

경제선 간략화를 이용한 일러스트레이션 기법

김혜선, 진희정, 유영중, 조환규

부산대학교 전자계산학과

e-mail: {hsukim, hjjin, yjyu, hgcho}@pearl.cs.pusan.ac.kr

The Illustration Technique Using Boundary Simplification

Hye-Sun Kim, Hee-Jeong Jin, Young-Jung Yu, Hwan-Gue Cho

Department of Computer Science

Pusan National University

요 약

일러스트레이션은 물체를 표현하는 최소한의 특징만을 가지고 사물을 표현하는 방법이다. 이를 자동화한 많은 연구들은 주로 많은 스트로크를 이용하여 실제와 유사한 일러스트레이션 결과를 생성하고자 하였다. 본 연구는 물체의 특징만을 표현할 수 있는 적은 스트로크를 이용하여 일러스트레이션을 생성하는데 목적을 둔다. 이를 위해 우리는 이전 연구[1]에서 간단히 추출된 경제선과 두 가지 스트로크 모핑 기법을 제시하였다. 본 논문에서는 물체의 특징을 표현하기 위해 유용한 경제선 추출 방법을 제시한다. 정확한 경제선의 추출은 스트로크와 더불어 최종 일러스트레이션 결과에 많은 영향을 미친다. 경제선 추출 방법과 이전 논문의 스트로크 모핑 방법을 이용한 일러스트레이션 결과는 본 논문의 실험결과로 주어졌다.

1 서론

일러스트레이션(illustration)은 대상 물체를 정확히 그리기 보다는 물체의 특징만을 추출하여 보여줌으로서 보는 사람이 물체를 쉽게 파악 할 수 있도록 한다.[6] 그러나 이 방법은 많은 수작업과 능숙한 일러스트레이터를 필요로 하기 때문에 생성이 쉽지 않은 단점을 가지고 있다.[2] 이 때문에 일러스트레이션을 자동으로 생성하는 시스템들이 많이 연구 개발되어왔다.

일러스트레이션 시스템은 그 입력양식에 따라 3차원 도형-기반의 시스템(3D geometry-based system)과 2차원 이미지-기반의 시스템(2D image-based system)으로 나뉘어진다.

3차원 도형-기반의 일러스트레이션 시스템은 물체의 3차원 정보를 입력으로 받아서 모델링(modeling)한 후, 2차원 평면으로 투영시켜 일러스트레이션을 생성한다. 따라서 이 방법은 일러스트레이션을 생성하기 위한 정보가 풍부하다. 그러나 3차원 정보를 모델링하고 물체 표면의 반사율과 특성을 파악하는데 복잡하고 많은 연산을 필요로 한다. 현재까지 연구된 도형-기반의 시스템들은 실시간 또는 비실시간으로 물체를 모델링하고 실행을 추출한 후, 내부의 스트로크는 텍스처(texture)를 입히거나 3차원 물체의 교차선을 이용하는 연구가 이루어졌다.[3, 4, 7]

2차원 이미지-기반의 일러스트레이션 시스템은 2차원 이미지를 시스템의 입력으로 받는다. 2차원 이미지는 물체 표면의 밝기 특성과 경제선이 이미지에 표현되기 때문에 복잡한 3차원 모델링과 표면의 반사율 특성을 파악하기 위한 연산을 하지 않아도 된다. 또한 입력으로 실제 사진, 그림, 패턴장 등을 다양하게 받아들일 수 있으며, 사용자가 입력을 직접 제어할 수 있는 장점이 있다.[6] 그러나 3차원 정보를 통해서만 알 수 있는 물체 표면의 여러 가지 특성들을 알 수 없기 때문에 2차원 이미지만으로는 물체의 특징을 충분히 살릴 수 없는 단점이 있다. 예를 들어 둥근 표면을 가지는 구를 그림 때에는 물체 표면을 채우는 스트로크(stroke)를 휘어지게 그려야한다. 그러나 2차원 이미지만으로는 물체 표면의 곡률정보를 알 수 없기 때문에 이것은 매우 어려운 작업이다. 현재까지의 연구에서는 주로 사용자가 개입해서 스트로크로 사용된 텍스처의 종류와 스트로크의 휘어진 정도를 지정해 주었다.[5, 6, 8]

현재까지 연구되어진 일러스트레이션 시스템들은 입력받은 정보들이 이용하여 입력과 최대한 유사하게 표현하기 위해 많은 수의 스트로크를 이용하여 결과를 생성하였다. 그러나 사진이나 교과서와 같은 실제 출판물에 사용되는 일러스트레이션들은 적은 스트로크로 사물의 특징만을 묘사하는데 중점을 둔다. 적은 스트로크를 이용하여 사물

을 표현하는 일러스트레이션의 몇 가지 특징을 요약하면 다음과 같다.

- 물체의 경제선이 확실하게 드러난다.
- 밝고 어두운 부분이 대조적으로 표현된다.
- 스트로크의 개수가 적다.
- 물체의 표면 곡률에 따라 스트로크가 휘어져있다.
- 스트로크들이 평행하게 군집을 이루어 그려지기도 한다.

본 논문에서는 위의 특징들을 고려하여 적은 수의 스트로크로 일러스트레이션을 생성하는 기법을 제시하고 있다. 적은 스트로크를 사용한 일러스트레이션에서는 표현하고자 하는 물체의 경제선과 스트로크의 방향이 물체를 인식하는데 중요한 역할을 한다. 적은 스트로크를 사용한 일러스트레이션 생성에 관한 이전 연구에서, 우리는 두 가지 스트로크 모핑 기법을 제시하였으나, 경제선 추출에 관한 연구가 부족하였다. 본 논문에서는 물체의 특징을 표현하기 위해 유용한 경제선 추출을 위한 한 가지 방법을 제안한다. 2장에서는 본 논문에서 사용된 경제선 추출방법에 대해 설명한다. 그리고 3장은 이전 연구에서 제시된 두 가지 모핑 기법에 대해 간단히 언급하고 있다. 4장에서는 본 연구의 경제선 추출 방법과 이전 연구의 모핑 기법을 적용한 몇 가지 일러스트레이션 결과를 보여준다.

2 이미지 경제선 추출

적은 스트로크를 사용하는 일러스트레이션은 많은 스트로크를 사용하는 일러스트레이션에 비해 스트로크만으로 사물을 표현하기가 어렵다. 본 논문에서는 이를 보완하기 위해 사물의 경제선을 추출하여 사용한다. 사물의 경제선은 스트로크와 더불어 물체를 표현하기 위해 사용되는 중요한 요소이다.

본 논문에서는 이전 방법에 비해 더욱 간결한 경제선 추출을 위한 한 가지 방법을 제안한다. 먼저 입력 이미지에서 라플라시안(Laplacian) 방법을 사용하여 경제선을 추출한다. 추출된 경제선은 많은 노이즈와 연결이 안된 부분들을 포함하고 있다. 이를 제거하기 위해 본 논문에서는 경제선 간소화와 경제선 재구성 방법을 사용한다.

2.1 경제선 간소화

경제선 추출을 위해 본 논문에서는 먼저 라플라시안 방법을 이용하여 경제점들을 추출한다. 추출된 경제점들은 경제선 간소화 작업과 재구성 작업을 위하여 폴리라인(polyline)의 집합으로 표현한다.

각 폴리라인에 대해 본 논문에서는 각과 길이의 두 가지 방법을 사용하여 경제선을 간소화한다. 첫 번째 방법은 폴리라인을 구성하

는 점들 사이의 각을 비교하는 것이다. 하나의 폴리라인에 저장되어 있는 세 점 p_{i-1}, p_i, p_{i+1} 사이의 각 $\angle p_{i-1}p_i p_{i+1}$ 이 사용자가 정의한 임계각 θ_{user} 보다 크다면, 그것은 하나의 직선위에 있는 것으로 간주하고, 폴리라인으로부터 점 p_i 를 삭제한다. 이 작업은 모든 폴리라인에서 사용자가 정의한 임계각 θ_{user} 보다 큰 각이 존재하지 않을 때까지 계속된다. 두번째 방법은 각 폴리라인에 저장되어 있는 점들간의 거리를 비교하는 것이다. 각 폴리라인에 저장되어 있는 점 p_i 와 p_{i+1} 사이의 길이가 사용자가 정의한 임계값 $Length_{user}$ 보다 작을 경우 점 p_{i+1} 을 삭제한다. 이것은 때때로 선으로 만들어주기 위해 필요한 작업이다. 이 방법을 수도(pseudo)코드로 나타내면 아래와 같다.

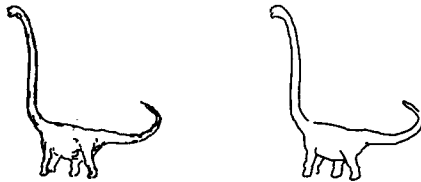
알고리즘: 경계선 간소화
입력: 폴리라인 집합 $\{P_1, P_2, \dots, P_n\}$, θ_{user} , $Length_{user}$
출력: 간소화된 폴리라인 집합 $\{P_1, P_2, \dots, P_n\}$

```

begin
// $\angle(p_{i-1}, p_i, p_{i+1}) =$  폴리라인  $P_k$ 의 세 점 사이의 각
while (각 폴리라인  $P_k$ 에 대하여)
begin
if ( $\angle(p_{i-1}, p_i, p_{i+1}) > \theta_{user}$ )
begin
폴리라인  $P_k$ 에서 점  $p_i$ 를 삭제한다.
end
end
end

// $Length(p_i, p_{i+1}) =$  점  $p_i, p_{i+1}$  사이의 길이
while (각 폴리라인  $P_k$ 에 대하여)
begin
if ( $Length(p_i, p_{i+1}) \leq Length_{user}$ )
begin
폴리라인  $P_k$ 에서 점  $p_{i+1}$ 를 삭제한다.
end
end
end
    
```

위의 알고리즘을 사용하여 간소화된 폴리라인들을 연결하면 직선의 집합으로 나타난다. 본 논문에서는 물체의 경계선을 부드럽게 연결하기 위해 베지어(Bézier) 곡선을 사용하여 이러한 직선들을 곡선으로 변환한다.



(a)이전연구[1]에 의해 추출된 경계선 (b)본 논문에서 추출된 경계선

그림 1: 이전연구에서 추출된 경계선과 본 논문에서 추출된 경계선

그림 1은 이전 연구에서 추출된 경계선과 본 논문에서 간소화하고 난 후의 경계선을 보여준다. 그림 1의 (a)는 이전 연구에서 이미지의 농도차이에 의해서만 추출된 경계선이다. 사진에 사용할 이미지처럼 단순하게 표현되어있지 않고 많은 불필요한 경계선들이 나타남을 볼 수 있다. 그림 1의 (b)는 본 논문에서 경계선 간소화 방법을 사용하여 추출한 경계선이다. 그림 1의 (a)와 비교해 볼 때, 불필요한 경계선들이 모두 제거되어 있고, 경계선들도 부드럽게 연결되어 있음을 알 수 있다.

2.2 경계선 재구성

경계선 간소화 과정을 통해 만들어진 경계선 이미지는 다시 재구성 과정을 거친다. 이전 단계까지의 경계선들은 실제로 이어져야 하는 선들이 끊어져 있고, 끊어져야 하는 경우의 선이 연결되어 있는 경우가 많다. 이 문제는 폴리라인 생성시에 현재의 점에서 이웃하는 점이 경계선일 경우 아무런 조건 없이 폴리라인에 저장하기 때문이다.

이 문제를 해결하기 위해 본 논문에서는 경계선 추출의 마지막에 경계선 재구성 과정을 거친다.

경계선 재구성 과정은 경계선 분리, 경계선 연결, 그리고 길이가 짧은 폴리라인을 삭제하는 세 가지 단계로 구성된다. 경계선 분리는 하나의 폴리라인에서 그 기울기가 급격히 변화되는 위치를 찾아 하나의 폴리라인을 두 개의 폴리라인으로 만들어 주는 과정이다. 경계선 분리는 이후 경계선을 베지어 곡선으로 변환할 때, 발생하는 문제를 제거하기 위해 필요한 단계이다. 경계선 연결은 각각의 폴리라인에서 사용자가 사진에 지정한 간격 사이에 있는 다른 폴리라인들과 그 기울기를 비교해보고, 기울기가 유사한 폴리라인들을 하나의 폴리라인으로 연결하는 과정이다. 이 과정은 연결되는 폴리라인이 나타나지 않을 때까지 계속된다. 폴리라인 재구성은 분리와 연결 과정이 끝난 후, 불필요한 짧은 폴리라인들이 있을 때, 사용자가 사진에 지정한 길이와 비교하여 그것보다 길이가 짧은 것을 노이즈로 간주하여 삭제한다.

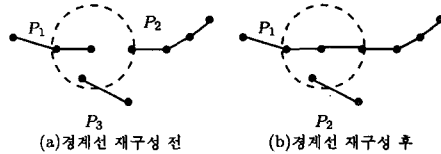


그림 2: 경계선 재구성

그림 2는 경계선 재구성 과정을 보여준다. 그림 2의 (a)와 같은 입력이 들어올 경우 (b)와 같이 경계선 재구성중 연결 과정의 결과이다.

3 스트로크 생성

본 시스템에서는 적은 수의 스트로크으로도 물체의 특징을 잘 나타낼 수 있도록, 물체의 표현 곡률을 따라 휘어진 스트로크들을 그린다. 또, 필요에 따라서는 이들을 일정한 방향으로 평행하게 모아서 그리기도 한다. 본 논문의 시스템은 이러한 시스템 스트로크를 생성하기 위해 사용자와 상호작용하여 샘플 스트로크(sample stroke)의 형태로 여러 정보들을 입력받는다. 입력된 샘플 스트로크들은 표현하고자 하는 물체의 특징을 충분히 살릴 수 있어야 하며, 이들을 모핑하여 물체 전체에 스트로크를 그린다.

3.1 샘플 스트로크

샘플 스트로크(sample stroke)이란 입력 이미지만으로는 부족한 물체의 특징 정보를 대신해서 사용자가 입력해주는 스트로크를 의미한다. 본 시스템은 물체의 경계선을 추출한 후 사용자가 샘플 스트로크를 입력할 수 있는 인터페이스(interface)를 제공해준다. 이 때 샘플 스트로크들은 물체의 특징을 충분히 살릴 수 있어야 한다. 예를 들어, 둥근 표면 위에 그려야 하는 스트로크는 둥글게 휘도록 그려주며, 거칠은 표면 위에 그려야 하는 스트로크는 짧게 그려준다. 샘플 스트로크는 나중에 모핑해서 사용하기 때문에 물체의 특징적인 부분에 몇개만 입력해주면 된다.

샘플 스트로크는 크게 모핑 스트로크와 방향 스트로크 두가지로 구분된다. 모핑 스트로크는 일러스트레이션에 직접 모핑해서 그려넣을 스트로크를 의미하며, 반드시 입력되어야 한다. 방향 스트로크는 스트로크들의 평행한 진행방향을 나타내는 것이며, 필요에 따라 입력해준다.

3.2 스트로크 모핑

일러스트레이션에 그려넣을 스트로크들을 사용자가 모두 지정해준다면 일러스트레이션의 질은 높아지지만, 이는 사람이 직접 그리는 경우와 같아서 효율성이 떨어진다. 본 시스템에서는 효율성을 높이기 위해 사용자가 입력해주는 몇 개의 샘플 스트로크들을 모핑해서 일러스트레이션에 그릴 스트로크들을 생성한다. 사용자가 몇 개의 샘플 스트로크들을 적당한 위치에 입력해주면 나머지 위치의 스트로크들은 시스템이 자동으로 생성해준다.

스트로크를 모핑하는 방법은 스트로크의 모양 중심으로 모핑하는 모양 중심 모핑과, 스트로크의 흐름 중심으로 모핑하는 흐름 중심 모핑이 있다. 모양 중심 모핑은 샘플 스트로크들의 모양을 최대한 유지하며 모핑하기 때문에 동물의 털이나 나뭇잎 모양 등의 스트로크 생성에 효율적이다. 흐름 중심 모핑은 구와 같은 둥근 곡면에 스트로크를 그릴 때 곡면의 곡률 흐름에 맞춰 스트로크를 생성한다.[1] 그림 3은, 같은 샘플 스트로크로 두가지 방법을 써서 모핑했을 때의 결과이고, 그림 4는 두 가지 방법을 실제 일러스트레이션에 적용한 결과이다.



그림 3: (a)샘플 스트록 (b)모양 중심 모핑 결과 (c)흐름 중심 모핑 결과

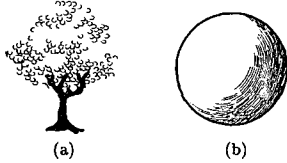


그림 4: (a)모양 중심 모핑 예(샘플 스트록 9개 입력) (b)흐름 중심 모핑 예(샘플 스트록 3개 입력)

4 실험결과

본 논문은 적은 수의 스트록만으로 사물의 특징을 표현하는 일러스트레이션을 만들기 위해 입력된 물체의 경계선을 추출하여 일러스트레이션에 적절하도록 간소화시키는 방법과, 샘플 스트록을 모핑하여 영역에 그리는 방법을 제안한다. 시스템 구현과 실험은 Pentium III 600에서 Visual C++ 6.0을 이용하였다.

그림 5는 본 논문에서 제시한 방법으로 구현된 시스템으로 테스트한 결과를 보여준다. 적은 수의 스트록을 그리면서도 입력된 이미지의 특징을 충분히 살림을 확인할 수 있다.

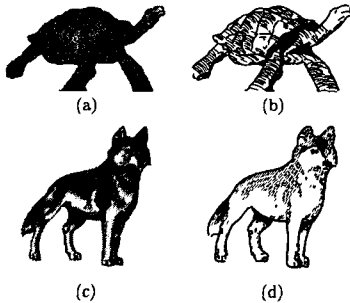


그림 5: (a)(c): 입력 이미지 (b)(d): 일러스트레이션 결과 이미지

5 결론 및 향후과제

본 논문은 사전이나 교과서와 같은 실제 출판물에 사용되는 일러스트레이션을 생성하기 위한 방법을 제시하고 있다. 이전의 일러스트레이션 기법들이 실제와 유사한 결과를 생성하기 위해 많은 스트록을 사용한 것과는 달리, 적은 스트록과 물체의 경계선을 이용하여 물체의 특징을 표현함으로써 간결한 일러스트레이션 결과를 생성하는 방법을 제시한다.

본 논문의 이전 연구[1]에서 적은 스트록을 사용하기 위해 두 가지 스트록 모핑 기법을 제안하였으며, 본 논문에서는 물체의 특징을 표

현하는 경계선을 간결하게 추출하는 방법을 제시하였다. 본 논문에서 제시한 방법이 이전의 일러스트레이션 방법들에 비해 적은 수의 스트록으로 물체의 특징을 표현할 수 있다는 장점을 가지고 있으나 몇가지 개선의 여지가 있다.

- 복잡한 이미지의 경우로 자동으로 경계선을 추출하는 것은 어려운 문제이다. 따라서 현재의 연구로는 사용자와의 상호작용을 거쳐야 하며 이는 시간과 노력을 필요로한다. 따라서 복잡한 이미지에 대한 자동화된 경계선 추출 방법의 연구가 필요하다.
- 현 시스템은 사용자로부터 샘플 스트록을 입력받아 결과를 생성한다. 따라서 사용자에 따라 서로 다른 결과를 획득할 수 있으며, 좋지 않은 결과를 획득할 수도 있다. 따라서 최소한의 결과를 얻을 수 있도록 사용자의 입력을 보정하는 방법에 대한 연구가 요구된다.

참고문헌

- [1] 김혜선, 진희정, 유영중, 조환규. 스트록 모핑을 이용한 간결한 pen-and-ink 일러스트레이션. 한국그래픽스 학회 제 6권, 2호, 2000.
- [2] 금복희, 이도훈, 조환규. 2차원 이미지를 위한 자동화된 펜과 구 현. 한국정보과학회 학술발표집 제 22권, 2호, pages 505-508, 1995.
- [3] Oliver Deussen, Jorg Hamel, Andreas Raab, Stefan Schlechtweg, and Thomas Strothotte. An illustration technique using hardware-based intersections and skeletons. *Graphics Interface '99*, pages 175-182, June 1999.
- [4] Lee Markosian, Michael A. Kowalski, Samuel J. Trychin, and Lubomir D. Bourdew. Real-time nonphotorealistic rendering. *Proc. SIGGRAPH 97*, pages 415-420, August 1997.
- [5] Michael P.Salisbury, Sean E.Anderson, Ronen Barzel, and David H.Salesin. Interactive pen-and-ink illustration. *Proc. SIGGRAPH '94*, pages 101-108, July 1994.
- [6] Michael P.Salisbury, Michael T.Wong, John F.Hughes, and David H.Salesin. Orientable textures for image-based pen-and-ink illustration. *Proc. SIGGRAPH '97*, pages 401-406, August 1997.
- [7] Georges Winkenbach and David H.Salesin. Rendering parametric surfaces in pen-and-ink. *Proc. SIGGRAPH 96*, pages 469-476, August 1996.
- [8] Georges Winkenbach and David H.Salesin. Computer-generated pen-and-ink illustration. *Proc. SIGGRAPH '94*, pages 91-100, July 1994.