   | 17  | 17  | 709 |

a : 볼록꼭지점(Convex Vertex) 갯수

b : 오목원호(Concave Arc) 갯수

c : 볼록원호(Convex Arc) 갯수

체 구성성분 갯수보다 매우 줄어들게 되므로, 도형의 모든 구성성분을 고려하여 최소거리를 구하는 방법보다 효율적인 방법이 된다. 또한 테스트 성분리스트간의 모든 계산경우를 고려하는 것이 아니라 3.3절에 설명된 [정리 2], [정리 3], [정리 4]에 해당하는 경우만 계산을 하므로, 보

다 효율적으로 두 도형간의 최소거리 계산량을 매우 줄일 수 있다.

### 5. 결 론

도형간의 위치관계와 기하학적인 특성을 이용하여 최소거리 계산에 고려해 주어야 할 축소된 성분들의 리스트(테스트 성분리스트)를 추출하고, 테스트 성분리스트간의 거리계산에 있어 최소거리 영향에 주지 못하는 불필요한 계산의 경우를 제거함으로써 효율적으로 최소거리를 구하는 방법을 제안하였다. 또한 본 논문에서 제안한 방법을 부품배치 프로그램에 적용하여 실험한 결과, 평균적으로 최소거리 계산량을 매우 줄일 수 있음을 알아 보았다.

본 논문에서 제안한 방법은 2차원 평면에서 원호와 직선으로 구성된 형상간의 최소거리 계산에 효율적으로 응용될 수 있다. 예를 들면, CAD tool로 작성된 도형 데이터를 이용한 부품 배치 프로그램(nesting program)과 GIS(Geographic Information System)[13, 14]에서 영역이 원호와 직선으로 구성될 때, 두 영역간의 최소 직선거리를 구하는 경우등이다.

본 논문에서 제안한 방법은 최악의 경우, 도형의 모든 성분을 고려해야 된다. 그러나 두 도형간의 평균 최소거리 계산량은 매우 줄일 수 있었다. 앞으로 테스트 성분리스트의 성분들의 공간적으로 연결되어 있는 제약조건을 최대한 이용하여 연산 차수로 줄일 수 있는 방법에 대한 연구가 필요하다.

### 참 고 문 헌

[ 1 ] Franco P. Preparata, Michael Ian shamos, "Computational Geometry-An introduction," Springer-Verlag, 1985, pp. 222-223

[ 2 ] D.T.Lee, Franco Preparata, "Computational Geometry-A Survey," I EEI Trans. on Computer, vol.c-33, No.12,pp. 1072-1101. Dec., 1984

[ 3 ] Atallah, "A liner time algorithm for the Hausdorff distance between convex polygons," Inform. Processing Lett., vol. 8, pp. 207-209, Nov.1983

[ 4 ] Rebert Sedgewick, "Algorithms in C++", Addison-Wesley, 1992, pp. 401-407

- [ 5 ] F.Chin and C.A.Wang, "Minimum vertex distance between separable convex polygons," Inform. Processing Lett., vol 18, pp. 41-45, Jan. 1984
- [ 6 ] J.T.Schwartz, "Finding the minimum distance between two convex polygons," Inform. Process. Lett., Vol.13, No.4, pp.168-170, 1981
- [ 7 ] F.Chin and C.A.Wang, "Optimal Algorithm for the Intersection and the Minimum Distance Problems Between Planar Polygons, IEEE Trans. on Computer, Vol. c-32, No.12, pp. 1203-1207, Dec., 1983
- [ 8 ] Bhattacharya and G.T.Toussaint, "Efficient algorithms for computing maximum distance between two finite planar sets," J.Algorithms, vol 4, pp.121-126, June 1983
- [ 9 ] CATIA Training Manual-Interactive user access(IUA), Dassault systems, 1988
- [ 10 ] PINS-Precision's Improved Nesting System (주)한국 CADD엔지니어링, 1992
- [ 11 ] 김민환, "판재절단을 위한 부품의 최적배치", 최종연구보고서, 한국기계연구소, 1992
- [ 12 ] 전봉기, 김종민, 홍봉희, 김민환, "X 윈도우 시스템을 이용한 부품 배치프로그램의 개발", 한국정보과학회 '92가을 학술발표논문지, 제19권 2호, pp.783-786, 1992
- [ 13 ] Understanding GIS, ESRI번역본, GIS의 이해와 실습, 통계청 자료처리국, 전산개발과, 1991
- [ 14 ] S.Aronoff, GEOGRAPHIC INFORMATION SYSTEMS : A Management Perspective, WDL Publications, Ottawa, Canada, 1989
- [ 15 ] Douglas A.young, "The X window System Programming and Applications with Xt (OSF/Motif Edition), "Prentice-Hall, 1990, pp.1-13

## 저 자 소 개



### 김종민(金鍾敏)

1966년 3월 1일생. 1992년 부산대학교 공대 컴퓨터공학과 졸업. 1994년 동 대학원 컴퓨터공학과 졸업(석사). 현재 동 대학원 컴퓨

터공학과 박사과정



### 김민환(金民煥)

1957년 12월 30일생. 1980년 서울대학교 공대 전기공학과 졸업. 1983년 동 대학원 컴퓨터공학과 졸업(석사). 1988년 동 대학원 컴

퓨터공학과 졸업(공학). 1991년~1992년 미국 워싱턴대학교 객원연구원. 현재 부산대학교 공과대학 컴퓨터공학과 부교수