

페인팅 매체 프로파일을 이용한 스트로크 기반 렌더링에 관한 연구

서상현[†], 윤경현^{**}

요 약

본 논문에서는 브러시 스트로크 프로파일을 이용한 새로운 스트로크 기반 표현기법을 소개한다. 우리가 제안하는 알고리즘은 이미지 검색기법에 기반하고 있으며 단순하지만 다양한 페인팅 스타일을 생성할 수 있는 유연성과 확장성이 있는 알고리즘이다. 이를 위해 우리는 실제 스트로크 데이터를 수집하여 구축한 확장 가능한 데이터베이스를 사용한다. 그리고 입력영상을 데이터베이스에 있는 스트로크의 조합으로 재생산 한다. 이때 적합한 스트로크를 결정하는 검색 프로세스와 검색된 스트로크를 캔버스에 그릴지 말지를 판단하는 판단 프로세스를 알고리즘에 포함시킨다. 또한 우리는 물리적인 시뮬레이션 없이 사용할 수 있는 시각적 효과가 좋은 효율적인 브러시 스트로크 모델 및 표현기술을 제안한다. 제안된 알고리즘은 다양한 사용자 파라미터를 제어함으로써 다양한 회화적 작품의 변이를 만들어 낼 수 있고 스트로크 데이터베이스의 구성을 변경함으로써 다양한 회화적 매체의 스타일을 갖는 결과물을 생산할 수 있는 확장성 있는 프레임워크를 제공한다.

A Study on Stroke Based Rendering Using Painting Media Profile

Sang Hyun Seo[†], Kyung Hyun Yoon^{**}

ABSTRACT

In this paper we introduce a new approach to stroke based rendering using brush stroke profile. Our proposed method, based on image retrieval method, is a simple but flexible and scalable method to create various painting styles, for which scalable database constructed with the collection of real stroke data is used. Input image is reproduced with combinations of brush stroke in the database, when a search process to determinate appropriate brush stroke and a judgment process to decide whether to draw the retrieved brush stroke on the canvas or not are presented. In addition, this paper suggests a new brush stroke model and a depiction technique in order to utilize effective height information which allows natural texture depiction, or good visual effect, without carrying out physical simulation. Our method is able to create diverse variations of painting by controlling various user parameters. It also provides scalable framework that can produce various painting styles with different artistic media by changing the stroke combinations of stroke database.

Key words: Non-Photorealistic Rendering(비사실적 렌더링), Painterly Rendering(회화적 렌더링), Brush Stroke(브러시 스트로크), Database(데이터베이스)

※ 교신저자(Corresponding Author): 윤경현, 주소: 서울 특별시 동작구 흑석동 221(156-756), 전화: 02)824-3018, FAX: 02)824-3018, E-mail: khyoon@cau.ac.kr
접수일: 2009년 6월 17일, 수정일: 2009년 8월 1일
완료일: 2009년 8월 28일

[†] 정회원, 중앙대학교 첨단영상대학원 영상공학과 박사과정

(E-mail: shseo@cglab.cau.ac.kr)

^{**} 정회원, 중앙대학교 컴퓨터공학부 교수

※ 본 논문은 2007년도 중앙대학교 우수연구자 연구비지원에 의한 연구임

※ 본 논문은 교육과학기술부의 재원으로 국가연구재단의 지원을 받아 수행된 연구임(No. 2009-0083169)

1. 서 론

비사실적 렌더링(NPR, Non-Photorealistic Rendering)은 사람이 만들어낸 듯 한 예술적인 영상들을 컴퓨터로 재현해 내는 것이 목적이고 컴퓨터그래픽스 분야에서 주요한 연구 분야중 하나이다[1]. 비사실적 렌더링은 펜과 잉크 일러스트레이션[2], 수채화[3], 만화[4], 목탄[5], 연필[6] 등 그 연구의 적용 분야가 다양하다.

본 논문은 기존 렌더링 시스템들에 비해 더욱 다양한 형태의 회화 매체를 갖는 결과물 생성이 가능한 시스템을 개발하는 것을 목표로 하고 있다.

스트로크 기반 렌더링(Stroke Based Rendering)에 대한 연구는 표현 방법에 따라 사용자와의 상호작용에 의한 물리적 시뮬레이션 방법, 3차원 오브젝트에 직접 스트로크를 매핑하는 오브젝트 기반 방법, 2차원 영상으로부터 스트로크를 생성하는 이미지 기반 스트로크 기반 렌더링 방법, 그리고 비디오 동영상으로부터 회화적 애니메이션을 생성하는 방법 등으로 구분할 수 있다[1]. 특히 이미지를 이용한 스트로크 기반 렌더링 알고리즘은 영상처리 기술을 통해 입력영상을 분석하고 분석된 데이터에 기반하여 방향, 위치, 색상, 길이 등 서로 다른 속성을 가진 다양한 브러시 스트로크를 생성한다.

스트로크 기반 렌더링에서는 소스 영상을 효과적으로 표현하기 위해 브러시 스트로크의 다양한 속성들을 얼마나 정확하게 자동적으로 추출 혹은 생성할 수 있는지가 가장 중요한 요소이다. 반면 본 논문에서는 기존의 이미지 기반 회화적 렌더링 방법에서 추구하던 페러다임(스트로크의 자동 검출/생성 방법)과 다른 새로운 방식을 소개한다. 우리는 회화적 영상이 무한한 브러시 스트로크의 집합으로부터 생성된다고 가정한다. 하지만, 무한한 스트로크의 집합을 생성해 내는 것은 거의 불가능 일이다. 따라서 우리는 각 미디어를 표현하기 위해 수집된 사용자 브러시 스트로크 집합들을 구성하고 이렇게 미리 정의된 스트로크 집합들을 효율적으로 생성, 검색할 수 있는 스트로크 데이터베이스를 사용한다. 데이터베이스에 저장된 유한의 브러시 스트로크들을 효과적으로 조합함으로써 다양한 형태의 회화적 결과물을 생성할 수 있다. 이러한 아이디어는 포토모자이크(Photomosaic) 알고리즘[7]의 아이디어와 유사한 측

면이 있다. 포토모자이크는 입력 영상을 서로 다른 수많은 사진이나 그림들의 조합으로 만들고 멀리서 보았을 때 마치 새로운 한 장의 모자이크 그림처럼 보이게 하는 기술이다. 그러나 포토모자이크에서 조합의 기본 단위가 자체로서 하나의 작품이 될 수 있는 일반적 사진이나 그림이었던 반면, 제한하는 방식에서는 실제 회화 작품의 구성 과정에 사용되는 기반 사용자 스트로크를 조합의 기본 단위로 한다는 점이 다르다.

지금까지의 다른 연구자들이 그려왔던 것처럼 우리의 접근 방식 또한 고유의 브러시 스트로크를 사용하여 피사체를 표현하고자 하는 일반적인 화가의 페인팅 과정을 따른다. 일반인은 스트로크라는 매개체를 이용해 피사체에 대한 해석의 과정을 2차원 캔버스 위에 점진적으로 표현한다. 다시 말해 피사체(입력영상 or 실세계)를 그리기위해 캔버스의 임의의 위치에 하나의 브러시 스트로크를 그릴 때마다, 해당 피사체와 가장 유사하다고 판단되는 브러시 스트로크를 캔버스 위에 칠하게 된다. 이러한 과정을 큰 브러시부터 작은 브러시로 반복함으로써 최종적인 회화가 완성된다.

본 논문에서는 스트로크 데이터를 수집한 스트로크 프로파일에 기반하여 데이터베이스를 구축하고, 데이터베이스에 포함된 다양한 브러시 스트로크를 조합하여 입력 영상을 회화작품으로 재생산한다. 우리는 스트로크를 조합하는 과정에서 질감표현을 위한 스트로크 모델을 소개함으로써 브러시의 물리적인 계산 없이 최종 결과의 질감 표현력을 증대시킨다. 또한 적합한 스트로크를 선택하는 검색 알고리즘과 찾아진 스트로크를 캔버스에 그릴지 말지를 판단하는 판단 알고리즘을 제시함으로써 추상화 단계나 결과의 정확도를 제어할 수 있는 구조를 갖는다.

본 연구에서 주로 실험한 결과물은 유화 스타일 렌더링이다. 하지만 이후 제안된 알고리즘이 브러시 스트로크 집합과 몇 개의 렌더링 파라미터에 대한 변형을 통해 다양한 페인팅 스타일을 표현할 수 있는 확장성 있는 프레임워크임을 추가적인 실험 결과와 예시를 통해 보여줄 것이다.

본 논문의 구성은 다음과 같다. 2절에서는 본 연구 관련된 기존 연구에 대해 서술한다. 3절에서는 우리의 렌더링 방식의 전체적인 개념과 사용된 알고리즘들을 설명한다. 4절과 5절에서는 제안된 방식에 의해

생성된 다양한 결과를 예시하고 이에 대한 분석 및 향후 응용 방안에 대하여 기술한다.

2. 관련연구

스트로크 기반 렌더링은 입력데이터의 종류(2D, 3D)에 따라서 다양한 접근 방법들이 있으며 우리의 연구는 2D 이미지기반 렌더링에 포함된다. 이미지기반 렌더링에 관한 연구는 Haerberli[8]에 의하여 처음 시도되었다. Haerberli는 픽셀이 아닌 브러시 스트로크를 렌더링 기본단위로 하여 결과영상을 렌더링하였다. 또한 브러시 스트로크의 시작 위치, 색, 크기, 방향, 모양과 같은 속성을 제안하였으며, 이들 속성을 달리하여 다양한 스타일의 브러시 스트로크를 이용하여 렌더링하였다.

이후 Litwinowicz[9]는 질감을 입힌 직선 브러시 스트로크를 입력 영상의 기울기 방향을 따라 배치하여 결과 영상에서 회화적 느낌을 표현하였다. 이 논문의 가장 두드러지는 특징은 브러시 스트로크의 방향을 결정하는데 있어서 그 이전 연구들이 사용자 입력에 의존했던 것과 달리 입력 영상으로부터 입력 영상의 기울기를 이용하여 자동으로 결정하였다는 점이다. 이후 자동화된 스트로크 생성방법에 대한 연구는 계속되었다.

Hertzmann[10]은 실제 회화 작품에서 브러시 스트로크의 크기가 일정하지 않고 모양이 곡선인 것에 주목하여 다양한 크기를 가진 곡선 브러시 스트로크를 제안하였다. 이 연구에서는 브러시의 크기에 따라 레이어를 나누고 각 레이어마다 브러시의 크기를 할당하여 곡선 스트로크를 그려 넣고 이를 적층하여 결과 영상을 완성하였다. 이 연구는 브러시 스트로크의 모양을 기존의 직선 형태에서 곡선 형태로 확장한 점에 큰 의의가 있다. 또한 다양한 크기의 브러시를 사용하여 서로 크기가 다른 브러시 스트로크를 만들었기 때문에 입력 영상에서 배경과 같이 넓은 부분은 굵은 브러시 스트로크로 표현이 되고 세세한 부분은 가는 브러시 스트로크로 표현할 수 있다는 장점을 가지고 있다.

이후, 그는 이전 알고리즘을 확장하여 회화적 애니메이션에 적용 가능한 광류(Optical Flow)기반 회화적 애니메이션 알고리즘[11]을 소개하였으며 하이트 필드(Height Field) 데이터를 활용한 브러시 스트

로크의 물리적인 효과를 추가시키는 기술도 제안하였다[12]. 또한 관련된 회화적 애니메이션 기술로 Hays 와 Essa[13]는 회화적 애니메이션을 위한 브러시 스트로크의 시간 간섭성(Temporal Coherence)유지 기술을 제시하였다.

그 외에 Nehab[14]는 이미지 모우먼트(Image-Moment)를 이용해 같은 칼라정보를 갖는 지역적 영역을 브러시 스트로크로 근사화하는 방법을 소개했으며 Shiraiishi[15]는 소스영상의 동질성 영역을 근사화하는 사각형 브러시 스트로크를 생성하는 방법을 소개하였다. Gooch[16]또한 브러시 스트로크를 생성하는데 있어서 입력 영상의 기울기에 의존하지 않고, 입력 영상에서 하나의 스트로크로 표현될 수 있는 영역을 추출하여 이를 브러시 스트로크로 만들었다. Gooch의 연구는 브러시 스트로크의 방향성을 입력 영상의 기울기에 의존하지 않았다는 점에서 기존 연구와 차별성을 갖는다.

최근에, Shugrina와[17]는 컴퓨터 비전의 인식기술을 이용하여 관찰자의 감정 상태를 평가하고 그 상태에 따라 회화적 효과를 달리하는 표현하는 스트로크 기반 회화적 렌더링 알고리즘을 소개하여 회화적 렌더링 알고리즘의 응용범위를 확장하였다. 또한 Baxter[18]는 실질적인 브러시를 모델링하여 물리적인 시뮬레이션을 통해 인터랙티브한 페인팅 소프트웨어를 소개하였다.

3. 알고리즘의 개념 : 스트로크의 조합에 의한 대상 표현

우리가 표현하고자 하는 것은 대상(Object)과 캔버스(Canvas) 사이의 불일치성을 이겨내고자 하는 예술가의 손재주와 노력이다.

예술가의 능력에 따라 표현 가능한 스트로크 종류나 정확성은 차이가 난다. 하지만 이러한 한정된 스트로크의 사용이 무한정으로 많은 스트로크의 사용보다 더 자연스럽게 예술적인 결과를 도출해 낸다. 이러한 방법은 인간이 그림을 그리는 과정과도 매우 유사하다. 제한된 스트로크를 가지고 캔버스와 대상 사이의 차이를 줄이는 방법은 다른 스트로크의 반복적인 시도를 통해서 가능하다.

그림 1은 대상 그림 1-a를 몇 개의 점진적인 스트로크의 조합을 통해 페인팅하는 과정을 보여주고 있

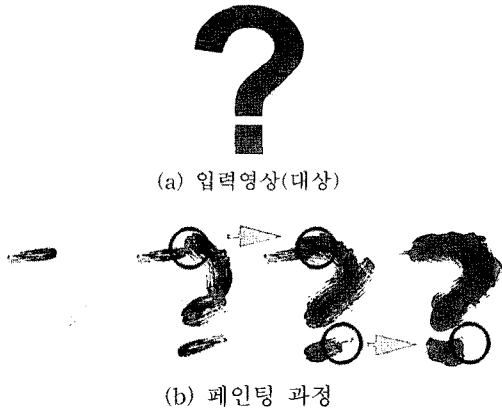


그림 1. 스트로크의 조합을 통한 Painting 과정

다. 그림 1은 다른 색 또는 모양을 갖는 스트로크를 덧칠함으로써 대상과의 차이가 나는 부분을 원본과 가까워지도록 진화하는 과정을 보여준다. 그림 1-b의 위쪽의 원에서는 붉은색 브러시 스트로크를, 아래쪽 원에서는 흰색 브러시 스트로크를 추가함으로써 원본과의 차이를 줄여 나간다. 보다 많은 시도 또는 보다 작은 스트로크의 덧칠이 이루어진다면 점점 원본과 가까워 질 것이다. 하지만 회화란 사물의 이미지를 캔버스에 옮기는 과정에서 반드시 브러시 스트로크로 인한 원본의 왜곡(변환)이 동반된다. 그러나 이러한 왜곡이 회화의 예술적 아름다움의 원인이 되기 때문에 이를 적절히 제어할 필요가 있다.

본 논문에서는 그림 1에 표현된 페인팅 과정을 시뮬레이션 하기위해 컴퓨터가 선택(표현)할 수 있는 스트로크를 제한하고, 이 제한된 스트로크들의 조합을 통해 최적 또는 차선의 결과가 나오도록 하는 페인팅 알고리즘을 제안한다. 우리는 제한된 스트로크들의 집합을 스트로크 데이터베이스(Stroke Database)라고 정의한다.

이는 이전의 회화적 렌더링 알고리즘들과는 근본적인 차이점을 가지고 있다. 이전 알고리즘들은 대상과 캔버스의 차이를 줄일 수 있도록 스트로크를 자동적으로 생성하는 방법에 초점을 맞추고 있는 반면, 우리의 알고리즘은 같은 목표를 위해 컴퓨터 또는 사용자로 하여금 제한된 스트로크에서 적절한 하나를 선택할 수 있도록 하였다.

제안하는 알고리즘은 구축된 스트로크 데이터베이스를 기반으로 3단계로 구성되어 있다. 첫 번째는 먼저 브러시 스트로크를 그릴 위치와 브러시의 크기

를 결정하고 해당 위치에 적합한 스트로크를 데이터베이스로부터 찾는 단계이다. 두 번째는 구축된 브러시 스트로크 데이터베이스로부터 해당 위치에 적합한 스트로크를 검색하는 단계이다. 세 번째는 찾아진 스트로크를 제안된 스트로크 표현기법에 의해 캔버스에 그려지는 단계이다. 이때 그려진 스트로크는 제안된 판단기준에 의해 원본 영상을 훼손하는지 보완하는지 판단하게 되고 훼손이 이루어질 경우 그린 것을 취소하게 된다. 이와 같은 과정은 점진적으로 이루어져 최종 결과를 생산하게 된다.

그림 2는 본 논문에서 제안하는 알고리즘의 각 프로세스를 보여주고 있다. 각 프로세스의 자세한 내용은 다음 절에서 자세히 설명하도록 하겠다.

3.1 스트로크 프로파일과 데이터베이스 구축

3.1.1 스트로크 프로파일

본 논문에서는 몇 개의 스트로크 프로파일(Stroke Profile)을 이용해 데이터베이스를 구축한다. 스트로크 프로파일이란 사용자가 실제로 그리거나 그래픽 소프트웨어를 이용해 만들어낸 몇 개의 스트로크 샘플들이다. 이는 실제 스트로크를 그려져 얻을 수도 있지만 포토샵과 같은 상용 그래픽 소프트웨어를 이용해 만들어진 스트로크 샘플을 활용 할 수도 있다.

본 연구에서 실험을 위해 주로 사용된 유화 프로파일의 경우는 후면에서 빛을 쏜 반투명 유리판위에 검은색 유화 안료를 실제로 그리고 캡쳐하여 얻었다. 그림 3은 실제 유화 프로파일을 얻는 과정을 보여주고 있다. 그려진 오일 안료의 두께는 밝기정보를 통해 얻을 수 있으며 여러 번의 시도를 통해 다양한 모양의 실제 스트로크를 얻을 수 있다. 캡쳐시 발생하는 노이즈 등을 제거하기위해 약간의 후처리 작업이 필요하다.

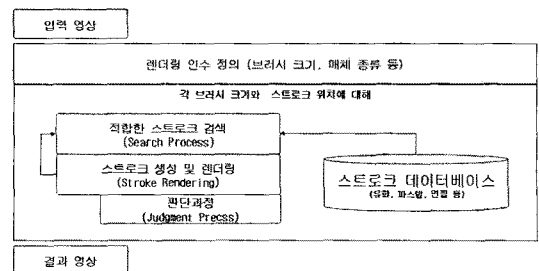


그림 2. 알고리즘 흐름도



(a) 실제 브러시 작업



(b) 얻어진 흑백 프로파일

그림 3. 유화 프로파일 생성 과정 및 결과

본 연구의 결론에서 사용한 다른 종류의 스트로크 프로파일의 경우는 상용 SW인 포토샵을 이용하여 제작하였다. 이렇게 얻어진 몇 개의 프로파일은 데이터베이스의 구축과정에 사용된다.

3.1.2 스트로크 데이터베이스

우리는 구축된 프로파일을 이용해 데이터베이스를 구축한다. 실제 얻어진 프로파일은 그 양이 많지 않으며 실제로도 많은 양의 데이터를 생성하기 힘들다. 그러나 우리는 다양한 브러시 모양과 방향을 갖는 브러시 스트로크를 제공하기 위해서 많은 양의 스트로크 집합이 필요하다.

이를 위해 우리는 프로파일의 변환된 복사본을 만들어 데이터베이스에 추가한다. 그림 4-a는 회전, 크기변환, 미러 등의 변환을 사용하여 스트로크 데이터베이스를 풍부하게 만든 과정을 보여준다. 회전변환은 다양한 방향을 표현할 수 있는 가능성을 높여주며, 크기변환은 작은 사이즈부터 큰 사이즈의 다양한 크기의 변이를 만들어 준다. 다양한 스트로크들을 담은 데이터베이스에는 검색을 위한 인덱스 이미지(Index Image)와 엠보싱(양각) 이미지도 함께 저장

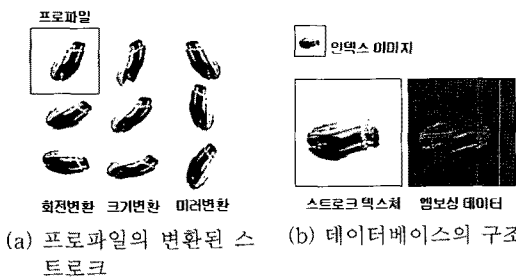


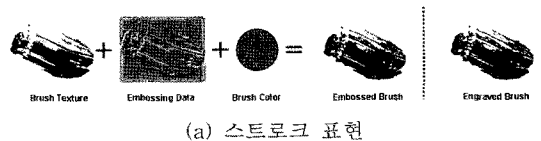
그림 4. 스트로크 프로파일을 이용한 데이터베이스 구축

한다(그림 4-b). 엠보싱 데이터는 안료의 깊이 질감(Impasto)표현을 위해 사용되는 데이터이다. 이는 스트로크의 표현방법(3.2절)에서 자세히 설명하겠다. 인덱스 이미지는 빠른 비교 계산을 위해 어느 정도 작은 크기의 이미지를 사용한다.

3.2 질감표현을 위한 스트로크 모델

우리는 3.1.2절에서 설명된 스트로크 데이터베이스에서 구성요소 중 브러시 스트로크 텍스처와 엠보싱(양각)이미지 활용하여 효과적인 스트로크의 질감 표현 기술을 제안한다. 그 과정은 다음과 같다. 스트로크 텍스처는 스트로크의 형태를 표현하는 투명정보로 활용된다. 임계치(TH)이상의 값을 갖는 부분은 투명하게 그려진다. 엠보싱 이미지는 텍스처의 깊이를 표현하기 위해 사용된다. 이는 엠보싱 필터(Embossing Filter)에 의해 만들어 진다. 이 필터는 주어진 빛의 방향에 따른 픽셀들의 색상 높이 차이를 수치화한 것이며 색상 표현가능 범위의 중간 값의 상대적인 값으로 저장된다.

스트로크를 그리기 위해서, 우리는 HSB 색상 모델을 사용한다. 먼저 브러시 스트로크의 RGB 색상을 HSB 색상으로 변경한다. HSB색상의 휴(Hue)와 채도(Saturation) 값은 그대로 상속되고 밝기 정보를 나타내는 밝기 값(Brightness)은 엠보싱 데이터를 참고로 하여 수정된다. 또한 엠보싱 데이터의 역(ET = -1)으로 사용하게 되면 음각 스트로크를 표현할 수도 있다. 이는 엠보싱 데이터 생성 시 빛의 방향을 반대로 하는 것과 같은 효과를 주며 안료가 마르지 않은 캔버스위에 스트로크를 덧칠하는 효과를 표현할 수 있다. 다음의 수도코드는 브러시 표현의 과정을 보여주고 있다. 그리고 이러한 과정은 그림 5-a에



(a) 스트로크 표현



(b) 음각/양각 혼합된 렌더링 영상

그림 5. 스트로크의 표현 프로세스와 렌더링 예

서 보여주고 있다.

우리가 제시한 방법은 물리적인 계산을 하지 않고 사용할 수 있는 아주 유용한 브러시 질감표현 방법이라고 할 수 있다. Hertzmann[12]은 2002년도에 질감표현을 위해 적층된 텍스처의 높이정보를 셰이딩 하여 최종 질감효과를 표현하였다. 이는 아주 간단하면서도 효율적인 방법이었다. 하지만 렌더링이 종료된 후 최종 스트로크의 높이 정보를 활용하여 질감표현을 했기 때문에 중간과정에서의 질감표현에는 어려움이 존재하였다. 또한 Baxter[17]가 제안한 방법은 안료의 물리적인 시뮬레이션으로 그 질감효과는 탁월하나 계산과정이 복잡하여 하드웨어에 의존해야 했다. 하지만 우리의 질감표현 기법은 단순한 픽셀계산만을 이용해 그 계산 비용이 적어 렌더링 중간단계에서도 그 질감효과를 확인할 수 있으며 스트로크의 시각적 효과 또한 우수하다. 물리적인 시뮬레이션을 통해서만 얻을 수 있는 음각의 스트로크의 표현형태도 쉽게 얻을 수 있는 장점을 가지고 있다.

그림5-b는 양각/음각의 효과를 갖는 브러시 스트로크를 캔버스에 그린 예를 보여주고 있다. 대단히 자연스러운 질감효과(Impasto)가 보인다는 것을 알 수 있다. 표 1은 스트로크 생성의 수도코드를 보여주고 있다.

3.3 브러시 스트로크의 시도 위치, 크기, 색상

우리는 Hertzmann[10]의 레이어드 그리드(Layered

표 1. 스트로크 생성 수도코드

```
function depth_effect(Brush Color C, Brush Texture
T, Embossing Data E){
    R := a new blank image
    Chsb = HSB(C) //RGB를 HSB모델로 변환
    for x=0 to T.width //T, S의 크기는 같음
        for y=0 to T.height{
            Alpha = T(x,y) //T(x,y) 밝기 값
            if(Alpha < TH){ //Find Transparency
                //양각(ET= 1), 음각(ET = -1)
                Chsb.brightness := Chsb.brightness +
                ET*(E(x,y)-128))
                R(x,y):= RGB(Chsb)
            }
            else{
                R(x,y): = Transparent value
            }
        }
    return R
}
```

Grid)기반 스트로크 생성 방법과 거의 같은 방식으로 스트로크의 위치와 크기를 결정한다. 그는 입력 영상의 LOD(Level Of Detail)을 표현하기 위해 브러시 사이즈 리스트를 큰 순서에서 작은 순서로 정의한다. 그리고 그리드 시스템을 이용해 각 그리드 셀에서 완료되지 않은 캔버스와 소스 영상과의 차이가 심한 곳(Maximum Error Point)을 찾고 그 지점에서의 이미지 기울기(Gradient) 정보를 이용해 새로운 스트로크를 칠해나가는 방법을 사용한다. 이것은 회화적 영상을 표현하는 매우 효율적인 방법이며 대부분의 스트로크 기반 회화적 표현 알고리즘에서 차용하고 있다.

우리는 Hertzmann의 접근법을 사용한다. 하지만 세부적인 부분에서 약간의 차이점을 가지고 있다. 첫 번째가 브러시의 크기이다. 우리는 미리 만들어진 브러시 스트로크 텍스처를 활용한다. 그러므로 우리의 브러시 크기 브러시의 반지름이 아니라 붙여질 브러시 텍스처의 크기를 의미하며 브러시 크기는 브러시 크기 리스트 $B_i = \{B_0, B_1, \dots, B_n\}$ 으로 정의 되고 각 브러시 사이즈에 해당하는 그리드 사이즈는 $G_i = F_y \times B_i$ 으로 결정된다.

그리고 우리가 사용하는 스트로크 텍스처의 의미 있는 영역(Shape Area)이 텍스처 전체 영역의 일부분이기 때문에 각 브러시 크기별(B_i) 격자의 간격(G_i)은 브러시의 크기보다 작은 것을 사용한다.

두 번째는 브러시의 시작위치이다. 우리도 Hertzmann과 같이 그리드 시스템을 이용한다. 그러나 우리의 스트로크 위치는 캔버스에 그려진 중간결과에 영향을 받지 않고 각 브러시 크기별로 한번에 계산되며 각 브러시 크기에 따라 생성되는 각 그리드 셀 안에서 랜덤하게 결정한다. 이는 스트로크의 일정간격으로 반복되는 것을 피하고 캔버스와 입력영상의 차이가 심한 최대 에러 위치(Maximum Error Position)를 찾는 과정에 대한 비용을 절약하는 장점이 있다.

스트로크의 위치는 브러시가 시작되는 점이 아니라 텍스처가 붙여질 위치의 중심점(Center Position)을 의미하며 중심점의 색상을 스트로크의 색으로 사용한다. 마지막으로 각 레이어에서 계산된 스트로크의 위치들은 입력영상의 밝기 값에 따라 옅음차순으로 정렬된다. 이는 어두운 곳에서 밝은 곳으로 그려나가는 일반적인 페인팅 시퀀스를 따르기 위해서이다.

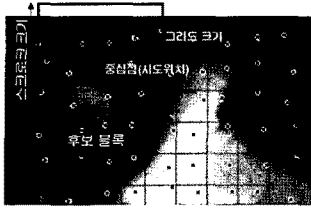


그림 6. 스트로크의 시도위치와 후보블록

우리는 브러시를 정사각형 텍스처로 사용하고 스트로크 텍스처를 검색기법을 이용해 그려나가기 때문에 몇 가지 용어를 다음과 같이 변경하여 사용하겠다. 스트로크 위치를 시도위치(Try Position)로 부르고 시도위치와 브러시 크기에 해당하는 영역을 후보블록(Candidate Block)이라고 정의한다. 이는 다음에 이어지는 3.4절과 3.5절을 이해하는데 도움을 준다. 후보 블록에 반드시 스트로크 텍스처가 그려지는 것은 아니며 스트로크의 수락 여부를 결정하는 과정에서 실제 그려질지를 판단하기 때문이다. 그림 6은 하나의 브러시 사이즈에 의해 생성되는 시도위치와 후보 블록 보여주고 있다.

3.4 스트로크 텍스처 검색

구축된 스트로크 데이터베이스는 스트로크 텍스처, 엠보싱 데이터, 그리고 검색을 위한 인덱스 이미지의 세 가지 데이터 형태로 구성된다. 우리는 3.2절에서 계산되어진 스트로크 시도위치와 크기에 의해 생성되는 입력영상의 사각형 영역(후보블록)을 결정하였다. 우리의 목표는 후보 블록 안에 존재하는 입력영상을 표현하기에 적당한 스트로크를 데이터베이스에서 찾는 것이다. 후보 블록 이미지와 데이터베이스 안에 있는 스트로크 텍스처 사이의 평균 색상 거리(Average Color Difference)가 검색기준이 된다. 이는 포토모자이크 알고리즘[7]에서 부타일 이미지(Subtile Image)를 찾을 때 사용하는 기준과 같다. 두 개의 이미지, I_1, I_2 사이의 평균 색상 거리, $D(I_1, I_2)$ 는 RGB 공간에서 두 이미지 좌표(x, y)에서의 각 픽셀 값 $I_1(x, y) = (r_1, g_1, b_1)$ $I_2(x, y) = (r_2, g_2, b_2)$ 의 RGB 각 색상의 유클리디안 거리(Euclidean Distance)의 평균으로 계산된다(수식-1).

$$D(I_1, I_2) = \frac{\sum_x \sum_y^h (|I_1(x, y) - I_2(x, y)|)}{w \times h} \quad (1)$$

$$|I_1(x, y) - I_2(x, y)| = \sqrt{(r_1 - r_2)^2 + (g_1 - g_2)^2 + (b_1 - b_2)^2}$$

이러한 검색 과정은 인간이 하나의 스트로크를 생성하는 과정과 유사성을 갖는다.

그림 7은 앞서 말한 검색 과정을 설명하고 있다. 데이터베이스는 스트로크 텍스처와 인덱스 이미지의 집합으로 구성된다(그림 7-(a)). 먼저 입력영상 I 로부터 후보 블록의 이미지 $B(I)$ 를 잘라낸 후 인덱스 이미지의 크기($w \times h$)로 흑백 변환된 이미지, $\bar{B}(I)$ 를 생성한다. $\bar{B}(I)$ 와 데이터베이스의 인덱스 이미지, DB_i 와의 색상 거리 값이 작은 순서대로 K 개의 스트로크, $T = \{T_1, T_2, \dots, T_k\}$ 를 찾은 후, 최종 하나의 스트로크 텍스처 T_i 를 선택한다.

$$T \leftarrow \text{Min}_k \{D(DB_i, \bar{B})\}, \text{ for } DB_i \in \text{Strokes DB}\}$$

$$T_i = \text{Sampling by Cosine probability}(T)$$

K 값은 스트로크의 정확성을 의미한다. K 값이 1에 가까울 경우 입력영상의 동질성 영역에서 같은 형태의 스트로크들이 반복되는 문제점을 내포하고 있다. 이는 같은 색상정보를 갖는 후보 블록에 항상 같은 스트로크 텍스처가 최적의 텍스처로 검색되기 때문이다. 이러한 문제는 그림 8의 (a)의 테두리 부분에서 발견할 수 있다. 반면 K 가 커지게 되면 결과영상에서 스트로크의 불규칙성이 높아지고 입력영상의 내부의 특징이 되는 형태정보가 손실될 가능성이 높아진다. 그러므로 K 는 적절히 조절되어야 한다.

본 연구에서는 K 가 커져도 일관성을 일정하게 유지하기 위해 데이터베이스에서 최적의 스트로크만을 검색하지는 않는다. 이를 위해 색상거리가 작은 값의 선택 확률을 높이고 작은 값의 선택 확률을 낮추는 샘플링 방법을 사용한다. 우리는 선택확률을 코

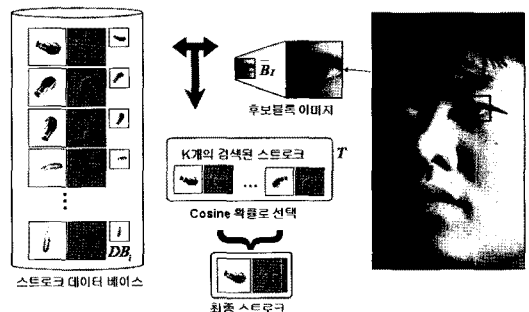


그림 7. 스트로크의 선택과정

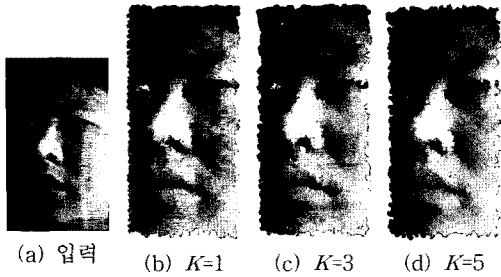


그림 8. K값의 변화에 따른 결과영상

사인 함수(Cosine Function)를 사용하였다. 이는 자연스러움을 위한 것이며 변화와 반복의 균형을 맞추기 위해서 이다.

그림 8은 K값의 변화에 따른 결과 영상을 보여주고 있다. K값이 커짐에 따라 입력영상의 경계가 정확하게 지켜지지 않음을 볼 수 있다.

스트로크 데이터베이스에 있는 스트로크 텍스처의 수는 무한히 확장이 가능하다. 무수히 많은 데이터베이스의 확장은 입력영상(대상)을 스트로크 수가 적은 데이터베이스보다 정확히 표현할 수 있을 것이다. 하지만 검색속도 등을 위해서 효율적으로 구축되어야 한다.

3.5 드로잉 판단 과정

본 절에서는 검색된 스트로크 텍스처를 캔버스에 적용하는 과정을 설명한다. 우리는 선택된 스트로크가 후보 블록에 그려진 후 입력이미지(대상)를 훼손하는지 보완하는지를 찾는다. 이와 같은 과정은 실제 화가들도 자신의 스트로크가 적절한지 아닌지의 판단이 머릿속에서 이루어진다. 우리는 이러한 판단과정에 대한 기준을 제안한다.

우리는 새로운 스트로크가 그려진 캔버스 영역과 입력 영상의 영역의 평균 색상 거리가 줄어들었는지 아닌지를 판단한다. 이를 위해서, 다음의 세 가지 데이터가 필요하다. 그 세 가지는 후보 블록에 해당하는 입력이미지 영역($B(I)$), 새 스트로크가 적용전의 캔버스 이미지 영역($\hat{B}(C)$), 그리고 새 스트로크가 적용후의 캔버스 이미지 영역($\bar{B}(C)$)이다. 최종 후보 블록에 해당하는 캔버스 이미지 $B(C)$ 는 다음의 수식에 의해 결정된다.

$$B(C) = \begin{cases} \hat{B}(C), & \text{if } comp < F \\ \bar{B}(C), & \text{otherwise} \end{cases}$$

$$comp = D(B(I), \bar{B}(C)) - D(B(I), \hat{B}(C))$$

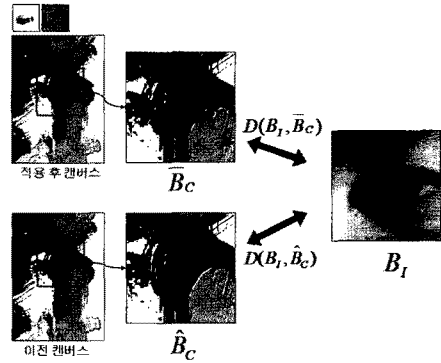


그림 9. 스트로크의 판단 과정

그림 9는 스트로크 드로잉 판단 과정(Judgement Process)을 간략하게 보여주고 있다. $D(B(I), \bar{B}(C))$ 가 $D(B(I), \hat{B}(C))$ 보다 작다면 이는 새로운 스트로크를 적용시켰을 때 이미지를 손상시킨다는 것을 의미하기 때문에 그려진 스트로크를 취소한다.

우리는 알고리즘의 다변성을 위해 제어 변수 F 를 정의한다. F 값은 스트로크의 시도에 대한 그려지는 스트로크의 적용 비율(Acceptance Ratio)을 제어하며, 이론적으로 평균색상 거리의 최대값 (1)과 최소값(-1) 사이의 값을 가질 수 있다. F 값이 -1의 의미는 그려지는 스트로크가 없다는 것을 F 값이 1이면 시도되는 모든 스트로크들을 그리는 것을 의미한다. 기본적으로 F 은 0을 사용한다. 많은 실험을 통해 우리의 알고리즘은 F 가 -0.1에서 0.1사이의 값에서 민감하며 이 사이의 값을 사용하는 것이 적당한 결과를 얻을 수 있었다. 그림 10은 F 값이 변화에 따른

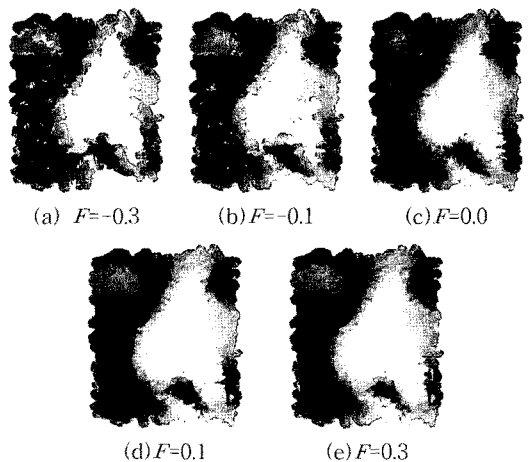


그림 10. F값의 변화에 따른 결과 (K=5)

스트로크의 변화를 보여주고 있다. F 값이 작아지면 평균 색상 거리가 큰 부분, 즉 어두운 부분에서 적용 비율이 높아진다는 것을 알 수 있다. F 값의 변화에 따라 결과물에 다양한 변이를 줄 수 있다.

4. 실험 및 결과

우리의 알고리즘은 확장성 있는 회화적 렌더링 프레임워크를 제공한다. 몇 개의 렌더링 인수의 변화를 통해 렌더링 프로세스를 제어할 수 있다. 또한 우리의 논문에서는 주로 유화의 관점에서 실험한 결과를 보여주고 있다. 하지만 사용자에게 스트로크 데이터베이스의 변경을 용인함으로써 다양한 회화 안료의 종류를 생성할 수 있다. 우리는 렌더링 결과에 영향을 많이 주는 몇 개의 대표적인 스타일 인수(Style Parameters)와 스트로크 데이터베이스의 종류(Type)를 사용하여 실험하였으며 표 2와 같다.

그림 11은 같은 원본 영상을 유화, 파스텔, 색연필, 흑백연필, 스타일의 프로파일을 기반으로 구축한 데

이터베이스를 이용해 렌더링한 다양한 스타일의 결과를 보여주고 있다. 그림 12는 기존에 존재하지 않는 독특한 사용자 정의 데이터베이스를 이용한 결과 영상이다. 왼쪽 작은 이미지들은 사용된 프로파일들이다.

5. 결과 비교

표 3은 이미지 기반 회화적 렌더링 연구 중 대표적인 Hertzmann[10,12] 과 Hays[13]의 연구와 우리의 연구를 비교하고 있다. 그림 13은 질감표현이 명확한 Hertzmann[12]의 결과와 우리의 것을 비교한 영상이다. 두 가지 연구 모두 우수한 시각적 효과를 얻을 수 있다.

표 3과 그림 11, 12, 13에서 볼 수 있듯이 제안된 알고리즘은 기존 연구에 비해 아래와 같은 장점을 가진다.

- 확장성: 스트로크 데이터베이스 변화를 통해 다양한 회화매체로 표현된 결과물 생성가능

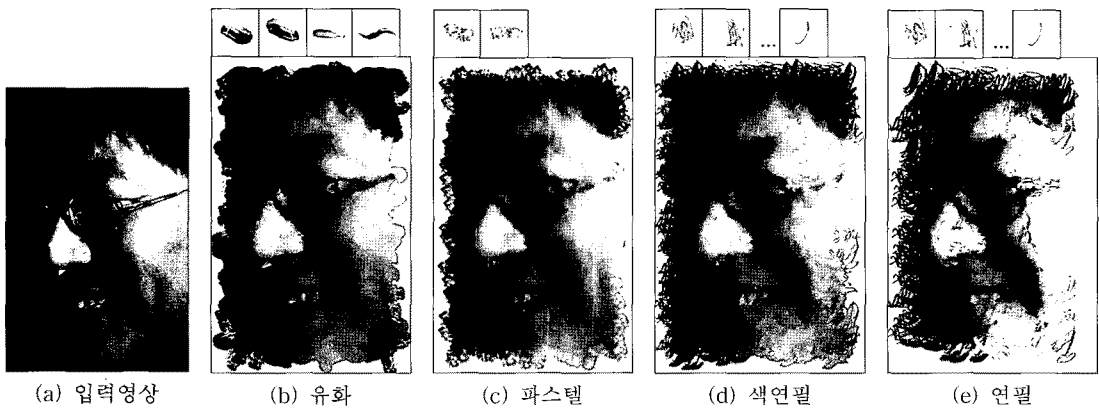


그림 11. 스트로크 프로파일과 데이터베이스의 변화에 따른 다양한 스타일의 결과영상($DB_{type} =$ 유화(b), 파스텔(c), 색연필화(d), 연필화(e), $F_y = 0.25$, $B_n = \{200, 150, 60, 40\}$, $DB_{cut} = 100$, $K = 5$, $F = 0$)

표 2. 실험에 사용한 스타일 인수

렌더링 스타일 인수	설명
스트로크 데이터베이스 종류(DB_{type})	유화, 파스텔, 색연필, 연필 등(3.1절)
데이터베이스의 스트로크 개수(DB_{cut})	데이터베이스를 구성하는 스트로크의 개수
스트로크 크기 리스트(B_n)	시도하는 스트로크의 크기 리스트(3.3절)
그리드 크기(F_y)	후보블록의 그리드 간격 제어(3.3절)
스트로크 검색 개수(K)	데이터베이스로 검색하는 스트로크의 개수 (3.4절)
적용비율 제어 변수(F)	검색된 스트로크의 적용여부를 제어 (3.5절)

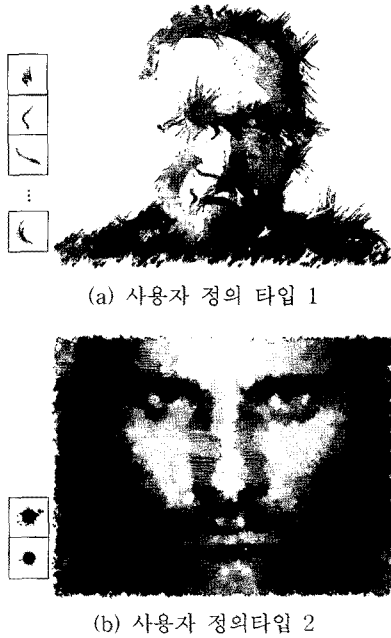


그림 12. 사용자 정의 타입을 이용한 결과영상($DB_{opt} =$ 사용자 정의1(a), 2(b), $DB_{cut} = 200$, $F_g = 0.25$, $B_n = \{200, 150, 100, 80, 40\}$, $K = 10$, $F_g = 10$, $F = 0$)

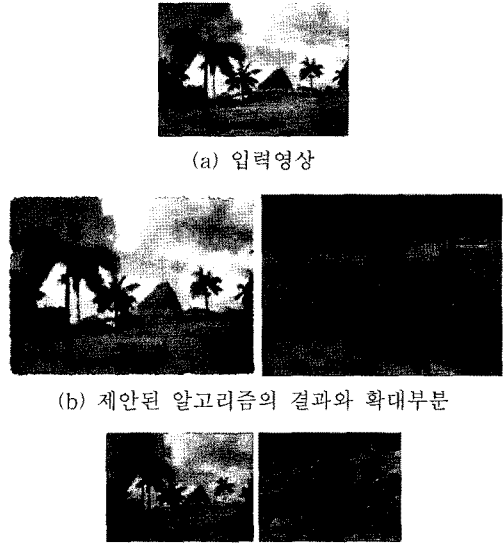


그림 13. 브러시 스트로크의 질감표현 비교

시각적 효과가 뛰어난 사실적인 페인팅 결과를 얻을 수 있음

표 3. 기존연구와의 비교

구 분	제안된 알고리즘	Hertzmann	Hays
표현 매체 종류	유화, 파스텔화, 연필화, 색연필화, 점묘화 등	유화, 점묘화, 수채화	유화, 점묘화, 수채화
표현 확장성	표현매체 확장가능	확장 불가	확장 불가
스트로크 질감 표현	내장 셰이딩 정보사용 질감 표현 명확함 양각, 음각모두 표현 가능	조명계산 질감 명확함 양각	단일 텍스처 질감 약함 양각
추상화 정도	다양한 추상화 변이 허용	한정된 변이 허용	한정된 변이 허용
스트로크의 형태	직선, 곡선 프로파일 에 따른 다양한 형태	곡선	직선

- 유연성: 스트로크의 조합방법(선택기준)을 제어함으로써 회화 결과물의 변환과 추상화 정도를 다양하게 조절가능

- 효율성: 실제 화가의 스트로크 샘플을 이용해 데이터베이스를 구축하고 간단하지만 물리적인 계산 없이 깊이정보 포함된 스트로크를 생성함으로써

6. 응용분야

우리의 알고리즘은 하나의 프레임 워크에서 다양한 스타일을 제공할 수 있는 확장성을 갖는다. 또한 낮은 컴퓨팅 타임과 표현력(Visual Effect)가 높은 브러시 표현기술로 인해 인터랙티브한 전시시스템에 적용 가능하며 복잡한 물리적 계산이나 복잡한 플로팅 연산이 필요 없기 때문에 모바일 기기와 같은 내장형 시스템(Embedded System)에도 쉽게 적용이 가능하다. 그림 14는 우리의 알고리즘으로 실제로 개발된 휴대형 인물화 제작 시스템과 동작과정을 보여주고 있다. 휴대성을 높이기 위해 여행용 가방크기로 제작되었으며 터치스크린, 컴퓨터본체(Mini Mac), 프린터, 웹캠으로 구성되어 있다. 동작과정은 캠으로 들어온 사진을 보면서 원하는 스타일(유화, 파스텔

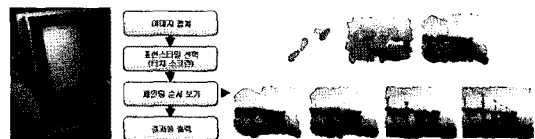


그림 14. 휴대형 인물화 제작 시스템 구성도

화, 언필화 등)을 선택하게 되면 선택된 스타일의 결과물을 페인팅과정을 보여주게 된다. 페인팅이 종료되면 프린터를 통해 출력된 결과물을 얻을 수 있다.

본 시스템의 환경은 Window XP, 1GB 메모리, 1.8GHz CPU이며 768×1024의 해상도를 갖는 이미지를 30초 이내로 렌더링할 수 있었다.

7. 결론 및 향후 연구

우리는 이미지로부터 회화 스타일의 결과물을 생산하는 새로운 방식을 소개하였다. 우리의 제안한 방식은 기본적인 이미지 검색 기술에 그 근간을 두고 있으며 기존의 이미지 기반 회화적 렌더링 방법에서 추구하던 패러다임인 브러시 스트로크의 자동 검출/생성하는 결정적인(Deterministic Method) 방법과 달리 간단하지만 효율적이고 결과의 다양한 변이가 가능하도록 하는 유연성 있는 과정들로 구성되어 있다.

스트로크의 종류와 형태를 사용자로 하여금 프로파일이라는 이름으로 정의할 수 있도록 하였으며 몇 개의 브러시 스트로크 샘플인 프로파일의 변환을 통해 대용량의 데이터베이스를 구축할 수 있도록 하였다. 이를 기반으로 입력영상을 사용자 정의 스트로크들의 조합을 통해 결과물을 만들어 내도록 하였다. 그리고 우리는 몇 개의 렌더링 스타일 인수를 제공함으로써 결과물의 다양한 변이를 용납하고 있다. 또한 단순한 엠보싱필터를 활용한 단순한 스트로크 모델을 사용하여 물리적인 계산을 하지 않고 사용할 수 있는 아주 유용한 브러시 질감표현 방법을 소개하고 있다.

우리의 접근방법은 몇 가지 장점을 가지고 있다. 첫째, 스트로크 데이터베이스의 변화를 통해 다양한 회화 매체(안료)로 표현된 회화영상을 만들 수 있는 확장성을 제공한다. 두 번째, 스트로크의 조합방법(선택기준)을 제어함으로써 회화 결과물의 변환과 추상화 정도를 다양하게 표현할 수 있는 유연성을 가지고 있다. 셋째 간단한 DB구조를 통해 효율적인 스트로크의 질감 표현기법을 제안함으로써 시각적 효과가 뛰어난 사실적인 페인팅 결과를 얻을 수 있다.

우리의 알고리즘에서 페인팅 결과의 질과 성능을 좌우 하는 것은 데이터베이스의 스트로크의 종류 및 수량이다. 데이터베이스 안에 다양한 형태와 방향을 갖는 스트로크 텍스처들이 많이 존재한다면 좋은 결

과를 얻을 수 있다. 하지만 우리 알고리즘 성능의 병목현상은 데이터베이스로 부터 최적의 스트로크를 검색하는 과정이기 때문에 전시 시스템과 같은 실시간성을 요구하는 부분에서 우리의 알고리즘을 사용하기 위해서는 데이터베이스의 양을 줄여야 하는 한계를 가지고 있다. 이러한 문제를 해결하기 위해 데이터베이스 내에서 효율적인 스트로크 형태의 분포 및 배열 방법, 그리고 검색속도 향상기법에 대한 연구가 진행되어야 한다. 또한 수채화, 펜화 등과 같이 다른 안료의 회화적 표현을 위해 추가적인 렌더링 인수의 추가와 후처리 방법에 대한 다양한 실험과 연구가 진행되어야 한다.

참 고 문 헌

- [1] T. Strouthotte and S. Schlechtweg, Non-photorealistic computer graphics: modeling, rendering, and animation, Morgan Kaufmann Publishers Inc, San Francisco, 2002
- [2] M.P. Salisbury, M.T. Wong, J.F. Hughes and D.H. Salesin, "Orientable textures for image-based pen-and ink illustration," In Proc. SIGGRAPH'97, pp. 401-406, 1997.
- [3] C.J. Curtis, S.E. Anderson, J.E. Seims, K.W. Fleischer and D.H. Salesin, "Computer generated watercolor," In Proc. of SIGGRAPH'97, pp. 421-430, 1997.
- [4] D.W. Kang, D.H. Kim and K.H. Yoon "A study on the real-time toon rendering for 3D geometry model," In Proc. of IV'01, pp. 391-396, 2001.
- [5] A. Majumder and M. Gopi, "Hardware accelerated realtime charcoal rendering," In Proc. of NPAR'02, pp. 59-66, 2002.
- [6] H.J. Lee, S.T. Kwon and S.Y. Lee, "Real-time pencil rendering," In Proc. of NPAR'06, pp. 37-45, 2006.
- [7] R. Silvers and M. Hawley, Photomosaics, Henry Holt, New York, 1997.
- [8] P. Haeberli, "Paint by numbers: abstract image representations," In Proc. of SIGGRAPH'90, pp. 207-214, 1990.

[9] P. Litwinowicz, "Processing images and video for an impressionist effect," In Proc. of SIGGRAPH'97, pp. 407-414, 1997.

[10] A. Hertzmann, "Painterly rendering with curved brush strokes of multiple sizes," In Proc. of SIGGRAPH'98, pp. 453-460, 1998.

[11] A. Hertzmann, and K. Perlin, "Painterly rendering for video and interaction," In Proc. of NPAR'00, pp. 7-12, 2000.

[12] A. Hertzmann, "Fast paint texture," In Proc. of NPAR'02, pp. 91-96, 2002.

[13] J. Hays and I. Essa, "Image and video based painterly animation," In Proc. of NPAR'04 pp. 113-120, 2004.

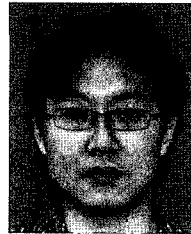
[14] D. Nehab and L. Velho, "Multiscale moment-based painterly rendering," In Proc. of SIBGRAPT'02, pp. 453-460, 2002.

[15] M. Shiraishi and Y.i Yamaguchi, "An algorithm for automatic painterly rendering based on local source image approximation," In Proc. of NPAR'00 pp. 53-58, 2000.

[16] B. Gooch and G. Coombe, and P. Shirley, "Artistic vision: painterly rendering using computer vision techniques," In Proc. of NPAR'02, pp. 83-90, 2002.

[17] M. SHugrina, M. Betke and J. Collomosse, "Empathic painting: interactive stylization through observed emotional state," In Proc. of NPAR'06, pp. 87-96, 2006.

[18] W. Baxter, J. Wendt and M, Lin, "IMPaSTo: A realistic, interactive model for paint," In Proc. of NPAR'04, pp. 45-56, 2004.



서 상 현

1998년 중앙대학교 공과대학 컴퓨터공학과 졸업
 2000년 중앙대학교 첨단영상대학원 영상공학과 석사졸업
 2003년~2005년 (주)지노시스템 기술연구소 선임연구원

2001년~현재 중앙대학교 첨단영상대학원 영상공학과 박사과정 재학중
 관심분야 : 컴퓨터 그래픽스, 비사실적 렌더링, Proceduralism, GIS



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
 관심분야 : 컴퓨터 그래픽스, 렌더링, Proceduralism, GIS, 영상기반모델링과 렌더링, 비사실적 렌더링