

객체의 움직임 해석을 이용한 회화적 스트로크 생성 방법

(Painterly Stroke Generation using Object Motion Analysis)

이 호 창[†] 서 상 현^{**} 류 승 택^{***} 윤 경 현^{****}
(Hochang Lee) (Sanghyun Seo) (Seungtaek Ryoo) (KyungHyun Yoon)

요 약 기존의 회화적 렌더링은 영상 기울기(image gradient) 정보를 사용해 스트로크의 방향, 크기, 길이 등을 결정하였다. 영상 기울기는 객체의 모양을 표현하는데 있어서 유용한 정보이지만 풍경화에서 표현되는 객체(물, 나무 등)의 현재 움직임이나 흐름을 나타낼 수 없는 한계를 가진다. 실제 화가들의 그림에서, 객체의 실제 움직임에 기반한 브러시 스트로크는 관찰자에게 객체의 움직임을 보다 쉽게 인지할 수 있게 하며, 그림이 그려질 당시의 느낌을 보다 생동감 있게 전달할 수 있는 장점을 가진다. 본 논문에서는 풍경화의 주 대상이 되는 객체들의 움직임 정보를 기반으로 동적인 객체의 움직임을 효과적으로 표현할 수 있는 회화적 스트로크 생성에 관한 연구를 제안한다. 이를 위해 동일한 시점을 가진 순차영상 집합으로부터 표현하고자 하는 장면의 움직임 정보(크기, 방향, 편차)를 추출한다. 그리고 움직임의 크기가 큰 영역은 움직임의 방향에 기반하여 스트로크를 생성하며 움직임의 정보가 작은 영역은 영상 기울기 값을 기반으로 방향을 결정한다. 우리의 알고리즘은 사실적인 움직임의 방향을 표현하는데 유용하며 이는 풍경화 영상을 렌더링 하는데 유용하다.

키워드 : 회화적 렌더링, 스트로크 생성, 움직임 방향

Abstract Previous painterly rendering techniques normally use image gradients for stroke generation. Although image gradients are useful for expressing object shapes, it is difficult to express the flow or movements of objects of objects. In real painting, the use of brush strokes corresponding to the actual movement of objects allows viewers to recognize objects' motion better and express the liveliness of the objects much more. In this paper, we propose a novel painterly stroke generation algorithm to express dynamic objects based on their motion information. We first extract motion information (magnitude, direction) of a scene from a set of image sequences from the same view. Then the motion directions are used for determining stroke orientations in the regions with significant motions. Where little motion is observed, image gradients are used for determining stroke orientations. Our algorithm is useful for realistically and dynamically representing moving objects.

Key words : Painterly rendering, Stroke generation, Motion direction

* 본 논문은 교육과학기술부의 재원으로 한국과학재단의 지원을 받아 수행된 연구
구입(No.2009-0083169)

논문접수 : 2010년 1월 11일

심사완료 : 2010년 5월 18일

† 학생회원 : 중앙대학교 컴퓨터공학과
fanpanic@cglab.cau.ac.kr

** 학생회원 : 중앙대학교 영상공학과
shseo@cglab.cau.ac.kr

*** 비 회 원 : 한신대학교 컴퓨터공학과 교수
stryoo@hs.ac.kr

**** 종신회원 : 중앙대학교 컴퓨터공학과 교수
khyoon@cau.ac.kr

Copyright©2010 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 시스템 및 이론 제37권 제4호(2010.8)

1. 서론

일반적으로 인상과 화가들은 그리고자 하는 대상이 결정되면 대상의 움직이는 형태 및 패턴을 인지하여 2D 캔버스 안에 그린다. 하지만 그림이 그려지는 캔버스는 [1]에서 언급된 것과 같이 평면적(flat)이며 정적(static)인 제약을 가지고 있다. 이런 한계점을 극복하기 위해 작가는 구상 단계에서 구도와 다양한 대비 기법들을 사용하며 이와 더불어 그리고기 단계에서는 브러시의 방향을 변형하여 사용한다. 예를 들어 흐르는 개울을 그릴 때는 물이 흐르는 방향대로 브러시를 그리며 구름이 흐르는 것을 표현하기 위해 수평 방향의 브러시를 사용하기도 한다. 이와 같이 표현하고자 하는 대상의 움직임과 유사한 스트로크 방향은 2D 캔버스 안에서 객체의 움직임을 쉽게 알 느낄 수 있게 하여 2D 캔버스의 제약을 극복하는 하나의 도구였다. 또한 관찰자에게 시간의 흐름을 느낄 수 있게 하며 그림의 생동감을 주는 효과를 얻는다.

기존의 2D 기반의 회화적 렌더링 연구[2,3]의 스트로크 방향은 일반적으로 영상 기술기 정보에 기반해 결정되었다. 이 기법은 객체의 형태를 잘 표현하는데 효과적이기 때문에 정물 및 정적인 장면을 표현하는데 적합하였다. 하지만 풍경화와 같이 움직임이 많고 다양한 방향성이 존재하여 생동감 있는 결과를 생성하기에는 한계가 있었다.

따라서 본 논문에서는 입력된 장면의 자연물(물, 풀, 나무) 등이 가지는 움직임 정보에 기반해 이를 효과적으로 표현할 수 있는 회화적 렌더링 알고리즘을 제안한다. 우리는 동일한 시선(view)을 가진 순차 영상을 입력으로 사용하고 각 프레임 사이의 움직임정보를 추출한다. 우리는 프레임 사이의 움직임 정보들을 통합하여 표현될 장면이 가진 움직임의 크기, 방향 정보를 추출한다. 그리고 움직임의 크기 정보에 따라 움직임의 방향 정보가 사용될 영역을 결정하며 이에 기반하여 스트로크의 방향을 결정한다. 추가적으로 우리는 추출한 움직임정보에 기반해 대상의 움직임을 보다 효과적으로 표현할 수 있도록 스트로크의 색상 결정 기법도 제안한다.

본 연구에서 제안한 알고리즘은 다음과 같은 기여도를 갖는다. 첫째로 풍경화와 같이 움직임이 많은 그림을 그릴 때 객체의 실제적인 흐름을 표현 가능하며 이를 통해 보다 사실적이며 동적인(dynamic)결과를 생성할 수 있다. 두 번째로 스트로크 생성에 관한 새로운 접근이다. 우리는 움직임 정보에 기반해 스트로크의 방향 및 색상을 결정하였으며 추후 다른 스트로크 기반 연구(stroke based rendering)에도 확장 가능할 것으로 기대한다.

2. 관련연구

회화적 렌더링 연구가 처음 소개된 Haberli[4]에서는 사용자 입력을 통하여 스트로크의 방향을 결정하였다. 1990년대 후반 들어 본격적인 스트로크 생성에 관한 연구가 진행되었다. [5]은 비사실적 렌더링의 초기 연구 중 하나로 스트로크 텍스처를 3D 데이터상에 사용자 입력으로 주어진 방향을 고려해 자연스럽게 합성시키는 연구를 제안하였다. 본 연구와 같은 2D기반연구 중, Litwinowicz[2]와 Hertzmann[3]은 영상 기술기에 기반해 스트로크 방향을 결정하였다. 영상 기술기에 기반한 스트로크의 방향은 대상의 윤곽선을 따르게 되어 객체의 외형을 잘 표현할 수 있다. 하지만 주변의 스트로크 간에 일관된 방향성을 표현하지는 못하였다. 이를 보완하기 위해 Hertzmann[6]과 Hays[7]는 기술기 정보에 기반해 보간 법을 제안한다. 보간 법은 주변 기술기 값들을 일정한 방향으로 통일시켜 결과 영상에 일관된 방향성을 표현하였다. Maxwell[8]에서는 인상파 화가 '반 고흐'의 휘몰아치는 것과 같은 방향을 표현하기 위해 유체 시뮬레이션에 기반한 스트로크의 방향성을 표현하였다. 이 연구에서는 휘몰아 치는 방향성을 표현할 수는 있었으나 사용자의 입력이 필요하며 주변의 형태를 고려하지 않은 방향성이 표현되는 한계를 가졌다. [9]에서는 2D 영상에 색연필효과를 표현하는 연구를 제안하였다. 이 연구에서 색연필 스트로크의 방향은 물체의 형태(boundary)를 고려해 결정하였고 이는 기존의 그래디언트 방향을 사용한 것과 동일하였다. 기존의 연구들과 다르게 우리의 알고리즘은 순차 영상으로부터 얻은 움직임 정보를 기반으로 스트로크의 방향을 결정한다. 따라서 대상의 실제적인 흐름을 표현 가능하며 보간법을 사용하여 일관된 방향성이 표현되도록 하였다.

3. 객체의 움직임을 표현할 수 있는 스트로크 생성

그림 1은 우리의 시스템 개요를 보여준다. 렌더링하고자 하는 장면의 움직임 정보를 추출하기 위해 본 연구에서는 입력으로 고정된 시점을 갖는 연속된 프레임 [1~n]을 받는다. 그리고 렌더링을 위해 객체의 움직임 분석과 형태 분석으로 나누어 진행한다. 움직임 분석 부분에서는 입력된 순차 영상을 분석하여 움직임의 크기와 움직임의 방향정보를 구한다. 형태 분석에서는 순차 영상 중 맨 마지막 장(n'th frame)의 영상 기술기에 기반해 형태정보를 분석한다. 우리는 두 분석에 기반해서 객체의 움직임을 나타내는 스트로크 생성하고 이를 2D 캔버스에 적용한다. 각 파트에 대해서는 다음 장에서 자세히 설명하겠다.

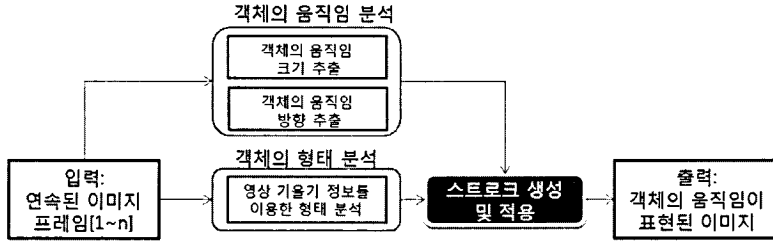


그림 1 시스템 개요

3.1 객체의 움직임 분석

입력으로 들어온 n장의 프레임으로부터 움직임 정보를 추출하기 위해 본 연구에서는 광류 흐름 법(optical flow)[10]을 이용한다. 광류흐름 법은 프레임 사이의 움직임 정보를 추출하는 대표적인 알고리즘으로 움직임 정보를 필요로 하는 많은 비사실적 렌더링 연구들에서 사용되어 왔다.

우리는 i'th 와 i+1'th 프레임 사이의 움직임을 \vec{m}_i 로 정의한다. \vec{m}_i 는 이차원 배열로써 두 프레임 사이에서 나타나는 각 화소(pixel)의 움직임의 이동 정보가 각도로 저장된다(0~360). 또한 \vec{m}_i 의 각 화소의 크기 값을 저장하고 있는 이차원 배열을 mag_i 로 정의한다. 입력된 프레임 들로부터 총 n-1개의 움직임 방향 과 크기 집합 (1)을 구할 수 있다.

$$\vec{m} = [\vec{m}_1, \vec{m}_2, \vec{m}_3, \dots, \vec{m}_{n-1}],$$

$$mag = [mag_1, mag_2, mag_3, \dots, mag_{n-1}] \quad (1)$$

각 프레임으로부터 구한 움직임의 움직임과 크기 집합으로부터 최종 영상 렌더링에 필요한 각 화소의 통합 방향, 크기를 구한다.

- 움직임의 방향(\vec{M})

각 화소의 움직임 방향은 \vec{m}_1 에서 \vec{m}_{n-1} 의 방향을 평균하여 얻는다. 이때 각 움직임의 크기 값이 일정 값 이상 되는 화소들만 더한다. 이때 $0^\circ = 360^\circ$ 이기 때문에 움직임의 평균 방향은 원형 자료구조 기반으로 계산한다. (2)는 j화소의 움직임 방향을 구하는 식을 보여준다. 각 화소는 [0~360]의 각도를 값으로 가지게 된다.

$$\vec{M}_j = avg\left(\sum_{k=1}^{n-1} \vec{m}_{k_j}\right) \quad (2)$$

- 움직임의 크기(Mag)

각 화소가 가지는 움직임의 크기는 각 프레임 사이의 움직임 크기로부터 평균을 구하고 정규화 시킨다. 우리는 움직임의 크기정보를 기준으로 하여 움직임의 방향이 사용될 영역을 결정한다. 식 (3)은 j화소의 움직임의 크기 값을 구하는 식이다.

$$Mag_i = N \left[\frac{1}{n-1} \left(\sum_{k=1}^{n-1} mag_{k_j} \right) \right] \quad (3)$$

여기서 $N_{0-1}[]$ 은 0부터 1의 정규화를 의미한다.

그림 2는 입력 프레임 집합 중 일부 프레임을 보여주고 있다. 사용된 영상에서 개울은 우측상단에서 좌측 하단으로 흐르고 있으며 다리에 사람이 건너는 움직임을 보여준다.

그림 3은 입력 프레임 집합으로부터 구한 움직임 정보를 보여주고 있다. 그림 3(a)는 객체의 움직임 정보를 0~255로 정규화하여 표현한 것으로 밝은 부분일수록 움직임이 큰 영역을 의미한다. 그림 3(b)와 그림 3(c)는 각각 움직임의 방향과 기울기 수직의 방향을 색상으로 표현한 것이다. 각각의 값은 0~360의 각도 값으로 표현가능하기 때문에 HSV 칼라모델의 H에 각도 값을 넣어 표현한 것이다. (각도가 0인 부분은 hue=0, saturation=1,value=1)을 가져 빨간색으로 표시된다.) 그림 2에서 개울과 다리에서 움직임을 가지고 있기 때문에 해당 영역에서만 움직임이 추출되고 그 이외의 영역에서는 객체의 움직임이 추출되지 않음을 알 수 있다. 개울의 흐름이 영상에서 좌측 하단으로 흐르고 있기 때문에 움직임의 방향도 약 230도 정도의 방향을 가지게 되며 그에 따른 HSV의 푸른색으로 표현된 것을 볼 수 있다. 영상의 다리 부분에서 나타나는 움직임의 경우 푸른색 방향이 나타나고 기울기 영상에서는 붉은색으로 표현되는 것은, 사람이 오른쪽에서 왼쪽으로 이동하여 움직임의 방향이 좌측을 향하고 있는 반면에 기울기 정보에서는 윤곽선에 따라 오른쪽으로 방향정보가 추출되었기 때문이다.

3.2 객체의 형태 분석

객체의 움직임이 없는 부분의 스트로크는 기존의 연구에서 사용되어 오던 영상 기울기의 수직방향에 해당하는 방향정보를 사용한다. 우리는 n번째 프레임을 분석하여 영상 기울기의 수직 방향을 추출한다. 이것은 (\vec{G})로 정의되며 각 화소의 값은 (\vec{M})와 마찬가지로 0~360의 값을 갖는다. 그림 3(c)는 영상 기울기 정보를 이용한 방향 정보를 보여주고 있다.

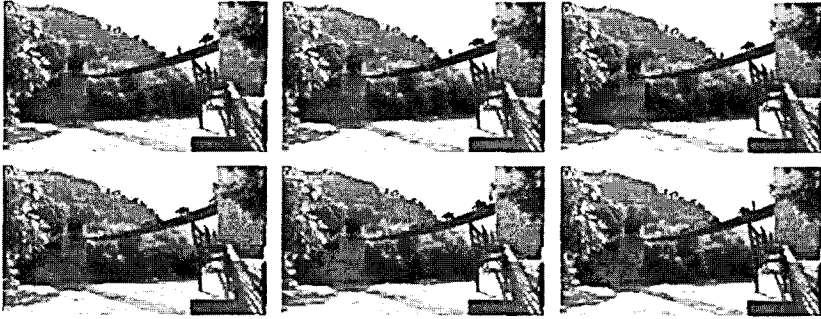


그림 2 입력 프레임(1,10,20,30,40,50,60번 프레임)

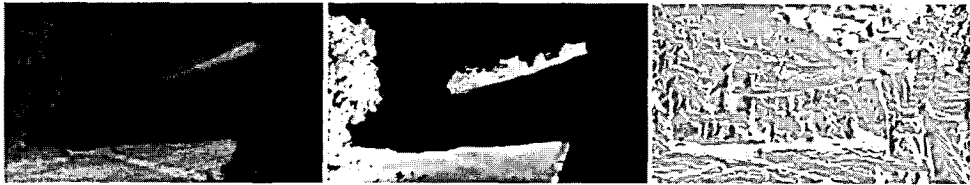


그림 3 객체의 움직임 크기 정보(a) 및 방향 정보(b), 영상 기울기 수직 방향(c)

3.3 움직임과 형태 정보에 기반한 스트로크 생성 및 렌더링

우리는 연속된 프레임으로부터 얻은 객체의 움직임 정보에 기반해 스트로크의 방향을 결정한다. 이때 움직임이 없는 정지된 객체에 대해서는 기존의 연구들에서 사용된 영상 기울기를 사용하여 객체의 형태를 유지할 수 있도록 한다.

i화소에서 스트로크 방향은 식 (4),(5)에 기반해 결정된다. 우리는 i화소와 그 주변의 화소들에 가우시안 가중치를 적용하여 얻어진 움직임의 평균값을 기반으로 결정한다. 이것은 주변의 화소 정보들을 고려하여 화소들 사이의 감박거림을 줄이기 위해서이다. 움직임의 크기 값이 있는 경우는 움직임의 방향성을 사용하며 작은 경우는 영상 기울기의 값을 사용한다.

$$\vec{D}_i = \alpha \vec{M}_i + (1 - \alpha) \vec{G}_i \tag{4}$$

$$\alpha = \begin{cases} 1 & \text{if } Mag_i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

우리는 각 화소의 움직임 크기 값에 따라 움직임의 방향과 기울기 기반의 방향이 쓰여지는 것을 결정한 후 초기값에 기반하여 에지 탄젠트[11] 보간법을 거친다. 보간을 통해 움직임과 기울기 방향 사이의 충돌을 막을 수 있으며 일관된 흐름을 가지는 방향성을 생성 할 수 있다. 그림 4는 객체의 움직임 방향성과 기울기가 사용되는 범위 및 통합된 방향성을 보여주고 있다. 그림 4의 우측 영상은 움직임 벡터를 선 적분 회전기법(LIC)[12]를 적용하여 벡터의 방향성을 보여주고 있다. 추가적으로 우리는 각 움직임이 가지는 표준편차 정보를 활용하여 다양한 움직임을 보이는 객체의 형태 표현에 대하여도 결론부분에서 아이디어를 제안한다.

회화적 렌더링을 위해서는 앞에서 결정한 스트로크의



그림 4 움직임 정보와 기울기의 통합 방향성 파란 영역은 움직임의 방향성을 사용, 붉은 영역은 영상 기울기를 사용한 영역

방향 이외에도 다른 스트로크 속성들을 설정하고 캔버스에 적용해야 한다. 우리는 Hays[6]의 렌더링 테크닉을 기본 알고리즘으로 사용한다. Hays의 렌더링은 예지 기반으로 스트로크의 속성들을 결정한다.

기존의 회화적 렌더링에서 스트로크의 색상은 입력 영상으로부터 스트로크가 적용될 위치의 화소 값을 사용한다. 본 연구에서는 움직임의 움직임을 시각적으로 쉽게 인지할 수 있도록 누적된 색상 값을 사용 가능하도록 추가적인 옵션을 제공한다. 스트로크가 그려질 위치 j 에서의 색상은 다음 식 (6)에 의해서 결정된다. 우리는 RGB 색상모델을 사용했으며, 각 채널에 대해 프레임별 가산 평균 색상을 사용하였다.

$$Color_j = w_c \cdot imgC(n_j) + (1 - w_c) \left(\frac{1}{n-1} \sum_{k=1}^{n-1} imgC(k_j) \right) \quad (6)$$

여기서 w_c 는 가중치 값을 의미하며 1을 사용한 경우 n_j 프레임의 색상만을 사용한다. $imgC(K_j)$ 는 k 프레임의 j 화소가 가지는 색상 값을 나타낸다.

5. 실험 결과

본 장에서는 현재까지 진행된 연구의 결과를 기존연구와 비교한다. 또한 향후 계획하고 있는 몇 가지 아이디어에 대한 테스트 정보를 보이며 활용방향을 설명하

고자 한다. 최종 결과 영상을 얻는데 걸리는 시간은 영상의 크기 및 입력된 프레임 수에 따라 의존적이지만 일반적으로 900×500 영상 70프레임 정도를 사용하였을 경우 약 30초 정도의 시간이 소요된다. 그림 5 기술기 값만을 사용한 기존의 결과와 움직임의 움직임 정보를 고려한 본 연구의 결과를 보여준다. 기술기 값에 기반한 렌더링 결과 영상에서는 개울의 흐름을 추출할 수 없으므로 개울이 흐르는 모습을 표현할 수가 없다. 또한 개울에 비친 다리의 그림자가 영상 기술기 값에 영향을 주어 그림자의 방향에 따라 브러시가 그려지게 된다. 하지만 본 연구에서 제안한 움직임의 정보를 고려하면 개울이 흐르는 방향에 따라 스트로크의 방향이 결정되어 기존의 결과 영상에 비해 사실적인 개울의 흐름을 표현할 수 있다. 하지만 이때 다리 위에 있는 객체가 가진 움직임 정보에 의해 원 영상에서 나타난 객체의 형태가 훼손된 결과를 얻게 되었다.

그림 6은 움직임 정보에 기반해 스트로크에 사용되는 색상을 조절한 결과를 보여준다. 0에 가까운 w_c 값을 사용할수록 프레임의 색상을 누적하여 사용하여 움직임 불러와 같은 효과를 표현할 수 있다. 이는 기차와 같이 색의 차이가 명확하며 형태가 고정된 객체를 표현할 때 유용하게 사용할 수 있다. 하지만 구름, 물과 같이 색상이 비슷한 장면에서는 일반적으로 $w_c = 1$ 을 사용한다.

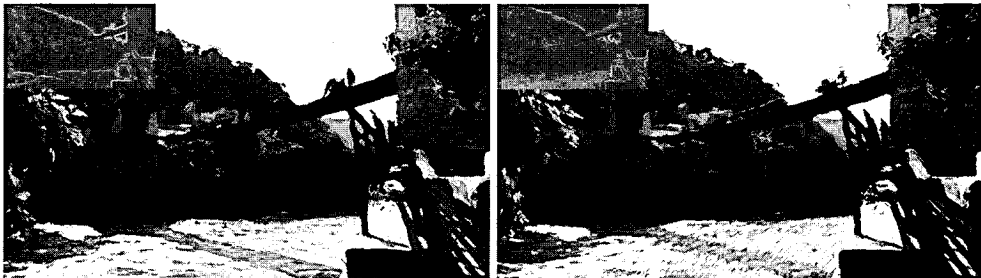


그림 5 기술기만을 사용한 결과(좌), 움직임을 고려한 결과(우) 비교

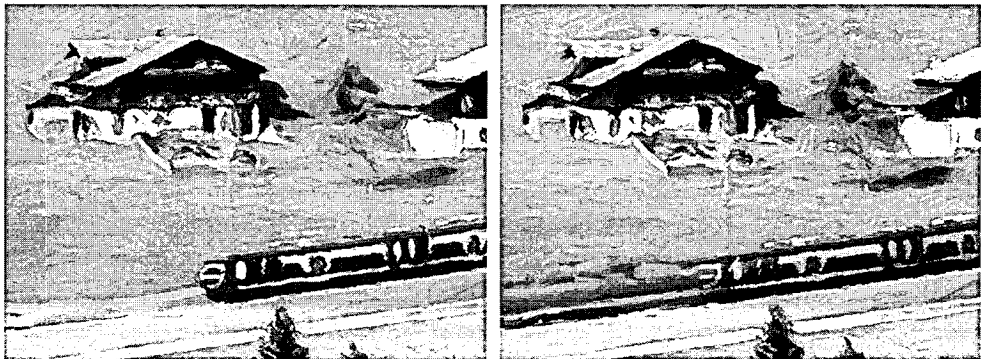


그림 6 누적 색상 값 적용에 따른 결과 (좌: $w_c = 0$ 우: $w_c = 1$)

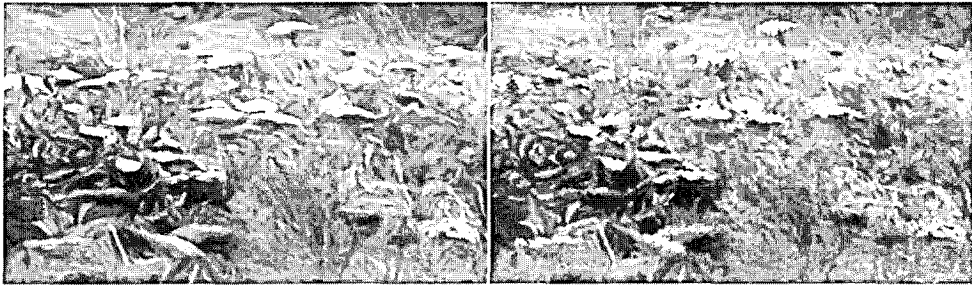


그림 7 표준편차를 고려 유(좌), 무(우) 결과 비교

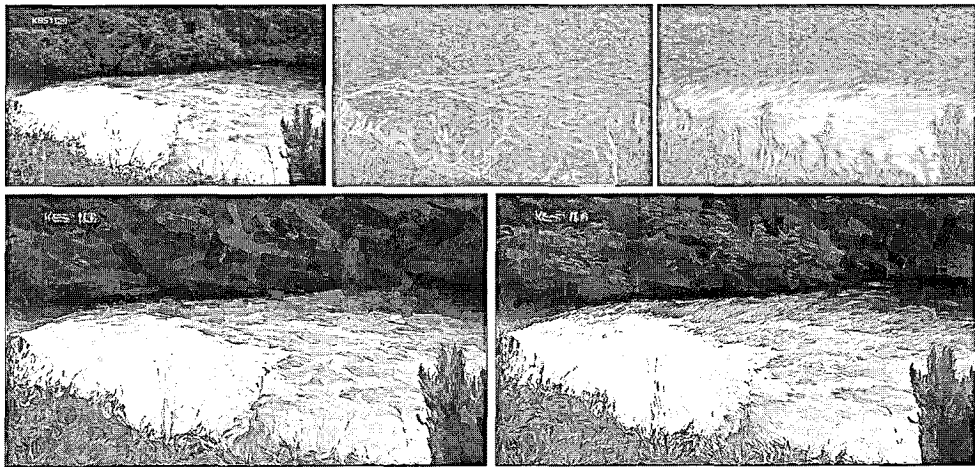


그림 8 다른 영상에 적용해본 결과

그림 7은 움직임의 크기뿐 아니라 표준편차까지 고려한 경우의 결과를 보여준다. 본 결과를 표현하기 위한 동영상은 꽃이 좌우로 휘날리고 있다. 우리는 표준편차에 기반해 기존의 스트로크 방향에 난수 값을 적용하였다. 표준편차를 고려함으로써 꽃의 형태는 유지가 되지 않았지만, 보다 더 동적인 느낌을 표현 가능하다. 그림 8은 다른 영상에 적용한 결과를 보여준다.

6. 결론 및 개선 아이디어 토의

본 논문에서는 객체의 움직임 정보를 표현하는 회화적 렌더링 알고리즘을 제안하였다. 우리는 고정된 화면의 순차 영상으로부터 움직임의 정보들을 추출하였고 추출된 정보에 기반해 움직임의 크기, 방향 정보를 구하였다. 그리고 움직임의 정보에 기반한 스트로크의 방향 및 색상을 결정하였다. 우리의 알고리즘은 풍경영상을 렌더링 할 때 장점을 가지며 특히 하늘이나 폭포와 같이 움직임이 명확한 영상을 렌더링하는데 효과적이다. 하지만 고정된 시점의 다중 프레임을 사용해야 한다. 또한 정물과 같이 움직임이 적은 장면을 렌더링 하기에는

한계가 있으며, 사람과 같은 형태를 가진 객체에 있어서는 형태를 훼손시키는 한계를 가지고 있다. 우리의 알고리즘은 다음의 기여도를 갖는다. 움직임 정보를 사용한 것은 스트로크 기반 렌더링(SBR) 분야에서 새로운 접근방법이며 추후 회화적 렌더링이 아닌 다른 분야에 확장 가능할 것으로 기대한다.

본 연구는 현재 진행 중이며 향후 개선 및 구현할 아이디어를 계획 중이다. 우리는 움직임의 크기와 방향 값 이외에 움직임의 표준편차 정보를 활용한 알고리즘으로 확장을 계획 중이다. 우리는 각 객체를 [움직임의 크기, 움직임의 편차] 정보를 활용해 그룹화 시키고 각 영역에 특성에 맞는 스트로크를 적용해볼 계획이다. 본 연구에서 표현한 개울과 같은 곳은 [움직임이 큼, 편차가 적음]으로 나타내어질 수 있으며, 꽃과 같은 부분은 [움직임이 큼, 편차 큼]으로 구분될 것이다. 또한 산과 같은 영역은 [움직임 없음, 편차 없음] 등으로 구분되어 각 영역을 보다 명확히 구분하고 각 영역의 특성에 맞는 스트로크의 요소(방향, 질감, 색상)를 적용함으로써 보다 향상된 결과를 얻을 수 있을 것으로 본다. 또한 단순히

각 프레임의 움직임정보를 통합하여 사용하는 것이 아닌, 움직임의 흐름을 추적하여, 시간적 일관성을 유지시킬 수 있는 정보를 추출하고 이를 적용하는 연구가 필요하다.

참 고 문 헌

[1] F Durand, "Perceptual and Artistic Principles for Effective Computer Depiction," *ACM SIGGRAPH 2002 Course Notes*, 2002.

[2] P. Litwinowicz, "Processing Images and Video for an Impressionist Effect," *Proc. Siggraph 97*, ACM Press, pp.407-414, 1994.

[3] A. Hertzmann, "Painterly Rendering with Curved Brush Strokes of Multiple Sizes," *Proc. Siggraph 98*, ACM Press, pp.453-460, 1998.

[4] P. Haeblerli, "Paint by Numbers: Abstract Image Representation," *Computer Graphics (Proc. Siggraph)*, vol.24, no.4, pp.207-214, 1990.

[5] Salisbury, M., Wong, M., Hughes, J., and Salesin, D., "Orientable Textures for Image-based Pen-and-Ink Illustration," In *Proceedings of SIGGRAPH 1997*, pp.401-406, 1997.

[6] A. Hertzmann, "Fast Paint Texture," *Proc. 2nd Ann. Symp. Non-Photorealistic Animation and Rendering (NPAR 2002)*, ACM Press, pp.91-96, 161, 2002.

[7] J. Hays and I. Essa, "Image and Video Based Painterly Animation," *3rd Ann. Symp. Non-Photorealistic Animation and Rendering(NPAR2004)*, pp.120-133, 2004.

[8] S. C. Olsen and B. A. Maxwell, "Fluid Simulation as a Tool for Painterly Rendering," *SIGGRAPH Poster*, 2004.

[9] Matsui, H., Johan, H., and Nishita, T., "Creating Colored Pencil Images by Drawing Strokes based on Boundaries of Regions," In *Proceedings of Computer Graphics International 2005*, pp.148-155, 2005.

[10] M. Black and P. Anandan, "Robust dynamic motion estimation over time," *Proc. computer Vision and Pattern Recognition*, pp.296-302, 1991.

[11] H. Kang, S. Lee, C. Chui. "Coherent Line Drawing," *Proc. ACM Symposium on Non-photorealistic Animation and Rendering*, pp.43-50, 2007.

[12] B. Cabral and L. Leedon, "Imaging vector fields using line integral convolution," *Proc. of SIGGRAPH '93, ACM SIGGRAPH*, pp.263-272, 1993.



이 호 창

2006년 중앙대학교 공과대학 컴퓨터공학과 졸업. 2008년 중앙대학교 일반대학원 컴퓨터공학과 석사졸업. 2008년~현재 중앙대학교 컴퓨터공학과 박사과정. 관심분야는 컴퓨터 그래픽스, 비사실적 렌더링, Image stylization, GPGPU



서 상 현

1998년 중앙대학교 공과대학 컴퓨터공학과 졸업. 2000년 중앙대학교 첨단영상대학원 영상공학과 석사졸업. 2003년~2005년 (주)지노시스템 기술연구소 선임연구원. 2009년 중앙대학교 첨단영상대학원 영상공학과 박사. 관심분야는 컴퓨터 그래픽스, 비사실적 렌더링, Proceduralism, GIS



류 승 택

1996년 중앙대학교 공과대학 전자계산학과 졸업. 1998년 중앙대학교 대학원 컴퓨터공학과 석사. 2002년 중앙대학교 영상공학과 공학 박사. 2004년~현재 한신대학교 컴퓨터공학부 부교수. 관심분야는 영상기반 렌더링, 비사실적 렌더링, 실시간 렌더링

간 렌더링



윤 경 현

1980년 중앙대학교 공과대학 전자계산학과 졸업. 1983년 중앙대학교 대학원 전자계산학과 석사. 1983년~1985년 한국전기연구소 연구원. 1988년 University of Connecticut 전자계산학과 석사. 1991년 University of Connecticut 전자계산학과 박사. 1991년~현재 중앙대학교 컴퓨터공학부 교수. 관심분야는 컴퓨터 그래픽스, GIS, 영상기반모델링 및 렌더링, 비사실적 렌더링

비사실적 렌더링