애니메이터의 예술적 특성을 반영한 라인패턴 자동생성 기법

Computer Assisted Line Pattern Drawing with Animator's Artistic Characteristics

임 훈, 이종원
세종대학교 컴퓨터 공학부

Hun Im(terehun@naver.com), Jong-Weon Lee(jwlee@sejong.ac.kr)

요약

본 논문에서는 완성된 수묵화나 여러 회화적 효과를 지닌 그림을 가지고 그곳에 있는 라인 패턴을 자동으로 추출한 후, 자동으로 스케치에 그 라인 패턴을 입혀주는 방식을 연구하였다. 이를 이용해 전체 프레임 중간에 애니메이터가 완성한 키 프레임을 받고, 다른 프레임들과의 대응 라인을 찾은 후 패턴을 자동으로 입혀 중간 프레임들을 얻어내는 결과를 보여준다.

■ 중심어 : | 애니메이션 | 키 프레임 | 패턴 | 수묵화 | 라인패턴 자동화 |

Abstract

Computer-assisted animation production tools have an important role in animation production industries. They could reduce the production cost and time, but they rarely copy artistic characteristics of an animators. In this paper, a new computer-assisted line pattern drawing system is proposed. The system automatically copies artistic characteristics of an animator and applies them to computer-generated animations. The artistic characteristics are captured from key frames drawn by an animator. The result shows that the presented approach can copy artistic characteristics from key frames well and reduce time and cost for producing artistic animations.

■ Keyword: | Animaition | Key Frame | Pattern | Sumi Drawing | Automatic Line Pattern Drawing |

I. 서 론

1. 연구 배경 및 서론

2D 애니메이션 산업의 기술은 날로 발전하고 있으며 그로 인해 산업에서 요구 하는 자동화 기술 역시 다양하고 복잡해지고 있다.

현재의 컴퓨터를 이용한 애니메이션 자동화 기술은 키 프레임에서 사이 프레임을 자동화로 생성해주는 [1][2] 방식이나 키 프레임 사이 적용 프레임에 컴퓨터를 이용한 자동 컬러링 방법[3][4]이 연구 되었다.

그렇지만 최근 2D 애니메이션 산업의 기술 발전으로 회화적인 느낌이 들어간 애니메이션들이 많이 등장하게 되었다. 특히 중국, 한국, 일본 등 아시아 애니메이션 시장에서 단순한 라인과 색채 패턴을 가진 애니메이션뿐만 아니라 수묵화와 같은 다양한 효과를 지닌 애니메이션들이 속속 등장하고 있다.

* 본 연구는 한국전자통신연구원 광통신연구센터의 광가입자망 서비스개발 실험사업 연구지원을 받아 수행되었음.

접수번호 : #060310-002
접수일자 : 2006년 03월 10일
심사완료일 : 2006년 03월 22일
교신저자 : 이종원, e-mail : jwlee@sejong.ac.kr

이런 애니메이션의 경우 일반 애니메이션에 비해 애니메이터의 공급 부족으로, 한 프레임에 드는 비용이 비싸고, 작업 속도도 느리다.

본 논문에서는 고 비용과 느린 작업 속도의 문제를 극복하기 위해 스케치 라인에 다양한 패턴을 입히면서도 애니메이터가 패턴을 입히는 것과 유사하게 입히는 방법을 소개 하고자 한다.

기존에 있는 라인 패턴을 입히는 대부분의 프로그램들은 기존에 이미 존재하는 라인 패턴을 입히는 방식으로 [5][6] 개개인의 애니메이터가 가지고 있는 예술적 특성을 살리기 힘든 면이 있다.

이 논문에서는 애니메이터의 예술적 특성을 고려하지 않고 패턴이 그려지는 것을 극복하기 위해서 애니메이터가 작업한 것을 기초 자료로 이용해 일정량의 프레임을 자동화 시키는 방법을 개발하였다.

애니메이터의 스케치를 세션화 시킨 후 그 세션화된 프레임과 이미 완성된 키 프레임의 대응선분을 찾아낸 후, 각 대응 선분에 맞는 라인 패턴을 입히게 된다. 이 방법의 특징은 애니메이터의 예술적인 특성을 컴퓨터가 생성한 이미지에 담았다는 것이다.

2. 시스템 개관

본 논문에서 설명하는 방식의 패턴 드로잉을 위해서는 [그림 1]과 같이 애니메이터에 의해서 만들어진 키 프레임의 스케치, 키 프레임과 적용될 스케치들이 필요하다. 키 프레임은 키 프레임 스케치위에 애니메이터가 회화적 패턴을 입혀 생성함을 가정한다.

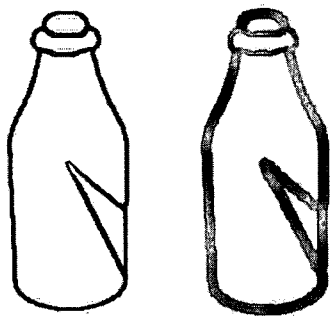

그림 1. 키 프레임 스케치와 키 프레임

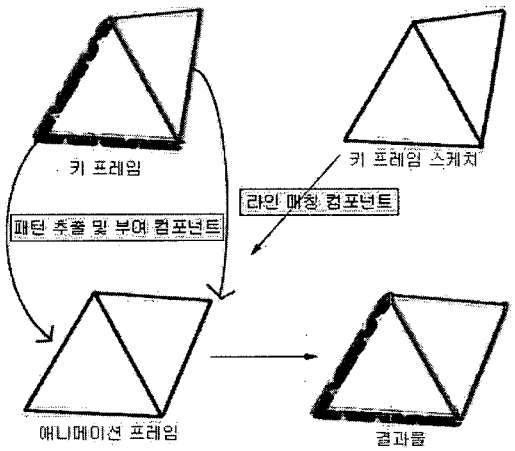

그림 2. 패턴 자동부여 플랫폼

[그림 2]와 같이 키 프레임과 키 프레임의 스케치를 이용하여, 새로운 애니메이션 프레임에 패턴을 부여 하게 된다. 이때 두 가지 컴포넌트를 사용하게 되는데, 패턴 추출 및 부여를 하는 부분과 영역 매칭을 통한 매칭 컴포넌트를 이용하게 된다.

시스템은 먼저 '키 프레임 스케치'와 '적용 프레임'을 쇄선화 하여 '쇄선화된 키 프레임 스케치'와 '쇄선화된 적용 이미지'를 만든다. 각각의 쇄선화된 이미지는 라인 세션화를 통해 각각의 라인들을 구분한다.

그 후 라인 매칭 컴포넌트가 쇄선화된 이미지 두 개에서 구분된 영역을 서로 비교하여 영역들을 매칭시키고, 매칭된 영역들을 이용하여 각각의 라인을 매칭 시킨다.

이렇게 매칭된 라인을 기반으로 패턴 추출 및 부여 컴포넌트에서 '쇄선화 키 프레임 스케치'를 백터화 한 후 백터화의 결과를 이용하여 키 프레임에서 패턴을 추출한다. 추출된 패턴은 '쇄선화된 적용 이미지'를 백터화 한 후에 각도 보간과 픽셀 보간을 하게 된다. 보간된 백터화 결과를 기반으로 적용 이미지에 부여 되게 된다.

본 논문의 2장에서는 어떻게 패턴을 추출하고 패턴을 그리게 되는지 설명하고 자동 애니메이션 프레임 부여 방법에 대해서도 설명한다. 3장에서는 실험결과와 결론을 내린다.

II. 본 론

1. 자동 패턴 드로잉

1.1 이미지 쇄선화(Image Thinning)

패턴 드로잉의 첫 번째 단계는 이미지 쇄선화이다. 스케치된 이미지를 가지고 이미지 쇄선화를 하게 된다. 쇄선화를 하는 이유는 기본 스케치에 있는 노이즈를 제거하고, 정확한 라인을 얻어내기 위해서이다.

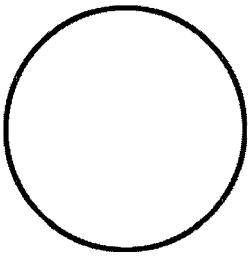

그림 3. 원본 이미지

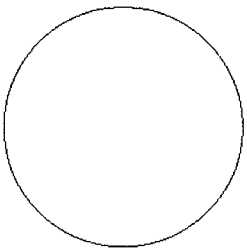

그림 4. 쇄선화된 이미지

이 작업을 위해서 Zhang Suen의 쇄선화 알고리즘을 사용하였다[7].

1.2 라인 세션화(Line Sessioning)

이렇게 쇄선화된 이미지를 각각의 라인으로 구분하기 위해서 선을 세션화 한다. 쇄선화된 이미지에서 실제 점을 찾게 되면, 그 라인의 8방향을 검사하면서 연결된 픽셀들의 정보를 모아간다. 이 과정에서 픽셀이 분할되거나 끊기는 경우 해당 픽셀은 하나의 선분의 시작점이나 끝점이 된다.

[그림 5]와 같은 경우 총 3개의 라인 세션이 구분 된다. 빨간 픽셀인 ①②③은 세션의 시작점이며, 파란 픽셀인 ④⑤⑥은 세션의 끝이다.

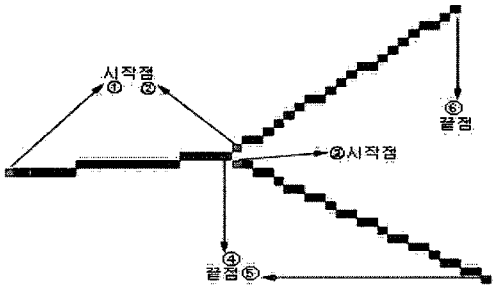

그림 5. 라인 세션화

이런 방식은 애니메이터가 직접 스케치를 그리는 순서와는 다르지만 패턴을 아티스트가 직접 부여한 곳에서 얻어와 적용 이미지에 부여하기 때문에 라인 세션이 나뉘어 있더라도 추출된 패턴은 연속적이기 때문에 문제가 되지 않는다.

1.3 영역 매칭(Area Matching)

키 프레임 스케치와 패턴이 입혀지지 않은 적용 프레임간의 라인 세션들을 매칭시키기 위해서 영역을 이용한다. '쇄선화된 키 프레임 스케치'와 '쇄선화된 적용 이미지'의 영역을 각각 구분한 다음 구분된 각각의 이미지에서 영역과 해당 영역 외각의 각 라인 세션들을 매칭시킨다. 그 후에 '쇄선화된 키 프레임 스케치'의 영역과 '쇄선화된 적용 이미지'의 영역을 비교하여 매칭시키고 그 결과를 이용해서 구분된 영역에 속하는 라인들을 매칭시키는 방법을 이용하였다.

쇄선화된 이미지에서 영역을 구분해 내기 위해서 범람채우기(flood fill) 방식을 이용하였다.

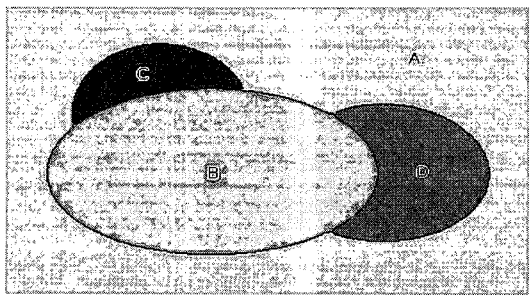

그림 6. 영역 구분

[그림 6]의 경우 회색(A)이 배경이며 붉은색으로 차 있는 부분(B, C, D)이 구분된 영역들 이다.

영역을 구분한 후에 영역의 매칭은 Seah의 방식[3][4]을 참조하여 구현하였으며 아래의 프로세서를 통해서 구한다.

가. 영역 넓이에 따라 정렬한다.

나. 영역 넓이와 겹치는 영역의 크기를 이용해 일치된 영역을 찾는다.

다. 위의 과정을 통해서 일치 영역을 발견하지 못한 영역의 경우 주변 영역들의 관계를 비교하여 일치 영역을 찾는다.

영역의 매칭을 위해 넓이에 따라 정렬하는 이유는 영역의 매칭이 상당한 지연이 걸리는 작업이기 때문이다. 넓이에 따라 정렬함에 따라 빠른 프로세스가 가능하게 된다.

$$TDV(i,i') = aADV_a(i,i')^2 + bADV_o(i,i')^2$$

$$ADV_a(i,i')^2 = \frac{|A(i) - A(i')|}{A(i)},$$

$$ADV_o(i,i')^2 = \frac{|A(i) - Overlap(i,i')|}{A(i)}$$

$A(i)$: i 번째의 영역 면적,

$Overlap(i, i')$: i와 i' 영역의 겹치는 면적 크기

a, b 각각의 트래쉬홀드 값.

식 1. 영역 구분

정렬된 영역들을 넓이와 겹치는 영역을 이용해 일치하는 영역을 찾게 되는데, [식 1]을 이용하여 TDV 값이 일정 값보다 높으면 일치한다고 판단한다. 실험에서는 넓이 값의 크기보다 겹치는 면적의 크기가 더 중요하기 때문에 겹치는 면적에 대한 트래쉬 홀드 값인 b값을 더 높게 주었다.

넓이와 겹치는 면적을 이용해서도 일치 영역이 발견되지 않는 경우, 주변 영역이 일치되는 영역이 있는지 찾게 된다. 만약 주변 영역이 일치되는 경우 TDV 값을 높여줘 일치 가능성을 높여준다.

1.4 패턴 추출

1.3 영역 매칭에서 매칭된 라인 세션들에서 패턴을 추출하고 부여하려면 쇄선화된 키 프레임 스케치의 각각의 픽셀에 대해 벡터화를 해야 한다. 벡터화란 라인에서 완전히 수직인 벡터를 찾는 것을 말한다. 이런 벡터화를 통하여 쇄선화된 이미지에 패턴을 입히는 위치를 얻어오거나 키 프레임에서 패턴을 얻어올 위치를 찾게 된다.

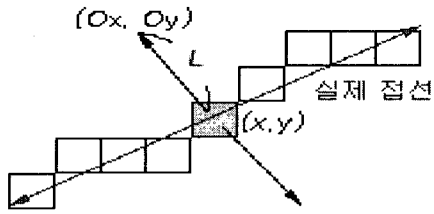

그림 7. 벡터화

[그림 7]에서처럼 벡터가 시작되는 픽셀에서 근접된 8개 픽셀과의 평균 기울기를 구하여 접선을 구한다. 그 접선의 기울기가 a 인 경우 a의 직교가 되는 벡터를 구한다. 각 벡터의 길이를 L이라 하면 이것은 패턴의 두께라고 볼 수 있다. x, y를 접선의 중점이라고 보면 실제로 패턴이 입혀질 좌표 Ox, Oy는 식2와 같다. 식 2는 점 x, y를 알고 기울기 $1/a$를 가지고 접선의 방정식을 푼 것이다.

$$C = x^2 - \frac{L^2}{a^2 + 1}$$

$$Ox = \frac{2x \pm \sqrt{4x^2 - 4C}}{2}$$

$$Oy = a(Ox - x) + y$$

식 2. 벡터화 공식

패턴의 추출은 벡터화 한 결과를 가지고 아티스트가 입힌 패턴 정보를 얻어온다. 패턴을 추출할 때는 [식 2]의 벡터화 공식에서 벡터의 길이 L은 아티스트가 입힌 패턴의 너비로 한다.

그림 8. 패턴 추출 이미지

그림 9. 추출된 패턴

패턴의 소스가 되는 [그림 8] 이미지에서 패턴 추출과정을 통해서 얻어온 패턴이 [그림 9]이다.

1.5 패턴 부여

본 논문에서 패턴을 부여하는 방식은 패턴을 라인으로 나누어 벡터의 크기 L에 비례하는 패턴의 위치를 획득하여 부여하는 방식이다.

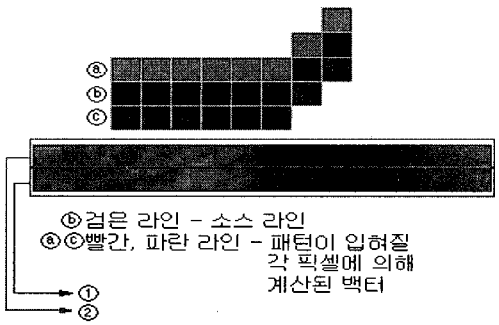

그림 10. 패턴 부여 방식

[그림 10]에서 a-파란색과 b-빨간색 라인의 경우 벡터화와 보간을 통해서 생성된 목표 라인이다. 이 생성된 픽셀을 아래의 패턴에서 각각의 색깔로 표시된 ①과 ② 박스 부분의 픽셀들을 대응 시켜준다. 패턴의 길이가 목표 라인의 길이 보다 짧은 경우는 패턴의 길이와 패턴이 입혀질 라인 길이의 비례식을 통해 대응 픽셀을 찾는다.. 반대로 패턴이 목표 라인보다 길 경우 자연스러운 패턴 부여를 위해 1:1로 대응 시키고, 목표 라인의 길이 이상의 패턴은 사용하지 않는다. 비례식을 이용하여 패턴 자체를 축소시킬 수도 있으나, 높이는 두고 너비만 줄여야 하기 때문에 심각한 엘리어싱을 발생시킨다.

순수한 벡터화 과정을 원에 적용한 결과는 [그림 11]과 같다.

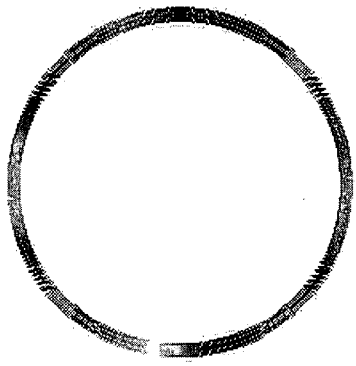

그림 11. 기본적 벡터화 결과

실질적인 결과에 많은 오차가 있는 것은 실제 평균 기울기를 구할 때 y 축과 평행으로 올라가는 픽셀들이 있는 경우 x축과의 사이 각이 90도로 기울기가 ∞ 이기 때문에 다른 접선의 기울기와 평균을 냈을 때 평균 기울기를 구할 수 없다. 그러므로 89도로 임의 조정하여 평균을 내었다. 하지만 픽셀이 y 축과 평행으로 올라가는 부분을 포함하여 평균 기울기를 구해야 하는 픽셀의 경우 기존의 픽셀에 비해 크게 증가 하게 되고, 그로 인해 급격한 차이를 보이게 된다. 또한 접선을 구하는 두 개의 픽셀의 기울기가 완만하지 않고, 큰 차이를 보이게 되면 기울기의 차이 역시 커지게 되는데, 두 픽셀에서 생성된 벡터의 거리가 멀 때 결과에 오차가 나오게 된다. 이런 문제점을 해결하기 위해서 기울기 보간을 이용한다.

기울기 보간 이란 **A** 벡터와 **B** 벡터 사이의 픽셀의 거리 d 가 실제 픽셀에서 8방향으로 한 픽셀이상 이면 **A** 벡터의 기울기와 **B** 벡터의 기울기의 사이 기울기를

이용해 C 벡터를 임의로 생성해주는 것을 말한다. C 벡터는 A 벡터와 B 백터 사이의 비어져 있는 픽셀을 채워주게 된다. 만약C 벡터와 A사이가 d 가 픽셀 길이로 다시 8방향으로 한 픽셀 이상이면 벡터 C'벡터를 만들어 사이의 픽셀을 재 보간 한다.

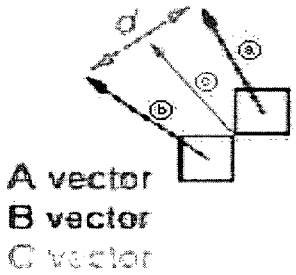

그림 12. 기울기 보간의 예

이런 보간 방법으로 y 축과 평행하여 x 축과의 기울기가 90도가 나와 평균 기울기를 정확하게 구하지 못하는 문제도 사이 기울기를 구 할 수 있기 때문에 보정된다. 기울기 보간을 적용한 후의 결과는 [그림 13]과 같다.

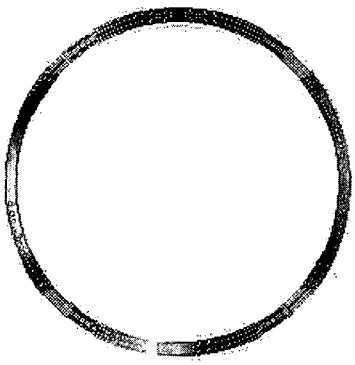

그림 13. 기울기 보간 후의 결과

기울기 보간 후 기존의 결과보다 훨씬 좋아졌지만 아직도 한 픽셀이 떨어지는 문제를 발생한다. 이것은 정확하게 45도 각도를 가진 즉 1의 기울기를 가진 접선으로 인해서 생기는 문제로 벡터의 결과 값은 소수점으로 나오는데 반해, 픽셀 값은 정수 값이기 때문에 어떤 각도에서도 채워지지 않는 픽셀들이 생기게 된다.

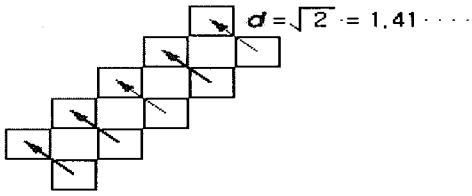

그림 14. 접선의 기울기가 1인 경우

기본적으로 d의 값이 정수로 이루어 져있고, 또한 픽셀도 정수로 이루어 져있기 때문에 각도가 45도 인 경우 사이에 비는 픽셀이 생기게 된다. 이것은 각도 보간만을 이용해서는 해결되지 않는 문제다. 다행히도 이런 문제는 4방향에 모두 픽셀이 차있음으로 픽셀 보간을 통해서 처리하게 된다.

픽셀 보간은 x, y 결과 값에 패턴을 드로잉 하기 전에 4방향 검색 후 비어 있는 픽셀을 찾는다. 만약 픽셀이 비어있을 경우 빈 픽셀의 4방향이 모두 차있는 경우에만 4방향의 픽셀들의 평균을 해당 픽셀에 넣는다. [그림 15]은 픽셀 보간 후 원에 패턴을 입힌 이미지이다.

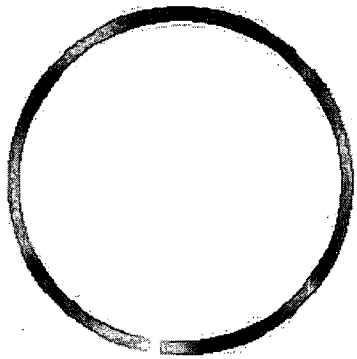

그림 15. 픽셀 보간 후 결과

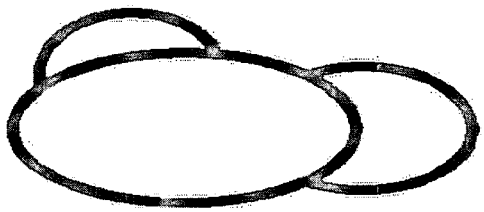

그림 16. 소스 이미지

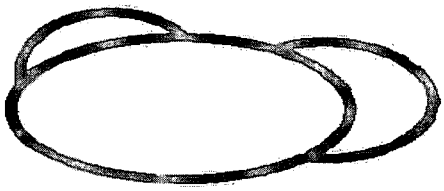

그림 17. 자동 패턴을 드로잉 결과

[그림 16]과 [그림 17]은 약간의 차이를 가진 이미지로써 키 프레임에서 패턴을 획득하여 스케치만 있는 적용 프레임에 패턴을 자동으로 드로잉한 결과 이다.

2. 애니메이션 프레임 드로잉의 자동화

실제 애니메이션 제작 시에 패턴이 입혀져 있는 키 프레임을 기반으로 각각의 스케치된 프레임을 적용 시키게 된다. [그림 18]과 같이 헤드 키 프레임 (Head Key Frame) 과 마지막 키 프레임 (Tail Key Frame) 이렇게 두 개의 키 프레임이 있으며 키 프레임 두 장 사이에 들어오는 프레임 스케치들이 적용 이미지가 된다. 적용되는 프레임 스케치들은 프레임 순서에 따라 더 가까운 쪽의 키 프레임에서 패턴을 얻어와 적용한다. 만약 6개의 프레임이 있다면 앞에 3 프레임은 기본적으로 헤드 키 프레임에 뒤에 3프레임은 마지막 키 프레임에서 패턴을 얻어온다.

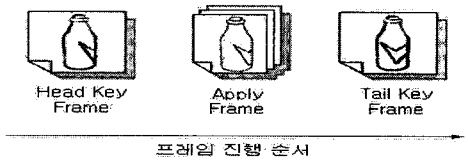

그림 18. 애니메이션 프레임 드로잉 자동화

각각의 적용 프레임 스케치는 바로 전 스케치의 영역과 비교하여 영역을 매칭 시킨 후 적용 프레임들을 검색해 키 프레임과의 링크되어 있는 라인을 찾아 키 프레임의 패턴을 획득하고 적용 시킨다.

두 개의 키 프레임을 기반으로 하기 때문에 한쪽의 키 프레임에 없는 영역이 다른 쪽 키 프레임에는 있을 수 있다. 헤드 키 프레임에 더 가까운 적용 프레임에 없는 영역이 마지막 키 프레임의 영역에 있다면 패턴을 추출하는 키 프레임을 마지막 키 프레임으로 변환하여 패턴을 추출, 부여 하게 된다.

III. 결 론

1. 실험 결과

[그림 19]는 실제로 키 프레임 두 개에서 영역 매칭을 통한 라인 매칭을 하고, 패턴을 추출 부여한 결과이다. 두 개의 키 프레임인 헤드 키 프레임과 마지막 키 프레임을 가지고 중간의 4개 프레임을 채웠다. 결과적으로 키 프레임의 느낌을 최대한 살려서 패턴이 부여 되었고, 프레임이 큰 변화가 없을수록 좋은 결과를 보였다.

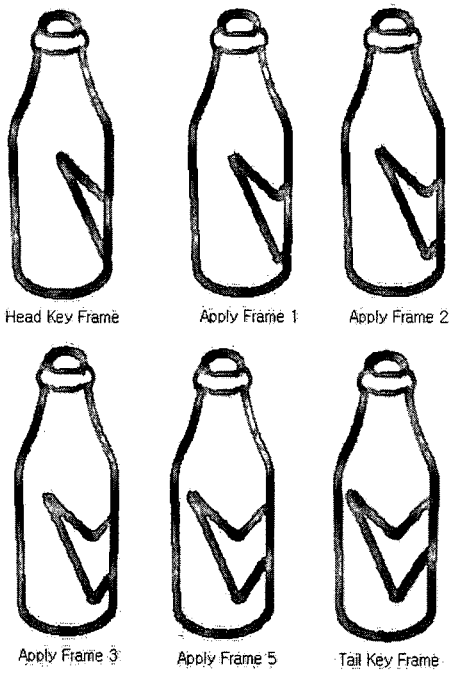

그림 19. 실험 결과

2. 결론

본 논문에서 소개한 방식은 기존에 있었던 패턴 부여 방식을 단순 이미지가 아닌 아티스트의 작업물을 근거로 사용하여 더 회화적인 패턴을 스케치 된 이미지에

아티스트의 특성에 맞게 부여 할 수 있게 되었다.

실질적으로 산업에서 사용하기 위해서는 다음부분의 보완점이 필요하다. 첫째 현재는 영역화 되어 있는 라인들만을 자동적으로 패턴을 부여 할 수 있다. 영역을 기반으로 다음 에는 영역화 되어 있지 않은 라인들도 패턴을 부여 할 수 있어야 할 것이다. 키 프레임에 없었던 새로운 영역이 들어 간 경우도 처리해줘야 한다.

둘째 패턴을 입히는 라인이 매 프레임 마다의 차이를 보이는데, 이 차이로 인해서 원하는 구역에 정확한 결과가 인식 되지 않는 경우가 있다.

마지막으로 현실적으로 애니메이션에서 사용되는 이미지는 상당히 크다. 그런 큰 이미지를 처리해야 하기 때문에 속도에 대한 성능도 더욱 향상 시켜야 한다.

참 고 문 헌

[1] N. Burtnyk and M. Wein, "Interactive Skeleton Techniques for Enhancing Motion Dynamics in Key Frame Animation," Communications of the ACM, 19, Oct., 1976.

[2] J. Lu, H. S. Seah, and F. Tian "Computer-Assisted Cel Animation: Post-processing After Inbetweening," Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia, Feb., 2003.

[3] H. S. Seah and T. Feng "Computer -assisted coloring by matching line drawings," The Visual Computer, 2000.

[4] J. Qiu, H. S. Seah, F. Tian, Q. Chen, and K. Melikhov, "Computer -Assisted Auto Coloring by Region Matching," Proceedings of the 11th Pacific Conference on Computer Graphics and Applications, Oct., 2003.

[5] S. C. Hsu, H. Irene, and H. LEE "Drawing and Animation Using Skeletal Strokes," Proceedings of the 21st annual conference on Computer graphics and interactive techniques, July, 1994.

[6] S. C. Hsu, I. H. H. Lee, and N. E. Wiseman "SKELETAL STROKES," Proceedings of the 6th annual ACM symposium on User interface software and technology, Dec., 1993.

[7] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Communications of the ACM,* March, 1984.

저 자 소 개

임 훈(Hun Im) 준회원

- 2004년 7월 : 세종대학교 컴퓨터공학 (공학사)
- 2004년 8월~현재 : 세종대학교 컴퓨터 공학부 석사

<관심분야> : 비실사렌더링, 게임 기획

이 종 원(Jong-Weon Lee) 정회원

- 1989년 6월 : B.S. in wj Ohio University, Ohio, USA
- 1991년 6월 : M.S. in Electrical Engineering, Univ. of Wisconsin, Madison, USA
- 2002년 5월 : Ph.D. in Computer Science, Univ. of Southern California, California, USA
- 2002년 8월 : Assistant Porfessor of Digital Contents, Sejong University, Seoul, Korea

<관심분야> : 컴퓨터 그래픽스, 증강현실, 컴퓨터 게임, HCI