

입자 패치 기반 가상 연필 및 에어브러시 가시화 알고리즘

이혜린⁰

rin0722@kpu.ac.kr

오건

한국산업기술대학교
ohhh123@naver.com

이택희*

watersp@kpu.ac.kr

Virtual Pencil and Airbrush Rendering Algorithm Using Particle Patch

Hye Rin Lee⁰

Geon Oh

Taek Hee Lee*

Korea polytechnic university

요약

최근 가상 현실 및 증강 현실 기술의 발달로 다양한 형태의 기술이 제안되고 발전되고 있다. 특히 가상의 학습공간이나 가상의 건축 공간들을 위한 기술로서 원하는 형태의 그림을 그릴 수 있는 가상 드로잉 툴은 필수적인 요소 중 하나로 볼 수 있다. 기존의 래스터 기반 드로잉 알고리즘은 공간 이동이 잦은 가상 현실에서는 부적합할 수 있는데, 가상 현실의 특성상 근접하거나 멀어지는 경우가 많기 때문이다. 본 논문에서는 연필 및 에어브러시의 물리적인 특성을 고려하여 실제 결과와 흡사하면서도 확대 및 축소에 강건한 가상의 드로잉 툴 렌더링 알고리즘을 제안한다. 캔버스로부터의 거리, 분사되는 잉크의 양, 움직이는 속도, 마찰력, 압력 등을 고려한다. 드로잉의 기본 단위는 정사각형 형태의 패치는 사용하며 각 패치는 슈도 랜덤 값을 기반으로 하는 입자 형태의 이미지를 가진다. 이러한 입자 패치는 슈도 랜덤 식을 사용하고 매 프레임마다 재생성되기 때문에 확대 축소 시 깨짐 현상이 발생하지 않는다. 제안된 알고리즘은 2차원 환경 및 가상 현실 환경에 구현되었다. 2차원 환경은 안드로이드 기반 스마트폰에서 구현하였으며 가상 현실 환경은 언리얼 엔진 4를 사용하여 구현하였다. 구현 결과 확대 축소 시에도 이질감 없이 원래의 질감을 유지하는 렌더링 결과를 얻을 수 있었다.

Abstract

Recently, the improvement of virtual reality and augmented reality technologies leverages many new technologies like the virtual study room, virtual architecture room. Such virtual worlds require free handed drawing technology such as writing descriptions of formula or drawing blue print of buildings. In nature, lots of view point modifications occur when we walk around inside the virtual world. Especially, we often look some objects from near to far distance in the virtual world. Traditional drawing methods like using fixed size image for drawing unit is not produce acceptable result because they generate blurred and jaggy result as view distance varying. We propose a novel method which robust to the environment that produce lots of magnifications and minimizations like the virtual reality world. We implemented our algorithm both two dimensional and three dimensional devices. Our algorithm does not produce any artifacts, jaggy or blurred result regardless of scaling factor.

키워드: 가상 현실, 드로잉 툴, 연필, 에어브러시

Keywords: Virtual reality, drawing tool, pencil, airbrush

1. 서론

최근 가상 현실 및 증강 현실 기술의 발달로 다양한 형태의 기술이 제안되고 발전되고 있다. 특히 가상의 학습공간이나 가상의 건축 공간들을 위한 기술로서 원하는 형태의 그림을 그릴 수 있는 가상 드로잉 툴은 필수적인 요소 중 하나로 볼 수 있다. 일반적인 래스터 기반의 방식으로 그림을 그리는 알고리즘은 2차

원 디스플레이에는 적합하나 공간 이동이 잦은 가상 현실에서는 부적합할 수 있다. 근접하여 그리거나 멀어져서 그리는 경우가 혼재되어 있고 이럴 경우 확대 축소에 따른 퍼짐 현상이나 계단 현상이 나타날 수 있기 때문이다. 이를 해결하기 위해선 벡터 그래픽스와 같이 확대 및 축소에 강한 드로잉 알고리즘이 필요하다.

본 논문에서는 연필 및 에어브러시의 물리적인 특성을 고려하면서도 확대 및 축소에 강건한 가상의 드로잉 툴 렌더링 알고리즘을 제안한다. 실제 에어브러시는 캔버스로부터의 거리, 분사

*corresponding author: Taek Hee Lee/Korea polytechnic university(watersp@kpu.ac.kr)

Received : 2018.06.26./ Review completed : 1st 2018.06.29. / Accepted : 2018.07.04.

DOI : 10.15701/kcgs.2018.24.3.101

ISSN : 1975-7883(Print)/2383-529X(Online)

되는 잉크의 양, 움직이는 속도에 따라 그려지는 형태가 변화한다. 실제 연필은 캔버스와 흑연 사이의 마찰력 및 압력에 따라 형태가 변화한다. 가상의 드로잉 툴은 위와 같은 물리적 특성들을 고려하여야 실제와 흡사한 렌더링 결과를 얻을 수 있다. 이를 위해 거리, 분사 속도, 마찰력을 입력으로 하며 렌더링 될 밀도를 출력으로 하는 매핑 테이블을 구성한다. 특히 연필의 경우 흑연이 가지는 강한 정반사 특성을 고려하여야 한다. 이를 위해 무작위 샘플링을 통해 부분적으로 정반사 특성을 강하게 주어 실제 연필 드로잉 결과와 흡사한 결과를 낼 수 있도록 한다.

매핑 테이블을 거쳐 결정된 밀도를 가상의 환경에서 표현하기 위해 사각형 패치를 사용한다. 밀도 값은 슈도 랜덤 값을 기반으로 하는 평면 입자 형태로 변환되며 이 평면 입자는 다시 원 형태의 커널을 거쳐 패치에 입혀진다. 이때 입자가 존재하는 영역의 투명도는 0 이상의 값을 가지며 존재하지 않는 영역은 0의 투명도를 가진다(이후 ‘입자 패치’라 칭함). 입자 패치는 정해진 시간 간격 및 드로잉 툴의 움직인 거리에 기반하여 생성된다. 생성된 패치는 마지막으로 평면(캔버스)에 중첩되어 그려진다. 본 논문의 입자 패치는 슈도 랜덤 식을 사용한다는 점에서 기존에 방식과 차별화 될 수 있다. 기존 방식은 에어브러시의 미세 잉크 표현이나 흑연이 뭉개지는 질감을 표현하기 위해 이미지를 사용한다. 이러한 방식은 실제 드로잉 툴의 결과 이미지를 사용할 수 있어 실제와 흡사한 결과를 만들어 낼 수 있는 장점이 있는 반면 확대 축소 시 깨짐 현상이 발생한다. 제안하는 방식은 매 렌더링 시 입자 패치의 질감을 크기에 따라 다시 계산하여 매핑하는 방식이다. 슈도 랜덤 기반이기 때문에 입자 패치가 매우 빠르게 생성이 되며 확대 축소 시에도 원래의 질감을 유지하기 때문에 가상 환경에 적합한 드로잉 알고리즘이라 할 수 있다.

제안된 알고리즘은 2차원 환경 및 가상 현실 환경에 구현되었다. 2차원 환경은 안드로이드 기반 스마트폰에서 구동된다. 가상 현실 환경은 언리얼 엔진 4를 사용하여 구현하였다. 구현 결과 확대 축소 시에도 이질감 없이 원래의 질감을 유지하는 렌더링 결과를 얻을 수 있었다. 특히 가상 현실 환경에서는 확대 축소에 강건한 가상 드로잉 툴 알고리즘이 매우 효과적임을 확인할 수 있었다.

2. 관련 연구

펜을 기반으로 하는 드로잉 알고리즘은 크게 벡터 그래픽스 기반과 래스터 그래픽스 기반으로 나눌 수 있다.

벡터 그래픽스 기반의 알고리즘은 스트로크를 커브의 조합으

로 분해하여 정보를 저장한다. 따라서 확대 혹은 축소에 따라 형태의 일그러짐이 발생하지 않는다 [1, 2]. 다만 벡터 그래픽스 기반의 자유 곡선 렌더링 기법은 스트로크 생성 방식이 까다롭고 급격한 스트로크 방향 변화 시 커브가 꼬이는 현상이 발생할 수 있다 [2]. 이러한 어려움으로 인해 자유 곡선을 벡터 그래픽스 기반으로 렌더링 하는 알고리즘은 많지 않은 상태이다.

래스터 그래픽스 기반의 스트로크 생성 기법은 가장 널리 사용되는 방식이다. 스타일러스 펜 등과 같은 입력 장치는 사용자의 손동작에 따른 2차원 좌표를 빠르게 생성한다. 래스터 기반의 방식은 이러한 좌표에 원 형태의 패치를 중첩하여 스트로크를 표현한다. 벡터 기반 방식에 비해 쉽게 구현이 가능하며 다양한 효과를 효율적으로 적용할 수 있기 때문에 많은 연구가 이루어져 있다[3]. 다만 확대 및 축소 시 픽셀이 도드라져 보이는 단점이 있다. 즉 블러와 앨리어싱 현상이 생기고, 벡터 그래픽스와 결과와 같이 깨짐 없는 연속적인 이미지를 보여주는 것이 불가능하다.

본 논문에서 제안하는 방식은 래스터 그래픽스의 패치 기반 렌더링 방식을 사용하되 패치의 내부에 입혀지는 텍스처는 실시간으로 생성하는 방식이다. 따라서 래스터 기반 방식의 용이한 구현 난이도의 장점을 취하면서도 벡터 그래픽스 방식의 확대 및 축소에 강건한 장점까지 취할 수 있다.

3. 입자 패치

이번 장에선 알고리즘의 가장 기본이 되는 입자 패치에 대한 내용을 다룬다. 잉크가 뿌려진 결과 및 흑연이 뭉개진 결과를 흉내내기 위해 필요한 매 프레임당 실시간으로 생성되는 랜덤 이미지를 기반으로 랜덤 이미지를 생성한다. 이때 정사각형 형태의 패치 상에 랜덤 이미지가 매핑 되어 생성되며 사용 목적에 따라 입혀지는 필터가 바뀌고 물리적인 특성에 따라 사이즈 및 재질이 결정된다.

큰 성능 저하 없이 입자 패치를 생성하기 위해선 매우 빠르게 랜덤 이미지를 생성할 수 있는 알고리즘이 필요하다. 또한 입자의 밀도를 조절할 수 있기 위해 0과 1사이의 값을 가지는 실수 값이 생성되어야 한다. 이를 위해 본 논문에선 Seed 값을 받아 빠르게 0과 1사이의 부동소수 값을 랜덤 생성하여 사용한다. 생성된 무작위 값은 원 형태의 커널에 씌워져 그림 (1) 과 같은 패치로 전환된다. 원 형태의 입자가 분포된 패치는 이후 드로잉 툴 종류에 따라 재차 가공되어 렌더링 된다.

입자 패치는 주어진 크기에 따라 랜덤 값 생성을 통해 내부를 채우고 이를 목적에 맞게 필터링 하는 순으로 생성된다.

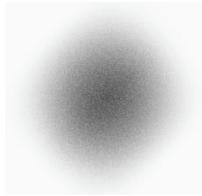


Figure 1. Example image of a single particle patch

3.1 랜덤 값 생성

본 논문에서는 랜덤 값을 생성하기 위해 seed를 입력 받는 Pseudo Random 함수를 이용한다. 입력된 seed를 기반으로 0-1 사이의 실수 값을 생성하는 공식은 아래와 같다.

$$\begin{aligned} & \text{Pseudo Random Value} \\ & = \text{fract}(\sin(\text{dot}(\text{Sedd2D}, \text{vec2}(13.6797, 83.917)))) \\ & * 48732.3637 \end{aligned}$$

위 공식은 동일한 seed 값이 입력되면 동일한 값을 생성하게 된다. 따라서 각 픽셀마다 모두 다른 값을 입력으로 주어야 한다. 패치의 크기에 무관하게 픽셀마다 모두 다른 seed를 주기 위해서 사각형 패치의 텍스처 좌표를 활용한다. 각 패치는 벡터스 4개로 이루어져 있으며 그림 (2)와 같은 텍스처 좌표를 가지고 있다. 텍스처 좌표는 프래그먼트 셰이더에서 선형 보간이 되기 때문에 픽셀 당 모두 다른 값을 가지며 이를 seed로 활용하면 랜덤한 값을 얻어 낼 수 있다.

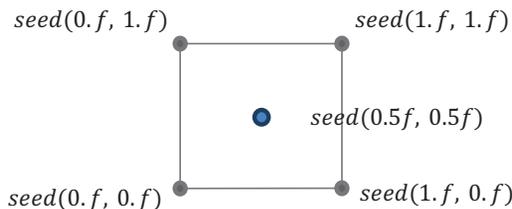


Figure 2. Quad patch and its seeds. The blue dot means the center point of the patch.

얻어진 사각형 랜덤 이미지를 사각형 패치에 매핑하고 필터링하여 본 논문의 최소 단위인 입자 패치를 생성하게 된다.

3.2 필터 및 색 적용

본 논문의 렌더링 최소 단위는 사각형으로 이루어진 하나의 입자 패치이다. 3.1 절에서 생성된 기본 입자 패치에 목적에 따라 각기 다른 필터를 적용한다. 이는 0에서 1사이의 실수 값을 가지는 랜덤 결과값과 다양한 형태의 필터를 적용하여 이루어진다. 먼저 입자 패치의 컬러는 기 선언된 R, G, B 값을 사용한다. 투명도는 각 입자의 색 농도를 결정할 때 사용되며 이에 대한 설명은 4장에서 다룬다. 입자 개수는 연필과 같이 뭉개지는 효과를 표현하기 위해 사용되며 5장에서 다루도록 한다.

패치는 사용자가 그리는 궤적을 그리는 패치의 크기에 비례하는 간격으로 배치된다. 즉, 패치의 크기가 클수록 화면에 배치되는 패치의 개수가 줄어들게 된다. 이러한 방식으로 부드러운 자유 곡선을 표현하기 위해선 중심으로 갈수록 가중치가 커지는 원 형태의 필터 적용이 필요하다. 그림 (2)은 이러한 필터 커널을 보여준다.

이러한 방식은 패치와 패치 사이의 거리가 증가할 경우, 패치의 사이즈가 클 경우, 자연스러운 선의 표현이 어려워지게 되는 단점을 가진다. 본 논문의 입자 방식 표현은 그림 (3)과 같이 이러한 단점을 효과적으로 보완할 수 있다. 기존의 단순 블렌딩 방식이 아닌 실제 입자들로 중첩이 되기 때문이다.

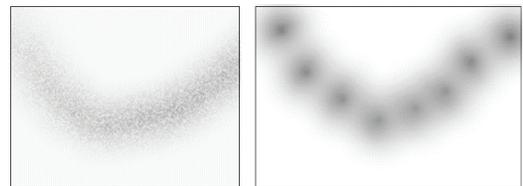


Figure 3. Left image shows the rendering result using our particle patches and right image show the rendering result using traditional patches.

추가로 입자 패치의 개수를 최소화하기 위해 그림 (4)처럼 그려지는 방향의 수직 위아래를 기준으로 짜그러진 원 형태의 필터를 적용한다. 이는 그려지는 중심선 주위에 좀더 많은 입자를 배치하는 효과를 주게 되어 적은 수의 입자 패치로도 끊김 없는 선 표현을 가능하게 해준다.



Figure 4. Directional filtering result

그림 (5)는 선의 방향과 패치에 적용된 필터를 도식화 한 결과이

다. 그러지는 방향에 따라 패치의 필터가 찌그러져 적용 되는 것을 확인 할 수 있다.

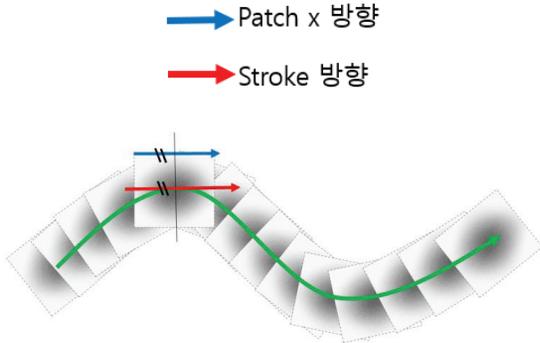


Figure 5. Patches are located perpendicular to the tangent direction of strokes. The direction filtering kernel is applied to the same direction with the tangent direction of strokes.

4. 가상 에어브러시

이번 장에선 3장에서 소개된 입자 패치를 가상 에어브러시에 적용하는 방법을 설명한다.

먼저 실제 에어 브러시가 동작하는 원리를 분석하여 보면, 노즐의 각도, 분사되는 잉크의 양, 캔버스까지의 거리를 기준으로 실제 그려지는 농도 및 두께가 결정되는 것을 알 수 있다. 즉, 그림 (6)에서 표현된 것처럼 노즐의 분사 각, 캔버스와의 거리, 분사 압력 조절을 통해 다양한 형태의 스트로크를 표현할 수 있다. 입자 패치는 크기 및 농도를 조절할 수 있는 특징이 있다. 에어 브러시의 물리적 특성을 기반으로 패치의 크기 및 농도를 계산하여 입자 패치에 적용한다.

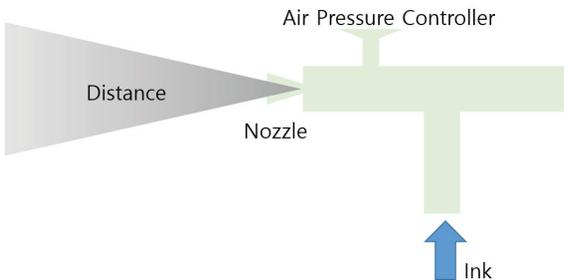


Figure 6. Virtual airbrush concept art. User can control the distance, air pressure, and nozzle angle.

4.1 패치 크기 결정

가상 에어브러시를 위한 패치의 크기는 캔버스와 노즐 사이의 거리에 정비례하도록 생성한다. 그림 (7)은 2차원 디바이스(패드)에 구현하기 위한 그래프이다. 2차원 디바이스의 경우 노즐과 캔버스 간의 거리를 스타일러스 펜의 압력을 기준으로 계산한다. 즉, 펜 압이 높을수록 가까운 거리를 나타내며 낮을수록 먼 거리를 나타내는 방식으로 생성한다. 가상 현실 상에서의 구현은 가상 현실 상에서의 거리를 측정하여 구한다.

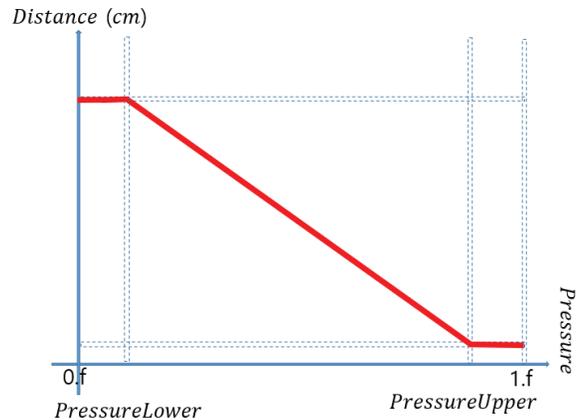


Figure 7. The distance graph

$$Distance = (1 - \min(\max(Pressure, MinPressure), MaxPressure)) * DistanceLimit$$

위 식은 입력 받은 *Pressure*를 기반으로 거리를 구하는 식을 나타낸다.

측정된 거리를 기반으로 실제 패치의 크기를 구하기 위해선 노즐의 분사 각을 계산하여야 한다. 노즐의 분사 각은 0에서 1사이의 실수 값을 입력 받아 분사 각으로 변환하는 식을 사용한다. 그림 (8)은 브러시 크기에 따른 분사 각을 표현하는 그래프이다. 즉, 브러시의 크기가 클수록 분사 각이 크고 작을수록 분사 각이 좁은 방식으로 처리한다.

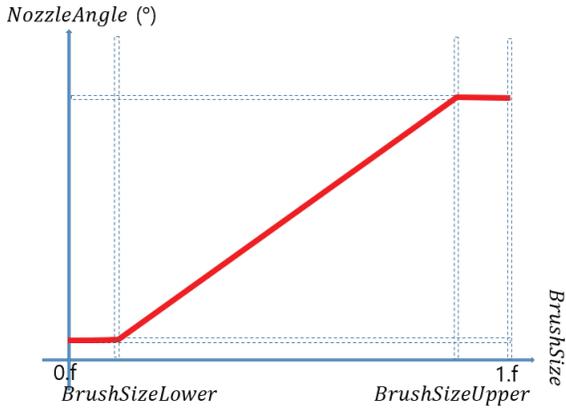


Figure 8. The nozzle angle graph

아래 식은 브러시 사이즈를 입력 받아 노즐의 분사 각을 계산하는 식이다.

$$\begin{aligned} \text{NozzleAngle} \\ &= \min(\max(\text{BrushSize}, \text{BrushSizeLower}), \text{BrushSizeUpper}) \\ &\quad * \text{NozzleAngleLimit} \end{aligned}$$

다음으로 앞서 구한 거리와 노즐 각도를 이용하여 입자 패치의 면적을 결정한다. 패치의 크기는 아래 식을 기반으로 계산되며 계산된 크기를 기준으로 패치의 폭과 높이를 계산한다.

$$\text{SprayedRegion} = \pi \left(\text{Distance} * \tan\left(\frac{\text{NozzleAngle}}{2}\right) \right)^2$$

패치가 정사각형일 경우 양의 제곱근을 구하여 사용한다. 방향성이 큰 패치일 경우 스트로크 방향 축에 가중치를 둔 직사각형 형태로 패치를 구성하고 폭을 계산한 후 면적에서 나누어 높이를 계산한다. 그림 (9)는 거리에 따른 패치의 면적을 나타내는 그래프이다. 에어 브러시와 렌더링 되는 캔버스 사이의 거리가 멀수록 급격하게 범위가 늘어나는 것을 확인할 수 있다.

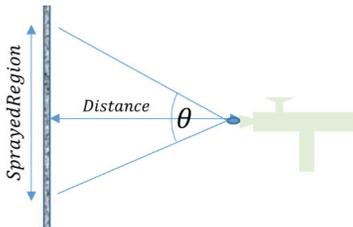


Figure 9. The size of sprayed region according to the distance

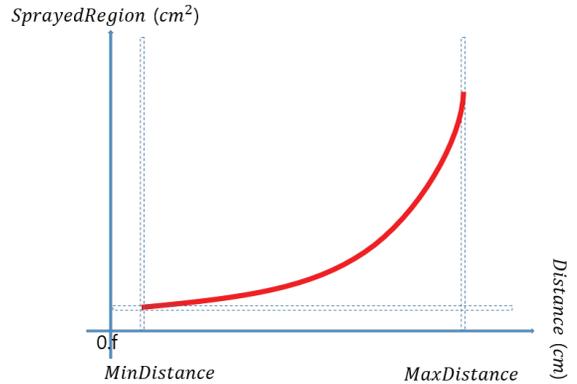


Figure 10. The sprayed region grows exponentially while distance from the nozzle to the canvas.

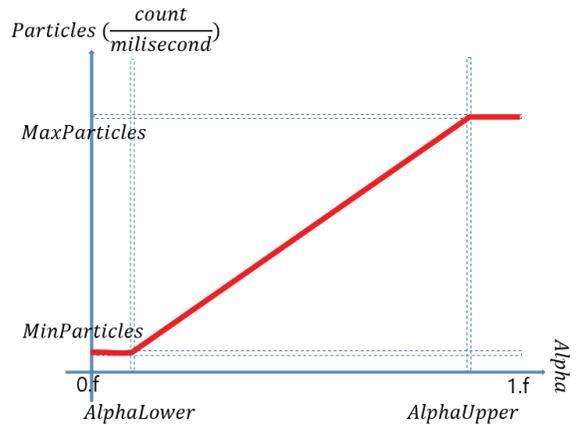


Figure 11. The number of particles according to the amount of particle emit.

4.2 패치 농도 결정

에어브러시는 미세한 잉크가 입자 형태로 분사되는 특징을 가지고 있다. 즉, 캔버스의 특정 위치에 머문 시간에 비례하여 색 농도가 짙어 지는 특징을 가진다. 가상의 에어브러시에서 이와 같은 특성을 나타내기 위해 머문 시간에 정비례하여 입자 패치의 농도 높여준다.

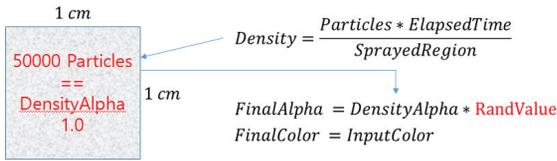


Figure 12. The calculation of particle patch alpha

그림 (12)는 입자 패치의 농도를 구하는 식을 나타낸다. 먼지 펜의 속도를 측정하여 해당 영역에 머문 시간을 초단위로 구한다 (ElapsedTime). 초당 배출되는 입자의 개수는 Particles 이며 실제 뿌려 지는 입자의 개수는 ElapsedTime 과 Particles 를 곱하여 구할 수 있다. 이 값을 4.1 절에서 구한 패치의 면적으로 나누면 단위 면적당 뿌려지는 입자의 개수(Density)를 구할 수 있다. 이렇게 도출된 Density를 최대 Density(50000 개)로 나눈 후 입자 패치의 랜덤 값 알파에 곱하여 최종 입자 패치를 완성한다. 즉, 입자의 농도는 컬러의 투명도를 사용하여 표현하는 방식이다. 그림 (13)은 농도에 따른 패치의 변화를 나타낸다.

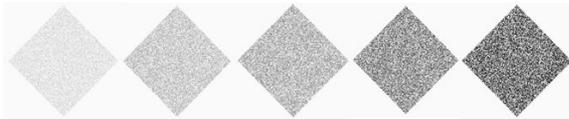


Figure 13. The series of rendering results while increasing the density of the virtual airbrush

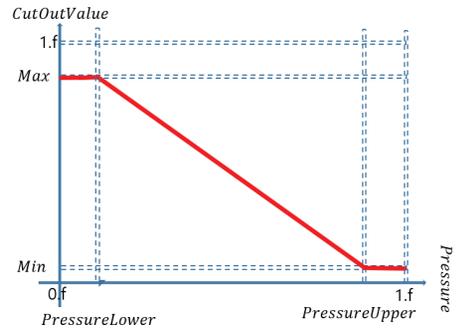
5. 가상 연필

가상 연필은 캔버스의 거친 표면과 흑연으로 이루어진 연필 심과의 마찰로 인해 흑연이 뭉개 지며 그려지는 특성이 있다. 따라서 4장의 에어브러시와는 다르게 캔버스와의 거리에 따른 계산이 필요 없다. 즉, 브러시 크기는 연필 심의 두께에만 영향을 받기 때문에 한번 결정된 패치 크기로 계속 그려지게 된다. 다만 그리는 압력에 따라 뭉개지는 양이 다르며 이에 따른 농도 차이는 계산하여야 한다.

5.1 패치 농도 결정

가상 연필의 농도는 입자 패치의 랜덤 값의 크기를 기반으로 한다. 이때 입자 패치의 각 픽셀은 0에서 1사이의 실수 값을 가지는 특징을 활용한다. 2차원의 경우 스타일러스 펜의 압력에 반비례하는 값을 기준으로 한다(그림(14) CutOutValue). 즉, 압력

이 셀 수록 0에 가까우며 이 값보다 큰 값을 가지는 입자만 렌더링 되는 방식이다.



INPUT: Pressure 0.f~1.f

Figure 14. Pressure-density mapping graph

그림 (15)는 CutOutValue에 따른 렌더링 결과를 나타낸다. 이 그림의 붉은색 수직선을 기준으로 절단면을 도식화해보면 그림 (16)와 같은 형태를 띤다. CutOutValue보다 작은 값을 가지는 영역을 렌더링 과정에서 무시하면 그려지게 되는 입자의 수와 크기가 줄어드는 것을 확인 할 수 있다. 가상 에어 브러시의 경우 투명도로 농도가 결정되는 반면 가상 연필은 실제 입자의 밀도에 따라 농도가 정해지는 차이를 확인할 수 있다.



Figure 15. The series of rendering results while increasing the density of the virtual pencil

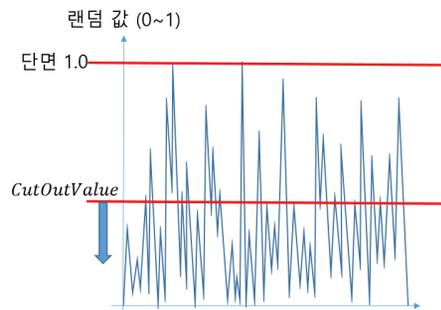


Figure 16. The section of the red line in Figure 15.

5.2 흑연 머티리얼

연필은 흑연이 뭉개지면서 그려지는 원리이다. 특히 흑연의 재질은 강하게 정반사 시키는 특성이 있다. 실제 연필로 그려진 그림을 시점을 다변화 시키며 관측 시 정반사되는 입자로 인해 반짝이는 특성을 보이는 것을 알 수 있다.

이러한 특성을 표현하기 위해 입자 단위로 랜덤 한 방향의 법선 벡터를 가지도록 한다. 정반사 모델을 표현하기 위해선 음영 모델을 적용해야 하는데 수많은 패치가 그려지는 것을 고려하여 가장 단순한 형태의 모델인 Blinn 조명 모델을 사용한다.

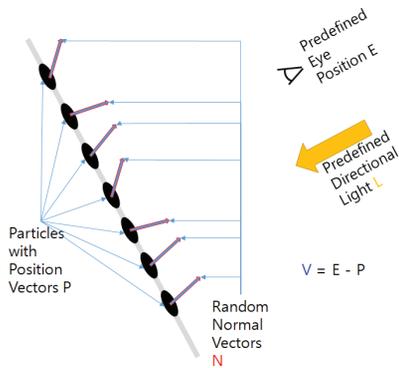


Figure 17. Normal vector random generation

그림 (17)은 시점과 광원, 입자의 랜덤 법선 벡터를 도식화 한 결과이다. 실제 흑연은 캔버스 표면의 미세한 틈에 의한 마찰력으로 랜덤 한 형태의 조각으로 부서지며 그려지게 된다. 이러한 특징은 랜덤 한 방향의 법선 벡터를 생성하는 원리에 부합된다.

6. 결과

본 논문의 구현은 2 차원 디바이스 및 가상 현실 디바이스 양측에서 이루어 졌다. 2 차원 디바이스는 안드로이드 기반의 스마트폰을 이용하였고 OpenGL 라이브러리를 기반으로 구현이 되었다. 이는 구글 플레이 스토어에서 무료로 다운받아 설치할 수 있다 (펜소울). 가상 현실상에서 구현은 게임 엔진을 활용하였다. 사용된 게임 엔진은 에픽 사의 언리얼 엔진 4 이다. 가상 현실 데모를 위한 HMD(Head mounted display)는 삼성 오디세이를 활용하였다

먼저 2D 환경에서의 렌더링 결과를 래스터 기반의 알고리즘과 비교하였다. 그림 18 는 본 논문의 가상 에어 브러시 렌더링 결과이다. 아래 확대된 두 이미지 중 왼쪽 이미지는 기존 방식을 기반으로 확대 후 렌더링하였을 때의 결과이며

오른쪽은 제안된 방식으로 확대 후 렌더링 하였을 때의 결과이다. 왼쪽 이미지의 경우 명확하게 사각형 형태의 앨리어싱이 생기는 것을 확인할 수 있다. 본 논문의 결과 이미지는 이러한 앨리어싱이 전혀 나타나지 않음을 확인할 수 있다.

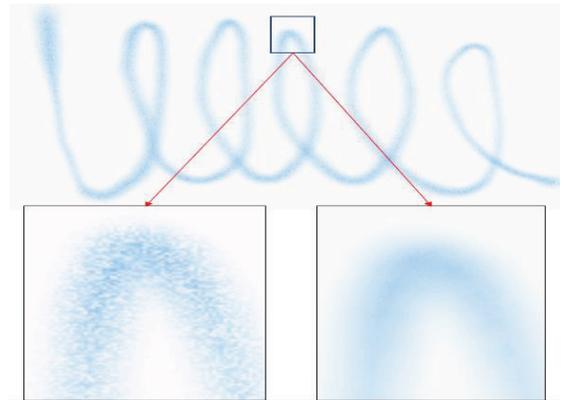


Figure 18. Rendering result comparison. The bottom left image shows the rendering result of traditional method. The bottom right image shows the rendering result of our method.

그림 19 및 그림 20은 가상 연필 및 에어 브러시를 가상 현실 공간에 구현한 결과이다. 이차원 디바이스를 사용했을 경우엔 거리 및 압력을 스타일러스 펜의 출력 값을 사용하지만 가상 현실 공간에선 가상 현실상의 거리 및 침습 깊이를 기준으로 자동으로 계산된다. 가상 에어 브러시 및 가상 연필은 가상 현실 기기의 모션 컨트롤러의 좌표 입력을 받아 배치시키며 모션 컨트롤러의 스틱의 움직인 양에 따라 잉크의 농도 및 밀도를 결정하게 된다. 각 렌더링 결과의 왼쪽은 비교적 원거리에서 캔버스를 보았을 때의 이미지이며 오른쪽은 근접하여 관찰하였을 때의 결과이다. 제안된 알고리즘은 거리에 상관없이 깨끗한 렌더링 결과를 나타내는 것을 확인할 수 있다.



Figure 21. Drawing on the virtual canvas using our virtual airbrush rendering algorithm.

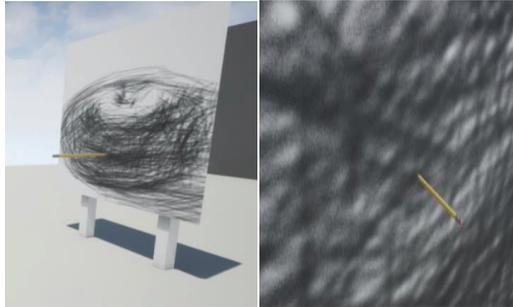


Figure 22. Drawing on the virtual canvas using our virtual pencil rendering algorithm.

7. 결론

본 논문에선 가상 에어 브러시 및 연필 렌더링 알고리즘을 제안하였다. 제안된 알고리즘은 확대 및 축소에 강건한 결과를 나타냈으며 이는 가상 현실과 같이 오브젝트의 확대 및 축소가 빈번하게 일어나는 환경에 적합한 렌더링 알고리즘이라 할 수 있다. 입자 패치는 가상의 크레용, 붓, 볼펜 렌더링 알고리즘에도 적용이 가능하다. 향후 추가적인 브러시에 대한 연구를 진행한다면 더욱 다양한 가상 드로잉 툴 알고리즘이 구현될 것으로 기대된다.

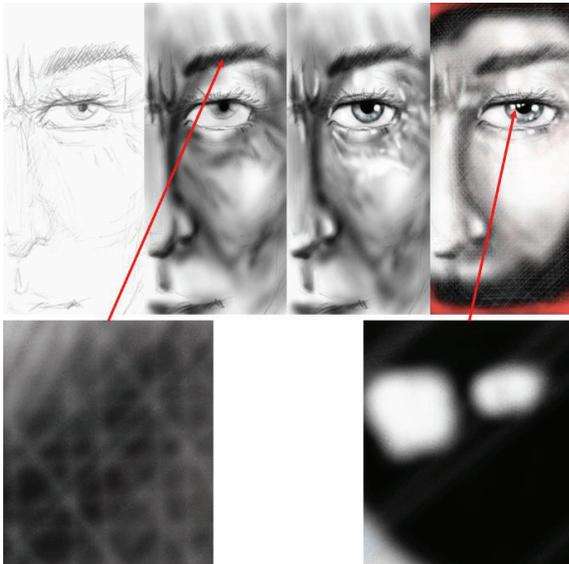


Figure 23. Sketch example using our method. Bottom images show the magnification result of the tiny part of top images. No aliasing occurs even we scale the image up 100 times.

감사의 글

본 연구는 (주)에프엑스기어 및 한국연구재단 신진연구사업 (NRF-2017R1C1B200730)의 지원으로 이루어졌습니다.

References

- [1] A. Quint, "Scalable vector graphics," in *IEEE MultiMedia*, vol. 10, no. 3, pp. 99-102, July-Sept. 2003
- [2] Kilgard, Mark J. and Bolz, Jeff, "GPU-accelerated Path Rendering", *ACM Transactions on Graphics (TOG)*, Volume 31 Issue 6, November 2012
- [3] Helena T.F Wong, Horace H.S Ip, "Virtual brush: a model-based synthesis of Chinese calligraphy", *Computers & Graphics*, Volume 24, Issue 1, 2000, Pages 99-113,
- [4] S. DiVerdi, "A Modular Framework for Digital Painting," in *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 7, pp. 783-793, July 1 2015.
- [5] Ming Lin, William Baxter, Vincent Scheib, and Jeremy Wendt. "Physically based virtual painting.", *Commun. ACM* 47, 8 (August 2004), 40-47
- [6] 최성욱, "간단하고 빠른 실시간 연필 스트로크 알고리즘", *정보과학회논문지: 시스템 및 이론*, 제33권, 제6호, pp.344-353, 2006.
- [7] 김대석, 박진아, "레이와 텍스처 기법을 이용한 실시간 스프레이 페인팅", *정보과학회논문지: 컴퓨터의 실제*, 제14권, 제8호, 2008.
- [8] 안지혜, 박진호, "캘리그래피에 특화된 래스터 이미지 확대 방법", *한국컴퓨터그래픽스학회 논문지*, 제21권, 제4호, pp.1-10, 2015.
- [9] 장호연, 류승택, 박진완, "로봇을 이용한 벽화 시스템", *한국콘텐츠학회논문지*, 9(12), pp.592-599, 2009.
- [10] Alexander V. ReshetovYevgeniy P. KuzminDenis V. IvanovAlexander N. Yakovlev. , "Spatial patches for graphics rendering", *US Patent US7102636B2*, March 31, 2001

〈저자소개〉



이혜린

- 2015년 3월 ~ 현재 한국산업기술대학교
게임공학전공 재학 중



오건

- 2013년 3월 ~ 현재 한국산업기술대학교
게임공학전공 재학 중



이택희

- 2001년 서울대학교 컴퓨터공학부 학사
- 2009년 서울대학교 컴퓨터공학부 박사
- 2009년-2012년 (주)삼성전자 무선사업부
- 2017년-현재 한국산업기술대학교
게임공학부 조교수