

# 유사 삼각 분할을 이용한 단순 다각형의 형태 변환 알고리즘

## (Morphing Algorithm of Simple Polygons using Similar Triangulation)

송은하<sup>†</sup> 이주희<sup>\*\*</sup> 이상호<sup>\*\*\*</sup>

(Eun-Ha Song) (Joohee Lee) (Sang-Ho Lee)

**요약** 본 논문에서는 단순 다각형을 대상으로 하는 형태 변환 알고리즘을 제안한다. 주어진 다각형을 삼각 분할하고 그 듀얼 트리로부터 구성된 듀얼 루트 트리를 이용하여 형태 변환을 유도하는 기하학적인 방법이다. 이 방법은 기존의 알고리즘이 수학적 모델링을 기반으로 하기 때문에 감수해야했던 많은 양의 함수계산을 피할 수 있으며 다각형의 속성을 유지하는 삼각 분할을 사용함으로써 중간 단계에 생성된 다각형들이 언제나 적합한 형태의 다각형이 될 수 있다는 특징을 갖는다. 이러한 작업이 가능하도록 하기 위해서 본 논문에서는 유사 삼각 분할(similar triangulation)과 유사 트리의 개념을 이용하였다.

**Abstract** In this paper, we propose a new morphing algorithm for simple polygons. In this algorithm, we triangulate polygons and then construct a dual root tree from the dual trees of the triangulations to make a deformation from one polygon to another. In this method, we don't have to calculate complex functions which are inevitable in traditional algorithms. Our algorithm guarantees the validity during deformation process. We define the concepts of the similar triangulation and the similar tree to make this deformation possible.

### 1. 서론

형태 변환(morphing)이란 하나의 물체로부터 다른 물체로 그 형태가 변화해 가는 효과를 나타내기 위해 사용하는 방법으로 컴퓨터 그래픽스 분야에서 오랫동안 널리 사용되어 온 중요한 연산이다(그림 1).

형태 변환에 대한 기존의 알고리즘들은 주어진 물체를 수학적 함수를 사용하여 모델링하고 변화하는 각 과정을 일일이 계산을 통해 얻어내는 방법을 사용한다. 따라서 기존의 형태 변환 기법을 이용하여 물체의 움직임을 표현하기 위해서는 복잡하고 많은 양의 함수 계산



그림 1 형태 변환의 예

을 하지 않을 수 없다. 또한 대응점 위주의 계산을 하기 때문에 변화해 가는 중간 과정에서 적절하지 못한 그림이 생성될 수도 있다는 문제점을 가진다[2][8][9]. 이러한 문제는 주어진 그림을 수학적으로 인식하고 두 그림 사이의 차이점을 계산하는 것에 치중할 뿐 그 물체가 가지고 있는 속성이나 기하학적 정보를 유지하고 있지 않기 때문이다.

1994년 Guibas와 Hershberger는 형태 변환의 대상인 다각형을 이전의 방법이 아닌 이진 트리를 이용하여 하나의 다각형(원시 다각형)에서 원하는 다각형(목적 다각형)으로 변환이 가능한 형태 변환 알고리즘을 발표하였다[4][5]. 이 알고리즘에서는 다각형 자체가 아니라

· 이 논문은 "99년도 정보통신부 <정보통신 우수대학 교육연구 기저제 지원사업>의 연구 지원에 의해서 이루어졌음."

† 비 회 원 : 한국과학기술연구원 연구원

ehsong@robot.kist.re.kr

\*\* 비 회 원 : 일본 京都大學校 공학연구소 연구원

jhlee@archi.kyoto-u.ac.jp

\*\*\* 종신회원 : 이화여자대학교 컴퓨터학과 교수

shlee@ewha.ac.kr

논문접수 : 1999년 2월 12일

심사완료 : 2000년 8월 16일

다각형의 속성을 그대로 유지하는 이진 트리로의 변환을 생각했고 이러한 이진 트리의 회전 연산은 회전 후에도 그 속성을 그대로 유지할 수 있기 때문에 형태 변환 되는 어떠한 다각형도 항상 조건을 만족하게 되므로써 기존의 알고리즘들이 가지고 있던 문제점을 근본적으로 해결할 수 있는 방안을 제시하였다. 그러나 그들이 제안한 알고리즘은 단순 다각형 중에서 원시 다각형과 목적 다각형의 대응하는 변이 항상 평행해야 하며 중간 단계의 모든 다각형들도 모두 그러한 성질을 유지해야 하는 제한적인 다각형만을 대상으로 하였다.

본 논문에서는 제한적인 단순 다각형을 연구 대상으로 하는 기존의 알고리즘과는 달리 일반적인 단순 다각형을 대상으로 하는 새로운 형태 변환 알고리즘을 제안한다. 이 알고리즘은 각 다각형을 삼각 분할하고, 삼각 분할의 듀얼 트리로부터 얻어진 듀얼 루트 트리(dual root tree)를 이용하여 형태 변환을 유도하는 것이다. 이 알고리즘은 다각형이 가지고 있는 속성을 그대로 유지하면서 물체의 형태가 변화하는 과정을 효과적으로 보여줄 수 있으며 수학적 함수를 사용하는 것이 아닌 듀얼 루트 트리라는 자료구조의 회전 연산을 사용하는 두 가지 특징을 가진 알고리즘이다.

2. 용어 정의

형태 변환을 위해 주어진 원시 다각형과 목적 다각형을 각각  $P$ 와  $Q$ 라 표기하기로 한다. 임의의 다각형을 삼각 분할한 후 얻어진 듀얼 트리의 중심(center)을 루트로 갖는 트리를 듀얼 루트 트리라 한다(그림 2)(3.1절 [정의 4]). 원시 다각형  $P$ 로부터 생성된 듀얼 루트 트리를 원시 트리( $t_p$ ), 목적 다각형  $Q$ 로부터 생성된 듀얼 루트 트리를 목적 트리( $t_q$ )라고 정의한다.



그림 2 듀얼 루트 트리로의 변환

듀얼 루트 트리에서 임의의 노드를  $p$ 라 할 때  $deg_{max}(p)$ 는 노드  $p$ 의 최대 자식 노드의 개수를 말하며  $i_p$ 는  $p$ 가  $i$ 번째 자식 노드를 가지고 있는지의 여부를 표시하는 부울(boolean) 변수를 말한다. 즉, 자식 노드가 있는 경우는 1, 없는 경우는 0의 값을 갖는다. 집합  $C_p$ 는 노드  $p$ 의 자식 노드들의 집합으로  $i_p$ 가 1인 인덱스  $i$ 를 원소로 갖는다( $C_p = \{i \mid i_p = 1\}$ ). 즉 노드  $p$ 가 몇

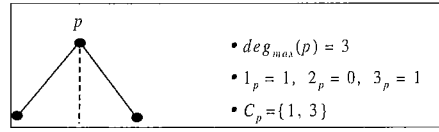


그림 3  $deg_{max}(p)=3$  인 노드  $p$

번째 자식 노드를 갖는지를 집합으로 표시한 것이다. 그림 3은 최대 자식 노드의 개수가 3인 노드  $p$ 와 그의 자식 노드에 대한 표시의 예이다.

[정의 1] 유사 노드(similar node)

다음 조건 중 하나를 만족하면 원시 트리의 노드  $p$ 와 그에 대응하는 목적 트리의 노드  $q$ 를 유사 노드라 한다. (단, 노드  $p$ 와 노드  $q$ 는 실제 존재하지 않는 가상 노드(dummy node)일 수 있다.)

- (1)  $C_p = C_q$     (2)  $C_p \subset C_q$     (3)  $C_p \supset C_q$

정의 1에서 조건 (1)은 원시 트리의 노드  $p$ 와 이에 대응하는 목적 트리의 노드  $q$ 의 자식 노드의 개수와 위치가 같다는 것이다. 조건 (2)는 원시 트리의 노드  $p$ 가  $i$ 번째 자식 노드를 갖는 경우, 대응하는 목적 트리의 노드  $q$ 도 반드시 그 위치에 자식 노드를 가져야 한다는 것을 의미한다. 조건 (3)은 반대로 목적 트리의 노드  $q$ 가  $i$ 번째 자식 노드를 가지고 있는 경우에 원시 트리의 대응 노드  $p$ 도  $i$ 번째 자식 노드를 가지고 있어야 한다는 것이다.

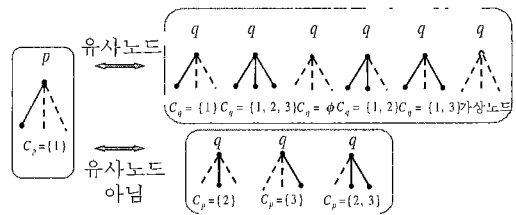


그림 4 자식 노드의 개수가 1인 노드  $p(deg_{max}(p)=3)$ 의 유사 노드

이 세 가지 조건 중 하나를 만족하면 노드  $p$ 와 그의 대응 노드  $q$ 는 유사 노드라고 한다. 그림 4는  $deg_{max}(p)$ 가 3이면서 하나의 자식 노드가 있는 경우에 대한 유사 노드의 종류와 유사 노드가 될 수 없는 경우를 보여 준다.

[정의 2] 유사 트리(similar tree)

임의의 노드  $p$ 에 대하여  $p_i$ 는  $p$ 의  $i$ 번째 자식 노드라 한다. 원시 트리  $t_p$ 의 루트( $p$ )와 목적 트리  $t_q$ 의 루트( $q$ )가 서로 유사 노드이고  $t_p$ 의 각 부분트리(subtree)  $t_{p_i}$  ( $1 \leq i \leq deg_{max}(p)$ )가  $t_q$ 의 각 부분트리  $t_{q_i}$  ( $1 \leq i \leq deg_{max}(q)$ )와 유사 트리인 경우  $t_p$ 와  $t_q$ 를 유사 트리라 한다.

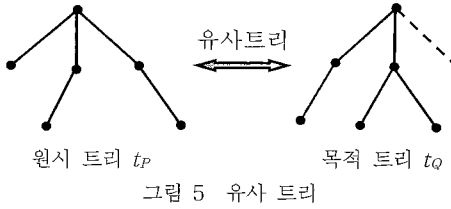


그림 5 유사 트리

듀얼 루트 트리인 원시 트리  $t_p$ 와 목적 트리  $t_q$ 가 서로 유사 트리라고 하는 것은 원시 트리  $t_p$ 에 속하는 노드와 그 노드에 대응하는 목적 트리  $t_q$ 의 노드가 모두 유사 노드가 됨을 의미한다. 그림 5의 원시 트리의 각 노드는 목적 트리에 있는 자신의 대응 노드와 모두 유사 노드이다. 이런 경우를 유사 트리라고 한다.

**[정의 3] 유사 삼각 분할(similar triangulation)**

두 개의 삼각 분할에 대한 각각의 듀얼 트리로부터 생성된 듀얼 루트 트리가 유사 트리인 경우, 이 두 삼각 분할을 유사 삼각 분할이라고 한다.

두 개의 삼각 분할이 유사 삼각 분할이 되게 하기 위해서는 두 트리가 유사 트리가 되도록 하면 된다. 그림 6은 유사 삼각 분할된 두 다각형  $P$ 와  $Q$ 이다.

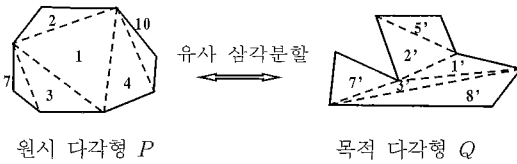


그림 6 유사 삼각 분할

**3. 형태 변환 알고리즘 Triangulation Morphing**

- 알고리즘 Triangulation Morphing**

  1. 주어진 원시 다각형  $P$ 와 목적 다각형  $Q$ 를 각각 임의로 삼각 분할한다.
  2. 삼각 분할된 다각형  $P$ 와  $Q$ 를 각각 듀얼 트리 구성한다.
  3. 각각의 듀얼 트리에서 중심(center)을 구한 후, 그 중심을 루트로 하는 듀얼 루트 트리  $t_p$ 와  $t_q$ 로 변환한다.
  4. 원시 트리의 듀얼 루트 트리인  $t_p$ 와 목적 트리의 듀얼 루트 트리인  $t_q$ 의 구조를 비교하면서 두 트리가 유사 트리가 되도록 원시 다각형의 삼각 분할을 변환시킨다.(에지 플립 사용)
  5. 원시 트리  $t_p$ 와 목적 트리  $t_q$ 가 같은 형태를 갖도록 모양을 조정한다.

본 알고리즘의 가장 큰 특징은 초기에 임의로 구성된 삼각 분할을 유사 삼각 분할로 변경하는 것이다. 일반적인 단순 다각형을 대상으로 한 **Triangulation Morphing**을 단계별로 자세히 살펴보기로 한다.

**3.1 삼각 분할과 듀얼 루트 트리**

삼각 분할의 종류는 매우 다양하나 다각형의 기하학적인 속성을 유지하기 위해 본 알고리즘에서 사용한 특별한 삼각 분할 방법은 없다. 임의의 삼각 분할로 시작한다. 그림 7은 두 다각형  $P$ 와  $Q$ 를 임의로 삼각 분할한 모양이다.

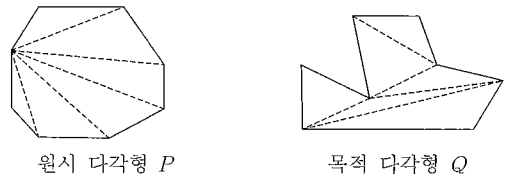


그림 7 다각형의 임의 삼각 분할

삼각 분할된 각 삼각형을 정점으로 하고 인접한 삼각형 사이를 에지로 연결하면 듀얼 트리를 얻을 수 있다(그림 8)[1][7]. 이러한 듀얼 트리의 정점은 삼각형에 대한 정보를 모두 유지할 수 있으므로 다각형의 속성을 잘 표현할 수 있다는 장점을 가지고 있지만 루트가 없는 자유 트리(free tree)이므로 일반적인 트리 연산을 적용할 수 없다는 단점도 가지고 있다. 따라서 듀얼 트리에 루트를 지정해 루트가 있는 트리로 변환하게 되었고 이렇게 변환한 트리가 듀얼 루트 트리이다.

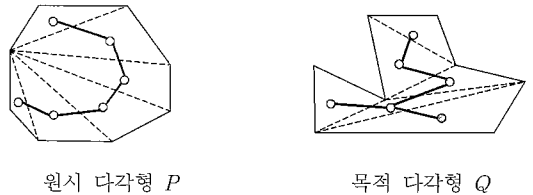


그림 8 삼각 분할의 듀얼 트리

**[정의 4] 듀얼 루트 트리**

다각형을 삼각 분할하고, 분할된 각 삼각형은 정점으로, 인접한 삼각형간의 관계를 에지로 연결한 듀얼 트리의 각 경로 중 가장 긴 경로의 중심을 루트로 하여 재구성한 트리를 듀얼 루트 트리라고 한다.

삼각 분할로부터 구성한 듀얼 트리는 자유 트리이다. 이러한 자유 트리는 일관성 있는 트리 연산을 적용하기

에 어려움이 있으므로 삭제, 삽입 또는 탐색 등의 트리 연산을 적용하기 위해서 듀얼 트리의 중심을 찾아 그 중심을 루트로 하는 듀얼 루트 트리로 변환한다(그림 9).

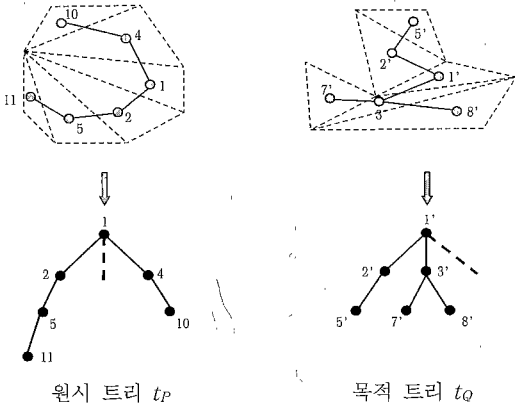


그림 9 듀얼 루트 트리로의 변환

루트가 있는 트리으로 변환하기 위해서는 듀얼 트리에서 중심이 되는 노드를 찾아야 한다. 자유 트리에서 그 중심을 결정하는 방법은 여러 가지가 있으나 본 논문에서는 주어진 트리에서 가장 긴 경로(path)를 찾아 그 경로 상의 가운데 위치하는 노드를 루트로 정한다.

트리 상의 가장 긴 경로를 찾는 방법은 다음과 같다. 트리 상의 임의의 단말 노드로부터 트리의 노드들을 너비 우선 탐색(breadth first search) 순으로 방문하여 가장 긴 경로를 가지는 노드( $t_1$ )를 찾고, 다시 그 노드로부터 너비 우선 탐색 순으로 트리의 각 노드를 방문하여 가장 긴 경로를 갖는 노드( $t_2$ )를 찾으면 이 두 노드 사이의 경로( $t_1$ 과  $t_2$  사이의 경로)가 그 트리의 최장 경로가 된다. 이 방법을 사용하면 선형 시간 내에 듀얼 트리의 중심을 찾을 수 있다.

**[성질 1]** 듀얼 루트 트리는 모든 노드가 최대 3의 차수를 갖는다. 따라서 루트의 경우는 최대 세 개의 자식 노드를 갖게 되고 루트가 아닌 일반 노드들은 최대 두 개의 자식 노드를 갖는다.

듀얼 트리의 중심이 결정되고 나면 루트에 해당하는 노드의 세 자식 노드의 순서 결정해야 한다. 이를 위해서 루트에 해당하는 삼각형의 세 점 중에서  $y$ 값이 가장 큰 점을 기준으로 삼각형의 세 변을 반시계 방향으로 방문한다. 첫 번째 변과 인접한 삼각형을 루트의 첫 번째 자식 노드로, 두 번째 변과 인접한 삼각형을 두 번째 자식 노드로 정하고, 마지막으로 인접한 삼각형을 세 번

째 자식 노드로 정한다. 만약에 방문중인 루트의 각 변 중에서 인접한 삼각형이 존재하지 않는 경우에는 가상 노드로 자식 노드를 대신한다. 이러한 가상 노드는 자식 노드간의 위치 관계를 결정하는데 중요하므로 직선 에지와 구별되도록 점선을 사용하여 나타낸다. 그림 10은 원시 다각형의 듀얼 루트 트리에서 루트의 자식 노드에 대해 순서를 정하는 과정을 보여준다.

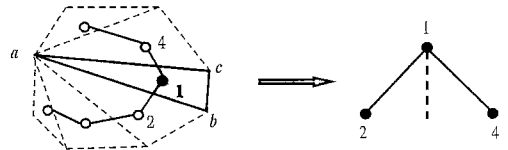


그림 10 듀얼 루트 트리의 루트의 자식 노드 순서 결정

노드의 번호는 루트를 1로 시작하여 너비 우선 탐색의 순서로 부여한다. 가상 노드가 있는 경우에는 그에 해당되는 번호는 건너뛴다.

루트의 자식 노드의 순서가 정해지면 루트 이하의 일반 노드들의 자식 노드의 순서도 정할 수 있다. 순서를 정하는 기준은 루트 노드의 자식 노드를 정하는 기준과 동일하다. 단지 시작하는 점이 가장 큰  $y$ 값을 갖는 정점이 아니라 부모 노드와 인접한 변이 되는 것이 차이점이다. 즉, 부모 노드와 인접한 변을 기준으로 나머지 두 변을 반시계 방향으로 방문하여 처음에 인접한 삼각형을 그 노드의 왼쪽 자식 노드로 두 번째 인접한 삼각형을 오른쪽 자식 노드로 정한다.

### 3.2 유사 삼각 분할

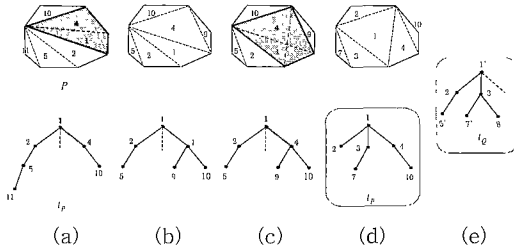
이 단계는 주어진 원시 다각형의 삼각 분할이 목적 다각형의 삼각 분할과 유사 삼각 분할이 되도록 하는 단계이다. 원시 다각형과 목적 다각형이 서로 유사 삼각 분할이 되도록 하기 위해서는 삼각 분할의 모양을 변경하여야 하는데, 삼각 분할을 변경하는 방법은 삼각 분할을 위해 삽입한 대각선을 변경하는 것이다. 다각형의 내부는 모두 삼각형으로 분할되어 있고 하나의 대각선은 언제나 두 개의 삼각형에 인접해있으므로 이중 한 대각선을 변경하면 그 대각선으로 인하여 형성되었던 두 개의 삼각형이 변경된다.

**[성질 2]** 삼각 분할을 구성하는 대각선은 언제나 두 개의 삼각형에 인접하기 때문에 대각선을 변경해도 이 두 삼각형 이외의 다른 삼각형에는 영향을 주지 않는다.

이러한 에지 플립을 이용하여 원시 다각형의 삼각 분할을 목적 다각형의 삼각 분할과 유사 삼각 분할이 되

도록 변경하는 과정은 다음과 같다. 원시 트리  $t_p$ 의 루트부터 그에 대응하는 목적 트리  $t_Q$ 의 노드와 서로 유사 노드인지를 비교한다. 트리는 너비우선탐색순으로 방문된다. 대응하는 두 노드가 서로 유사 노드이면 다음 노드를 방문하여 다시 그들이 유사 노드인지를 비교한다. 하지만 대응하는 두 노드가 유사 노드가 아닌 경우에는 원시 다각형의 해당되는 두 인접한 삼각형에서 에지 플립을 적용해 삼각 분할의 상태를 변경하여 두 노드가 유사 노드가 되었는지를 다시 검사한다.

원시 트리의 노드와 목적 트리의 대응하는 노드가 서로 유사 노드가 아니라는 의미는 자식 노드의 위치 관계가 정의 1의 조건에 맞지 않는 것을 말한다. 이러한 경우는 삼각형의 각 변에 인접한 삼각형들의 위치 관계가 다르다는 의미가 된다. 바꾸어 말하면 삼각 분할의 모양을 바꾸어 최대한 비슷한 모양의 삼각 분할로 변경하고자 하는 것이 유사 삼각 분할 과정이다.



(a) 플립 결정 (b) 에지 플립 (c) 플립 결정 (d) 에지 플립 (e) 목적 트리

그림 11 유사 삼각 분할

그림 11은 주어진 원시 다각형에 에지 플립을 적용함으로써 변환되는 삼각 분할과 각각의 삼각 분할에 대응해서 변환되는 듀얼 루트 트리의 상태를 나타낸 것이다. (a)의 원시 트리과 맨 오른쪽 목적 트리 (e)와 비교해보면 두 루트 노드의 자식 노드의 위치가 다를 수 있다. 즉 두 트리의 루트는 서로 유사 노드가 아니다. 그러므로 원시 트리의 루트가 가지는 자식 노드 중 유사 노드의 조건을 위배하는 노드와 루트 사이의 에지를 플립하여 유사 삼각 분할이 되도록 한다.

탐색은 언제나 너비 우선 순서로 일어난다. 두 트리의 루트가 유사 노드인지를 살피기 위해서는 자식 노드를 차례로 살펴본다. 첫 번째와 두 번째 자식 노드까지를 비교하면 정리 1의 조건 (2)를 만족한다. 그러나 세 번째 자식 노드를 비교하게되면 이 조건은 더 이상 만

족하지 못하게 된다. 따라서 두 트리의 루트가 유사 노드가 되지 못하게 하는 노드는 세 번째 자식 노드가 되고, 이 노드와 루트를 연결하는 에지가 에지 플립을 적용하는 대상이 된다. 따라서 원시 트리의 루트 노드 1과 그의 세 번째 자식 노드인 노드 4를 연결하는 에지를 플립한다.

에지 플립이 일어나면 삼각 분할이 변형되므로 이에 대응하는 듀얼 루트 트리의 구조도 수정해 주어야 한다. 그림 11의 (b)는 한번의 에지 플립이 일어난 후의 변화된 삼각 분할과 변경된 원시 트리의 형태이다. 노드의 번호를 너비 우선 탐색의 순서이기 때문에 같은 삼각형이라도 해당하는 노드의 번호가 변경되기도 한다.

수정된 원시 트리 (b)의 루트를 목적 트리 (e)의 루트와 다시 비교하면 아직도 유사 노드가 아님을 알 수 있다. 따라서 앞에서와 같이 다시 한번 에지 플립을 적용한다. 두 번째로 수정된 원시 트리는 (d)와 (e)의 루트들은 이제 유사 노드가 되었다. 위와 같은 과정을 원시 트리  $t_p$ 와 목적 트리  $t_Q$ 에서 대응하는 모든 노드들이 유사 노드가 될 때까지, 즉 원시 트리와 목적 트리가 서로 유사 트리가 될 때까지 반복한다.

원시 트리가 목적 트리와 유사 트리가 되기 위해서 필요한 에지 플립의 수를 생각해 본다. 그림 11의 경우, 원시 트리  $t_p$ 와 목적 트리  $t_Q$ 는 두 번의 에지 플립으로 유사 트리가 되었고, 원시 다각형의 삼각 분할과 목적 다각형의 삼각 분할이 유사 삼각 분할이 되었다.

일반적인 경우에 필요한 에지 플립의 수를 계산해본다. 루트가 아닌 노드에 대해서는 유사 노드로 변환하기 위해 한 번의 에지 플립이면 충분하다. 문제는 세 개의 자식 노드를 가지고 있는 루트의 경우이다. 루트이면서 자식 노드가 하나 밖에 없는 경우는 두 개의 삼각형으로만 이루어진 간단한 다각형인 경우이므로 설명을 생략한다. 원시 트리의 노드나 목적 트리의 노드 중 하나가 세 개의 자식 노드를 모두 가지고 있는 경우는 무조건 대응 노드와 유사 노드가 되므로 고려하지 않아도 된다. 따라서 우리가 고려해야 하는 경우는 자식 노드가 두 개인 경우이다. 자식 노드가 두 개인 경우 유사 노드가 되지 못하는 경우는 그림 12에 보이는 세 유형의 노드이다.

그림 12의 노드  $p$ 에서 1번 자식 노드와 연결된 에지를 플립하면 노드  $q$ 와 같이 된다. 마찬가지로 노드  $p$ 에서 3번 자식 노드와 연결된 에지를 플립하면 노드  $r$ 이 된다. 노드  $q$ , 노드  $r$ 의 경우에도 마찬가지이다. 이와 같이  $p$ ,  $q$ ,  $r$  세 가지 유형의 노드는 모두 한 번의 에지 플립으로 다른 두 유형의 노드로 변경이 가능하다.

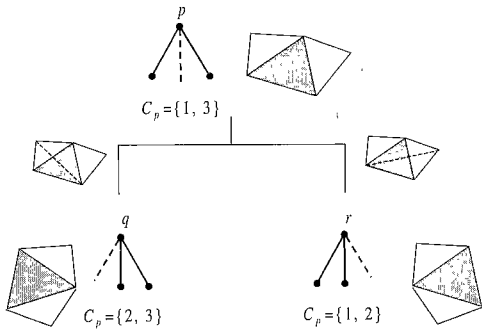


그림 12 자식 노드가 두 개인 루트 노드의 에지 플립

따라서 에지 플립은 한번만 일어나면 된다. 그러나 그림 11에서와 같이 에지 플립이 일어나도 가상 노드의 자리가 변하지 않아서 유사 노드가 될 수 없는 경우도 있다.

이런 경우는 듀얼 루트 트리를 구성하기 위해 루트를 결정할 때 현재 루트로 정해진 노드 이외에 후보 노드가 하나 더 있는 경우이다. 즉 듀얼 트리의 가장 긴 경로상의 노드의 수가 짝수인 경우이고, 한번의 에지 플립으로 트리를 재구성하면서 기존의 루트가 아니었던 루트 후보가 새로 루트가 되면서 일어나는 현상이다. 이 현상은 다시 한번 에지 플립을 함으로서 해결된다. 이러한 내용은 다음과 같이 정리하고 증명한다.

**[정리 1]** 두 개의 듀얼 루트 트리  $t_p$  와  $t_q$  가 있고, 주어진 다각형  $P$  의 정점의 개수가  $k$  개라 할 때  $t_p$  와  $t_q$  를 유사 트리로 만들기 위해서는 최대  $k+1$  번의 에지 플립이 필요하다.

**(증명)** 앞에서 설명한 것과 같이 루트인 경우 최대 두 번, 그리고 루트가 아닌 모든 노드에 대해서는 많아야 한 번의 에지 플립이 일어나게 되므로 총  $k+1$  번의 에지 플립이 일어나게 된다.

앞에서도 살펴본 바와 같이 에지 플립이 일어나야 하는 에지는 두 노드가 유사 노드가 되지 못하게 하는 자식 노드와 연결된 에지이다. 그림 12에서 노드  $p$  와  $q$  의 경우를 살펴보자. 노드  $p$  를 원시 트리의 노드라고 할 때, 노드  $p$  의 두 번째 자식 노드가 가상 노드인 것이 조건을 위배하게 된다. 따라서 두 번째 자식 노드와 연결된 에지를 플립해야 하는데 이 경우 두 번째 자식 노드는 가상 노드이므로 플립을 할 수가 없다. 다른 에지를 플립의 대상으로 정해야한다. 노드  $q$  와 유사 노드로 만들기 위해서는 그림에서 보는 것처럼 1번 자식 노드와 연결된 에지를 플립하면 된다. 노드  $q$  에서는 첫 번째 자식 노드가 가상 노드이다. 따라서 플립을 해야 하

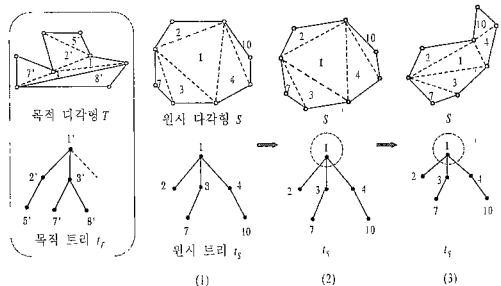
는 에지가 가상 노드와 연결된 에지인 경우에는 대상이 되는 목적 트리의 노드에서 가상 노드로 되어있는 자식 노드에 대응되는 자식 노드와 연결된 에지를 플립하면 간단히 유사 노드로 만들 수 있게 된다.

에지 플립이 일어나면 트리를 재구성한다. 일반적인 노드에서 에지 플립이 일어나는 경우에는 트리 전체를 재구성할 필요는 없다. 성질 2에 의해 플립이 일어난 레벨과 그 아래 레벨을 포함한 두 레벨의 노드만 수정하면 된다. 일반 노드가 아닌 루트에서 에지 플립이 일어나게 되면 트리 전체를 재구성해야 한다. 하지만 루트의 경우라도 많아야 두 번의 에지 플립이 일어나므로 이 단계에서 필요한 트리의 재구성 회수도 최대 두 번이다.

### 3.3 모양 조정 및 형태 변환 알고리즘을 적용한 예

본 장에서 제안한 알고리즘의 마지막 단계는 유사 트리로 변환된 두 트리  $t_p$  와  $t_q$  를 사용하여 원시 다각형이 목적 다각형의 형태를 갖도록 변환하는 단계이다. 본 단계는 원시 다각형의 각 삼각형을 목적 다각형의 해당하는 삼각형의 모양과 크기가 동일해지도록 조정함으로써 궁극적으로 원시 다각형이 목적 다각형과 동일한 다각형이 되도록 하는 과정이다.

앞에서의 설명처럼 이 단계는 두 개의 다각형 대신 그 다각형의 속성을 표현하고있는 두 개의 듀얼 루트 트리 사이에서 일어나는 과정이다. 작업 순서는 각 트리의 루트로부터 시작되며 너비우선 탐색의 순서로 각 트리의 노드를 모두 처리한다. 제일 먼저 원시 트리의 루트에 해당하는 삼각형부터 조정해 준다. 그림 13의 (2)~(4)는 원시 트리의 루트인 노드 1에 해당하는 삼각형의 모양과 크기를 조정하는 과정이다. 루트의 모양 조정이 끝나면 자식 노드들을 너비 우선 탐색 순서로 방문하여 목적 다각형과 같은 모양과 크기를 갖도록 해당 삼각형의 속성을 바꾸어 준다. 그림에서 주어진 원시 트리의 노드 2와 노드 3은 그에 대응하는 목적 트리의 노드가 존재하므로 그림 13의 (5)~(9)처럼 삼각형의 모양만 변환시켜 주면 된다.



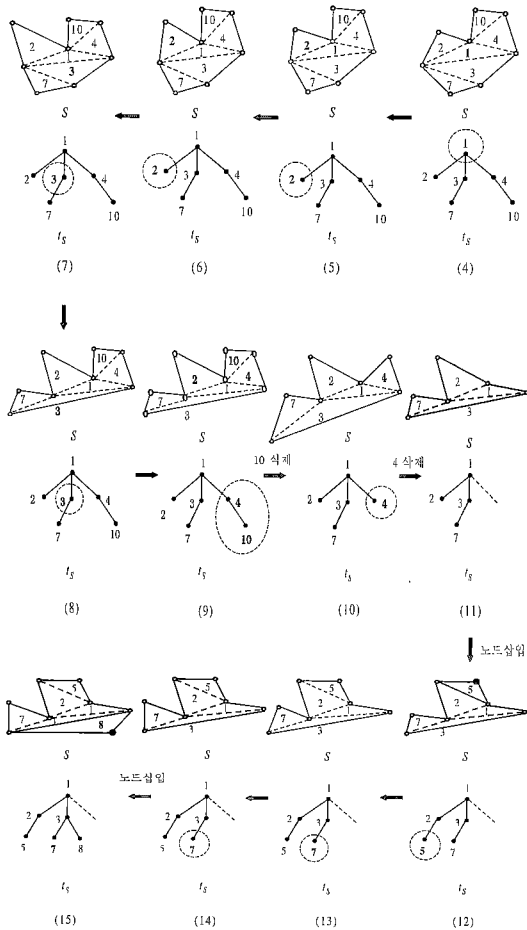


그림 13 알고리즘 Triangulation\_Morphing의 형태변환 과정

앞에서의 설명처럼 이 단계는 두 개의 다각형 대신 그 다각형의 속성을 표현하고있는 두 개의 듀얼 루트 트리 사이에서 일어나는 과정이다. 작업 순서는 각 트리의 루트로부터 시작되며 너비우선 탐색의 순서로 각 트리의 노드를 모두 처리한다. 제일 먼저 원시 트리의 루트에 해당하는 삼각형부터 조정해 준다. 그림 13의 (2)~(4)는 원시 트리의 루트인 노드 1에 해당하는 삼각형의 모양과 크기를 조정하는 과정이다. 루트의 모양 조정이 끝나면 자식 노드들을 너비 우선 탐색 순서로 방문하여 목적 다각형과 같은 모양과 크기를 갖도록 해당 삼각형의 속성을 바꾸어 준다. 그림에서 주어진 원시 트리의 노드 2와 노드 3은 그에 대응하는 목적 트리의 노드가 존재하므로 그림 13의 (5)~(9)처럼 삼각형의 모양

만 변환시켜 주면 된다.

다음으로 노드 4를 보면 그에 대응하는 목적 트리의 노드가 존재하지 않음을 알 수 있다. 이런 경우 노드의 삭제가 일어나는데 우선 노드 4를 제거하기 전에 노드 4의 모든 자손 노드들을 먼저 제거한다. 노드 4의 자손 노드들을 단말 노드(leaf node)부터 차례로 삭제하고 마지막으로 노드 4를 삭제한다. 그림 13의 (10)과 (11)의 다각형에서는 노드 10과 노드 4에 해당하는 삼각형이 순서대로 삭제된 것을 볼 수 있다.

다음에 비교해야 하는 노드는 노드 2의 자식 노드들이다. 원시 트리에는 노드 2의 자식 노드가 하나도 존재하지 않는다. 그러나 목적 트리의 노드 2'에는 왼쪽 자식 노드가 존재함을 알 수 있다. 이러한 경우에 노드의 삽입이 일어난다. 원시 트리의 노드 2의 왼쪽 자식 노드를 해당 위치에 삽입한다. 그림 13의 (12)는 노드 2에 대응하는 삼각형의 왼쪽 자식 노드가 새로 삽입되는 것을 보여준다. 그림 13의 (15)에서는 노드 8이 삽입되는 과정을 볼 수 있다. 원시 다각형 P는 그림과 같이 목적 다각형 Q로 그 형태가 변환되었다. 이 과정에서는 두 번의 번의 삭제와 두 번의 삽입이 적용되었다.

#### 4. 실험 및 알고리즘 분석

알고리즘 Triangulation\_Morphing은 C++와 OpenGL을 사용하여 직접 구현하였다. 그림 14는 듀얼 루트 트리의 루트로부터 순서대로 삼각형들이 변화되는 모습을 저장한 후 모든 삼각형들이 동시에 형태를 변환하는 모습을 구현한 것이다. 이해를 돕기 위해 분할된 각 삼각형간의 관계를 보여주는 듀얼 트리의 정점과 에지를 첨가하였다.

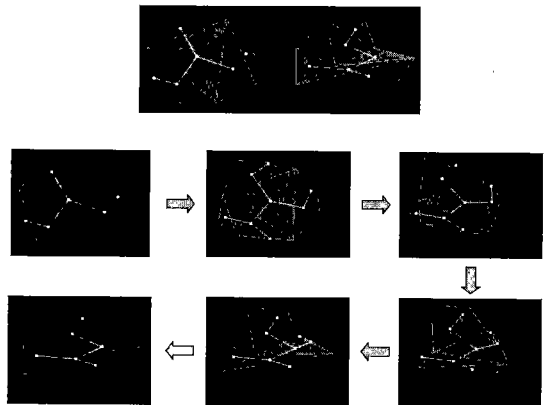


그림 14 실험 결과

이제 알고리즘 `Triangulation_Morphing`의 시간 복잡도를 고려하고 알고리즘의 중요한 개념인 유사 삼각 분할이 언제나 가능하다는 것을 정리하고 증명한다.

[정리 2]  $n, m$ 을 각각 주어진 원시 다각형과 목적 다각형의 정점의 개수라 할 때, 유사 삼각 분할을 이용한 단순 다각형의 형태 변환 문제는  $O(\max(n, m))$  시간에 해결된다.

(증명) 단계 1의 주어진 다각형을 삼각 분할하는 과정은 선형 시간에 분할이 가능한 방법을 이용하면 원시 다각형과 목적 다각형의 삼각 분할 문제를 각각  $O(n), O(m)$ 에 해결할 수 있으므로  $O(\max(n, m))$ 시간에 수행 가능하다[3][6]. 단계 2와 3은 삼각 분할된 다각형을 그에 대응하는 듀얼 트리로 구성하고 듀얼 루트 트리로 변환하는 단계로 모두 선형 시간 내에 수행할 수 있다. 단계 4는 원시 다각형을 목적 다각형과 유사 삼각 분할 되도록 에지 플립을 이용하여 삼각 분할의 일부를 변환시키는 과정이다. 정의 2에 의해 에지 플립으로 유사 삼각 분할을 만드는 단계 4의 전체 수행 시간은  $O(n)$ 이 된다. 마지막 단계는 실제 형태 변환이 일어나는 단계이다. 트리의 루트부터, 즉 다각형의 중심으로부터 삭제 및 삽입 연산을 이용하여 삼각형을 삭제하고 삽입하면서 모양을 맞춰준다. 단계 5 역시 원시 트리  $t_p$ 와 목적 트리  $t_q$ 를 루트부터 한 번만 따라가게 되고 두 트리의 구조가 같아질 때까지 수행해야 하므로  $O(\max(n, m))$ 에 처리할 수 있다. 이와 같이 유사 삼각 분할을 이용한 2차원 단순 다각형의 형태 변환 알고리즘은  $O(\max(n, m))$  시간 내에 수행 가능하다. □

[정리 3] 임의의 삼각 분할이 된 두 다각형은 에지 플립을 이용하여 언제나 유사 삼각 분할로 변경할 수 있다.

(증명) 두 삼각 분할을 유사 삼각 분할이 되도록 만드는 것은 그 삼각 분할을 트리로 표현한 두 개의 듀얼 루트 트리를 유사 트리로 만드는 것이다. 그림 12에서 살펴본 것처럼 듀얼 루트 트리 상에서 유사 노드가 되지 못하는 모든 노드는 최대 두 번의 에지 플립으로 모두 유사 노드로 만들 수 있다. □

듀얼 트리로부터 듀얼 루트 트리로 만드는 과정에서 트리의 중심을 구하기 위해 사용한 방법은 듀얼 트리의 경로상의 가장 긴 경로를 찾아 그 중심을 루트로 정하는 것이었다. 이 방법을 사용하면 다각형의 중심에 해당하는 삼각형이 듀얼 루트 트리의 루트로 결정된다. 하지만 다각형의 중심 부분에 해당하는 삼각형이 아닌 임의의 삼각형을 루트로 정해도 정리 3에 의해 언제나 유사 삼각 분할을 얻어낼 수 있다.

에지 플립은 두 개의 삼각형으로 이루어진 사변형이 볼록 사변형인 경우에만 가능하다. 따라서 때로는 볼록 사변형이 아니라서 에지 플립을 적용할 수 없는 경우도 있을 수 있다. 루트와 그 자식 노드 사이를 연결한 에지에서 이러한 상황이 일어나는 경우에는 루트의 자식 노드 중의 하나를 루트로 변경함으로써 이러한 문제를 해결할 수 있다. 루트가 아닌 일반 노드에서 이 상황이 일어나는 경우에는 에지로 연결된 두 노드 중 자식 노드의 위치를 에지 플립이 가능하도록 조금 변경함으로써 가능하게 할 수 있다. 본 알고리즘에서 사용하는 삼각 분할은 분할된 삼각형의 모양이나, 삼각형을 이루는 내각 등 일반적으로 좋은 삼각 분할의 기준이 되는 사항들이 중요시되는 것이 아니라 다각형의 모양과 속성을 유지하기 위한 삼각 분할이므로 그 모양이 변환될 원시 다각형의 정점의 위치가 변경된다고 하더라도 다각형의 모양이나 속성을 유지하는데 지장이 없다. 또한 삼각 분할이 된 각 삼각형은 각 정점의 절대 좌표로 저장되어 있지 않고 루트로부터 자식 노드로 서로의 상대적인 위치 관계를 유지하고 있으므로 하나의 정점의 위치를 변경해도 다각형 전체의 모양이나 속성을 그대로 유지할 수 있다.

### 5. 결론

본 논문에서는 유사 삼각 분할이라는 새로운 개념을 도입하여 단순 다각형에 대한 형태 변환 알고리즘을 제안하였다. 이 알고리즘은 수학적 함수를 사용하지 않고 트리의 연산을 이용하는 것으로 각 다각형의 정점의 개수를  $n$ 과  $m$ 이라고 할 때  $O(\max(n, m))$  시간에 수행 가능한 알고리즘이다. 이러한 연구를 바탕으로 하여 단순 다각형보다 복잡한 위상을 갖는 2차원 물체나 일반적인 3차원 물체에 대한 체계적인 형태 변환 알고리즘을 개발하여 실제 상황에 유용하게 적용될 수 있도록 확장하는 것이 향후 중요한 연구과제가 될 것이다.

### 참고 문헌

- [1] 신금립, "삼각 분할에 대한 문헌 연구", 이화여자대학교 전자계산학과 석사학위논문, 1992.
- [2] T. Beier and S. Neely, "Feature-based Image Metamorphosis," *Proc. of ACM Conference on Computer Graphics*, pp. 35-42, 1992.
- [3] B. Chazelle, "Triangulating a Simple Polygon in Linear Time," Technical Report CS-TR-264-90, Dept. of Computer Science, Princeton Univ., 1990.
- [4] L. Guibas and J. Hershberger, "Morphing Simple Polygons," *Proc. of the 10th ACM Symposium*



- on *Computational Geometry*, pp. 267-276, 1994.
- [5] J. Hershberger and S. Suri. "Morphing Binary Trees," *Proc. of the 6th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 396-404, 1995.
- [6] Joseph O'Rourke, *Art Gallery Theorems and Algorithms*, Oxford University Press, 1987.
- [7] F. P. Preparata and M. I. Shamos, *Computational Geometry*, Springer-Verlag, New York, 1985.
- [8] T. Sederberg and E. Greenwood, "A Physically Based Approach to 2-D Shape Blending," *Proc. of ACM Conference on Computer Graphics*, pp. 25-34, 1992.
- [9] T. Sederberg, Peisheng Gao, Guojin Wang, and Hong Mu, "2-D Shape Blending : An Intrinsic Solution to the Vertex Path Problem," *Proc. of ACM Conference on Computer Graphics*, pp. 15-18, 1993.



송 은 하

1996년 가톨릭대학교 수학과. 1999년 이화여자대학교 컴퓨터학과. 1999년 ~ 현재 한국과학기술연구원 위촉연구원.



이 주 회

1987년 이화여자대학교 전자계산학과 이학사. 1989년 이화여자대학교 대학원 전자계산학과 이학석사. 2000년 이화여자대학교 과학기술대학원 컴퓨터학과 공학박사. 2000년 3월 ~ 현재 일본 京都大 學校 공학연구소 초청연구원. 관심분야는 알고리즘 설계, 계산 기하학, 컴퓨터 그래픽스, 데이터 마이닝 등.

이 상 호

정보과학회논문지: 시스템 및 이론  
제 27 권 제 9 호 참조