

레이와 텍스처 기법을 이용한 실시간 스프레이 페인팅

(Real-time Spray Painting using Rays and Texture Map)

김 대 석 * 박 진 아 **
(Daeseok Kim) (Jinah Park)

요 약 본 논문에서는 가상환경에서 페인트를 분사하여 실시간으로 물체를 도색 하는 시뮬레이션을 위한 충돌처리 및 시각화 알고리즘을 제시한다. 이를 통하여 물체에 페인트가 뿌러지면서 도색 되는 모습을 사실적으로 표현해 줄 뿐만 아니라, 페인트 누적 모델을 이용하여 물체에 누적된 페인트의 두께 정보까지 시뮬레이션 하여 시각화함으로써 가상훈련 시스템에 적용할 수 있도록 한다. 분사되는 유체 시뮬레이션을 위해서 기존에는 파티클 시스템이 이용되고 있으나 실시간으로 도색이 되는 과정을 시각화하기 위해서는 수백만 개의 파티클에 대하여 충돌 검사를 수행해야 하기 때문에 적절하지 않다. 따라서 본 연구에서는 소수의 레이와 텍스처 기법을 이용하여 효율적으로 충돌 검사를 수행하는 알고리즘을 제안하고 이를 구현하였으며, 실시간 페인트 시뮬레이션 구현 결과와 수행 시간 분석을 통하여 알고리즘의 효율성을 검증하였다.

키워드 : 시뮬레이션, 누적모델, 가상훈련, 시각화, 충돌 검사, 스프레이 페인팅

Abstract The purpose of this study is to develop realistic painting simulation in real-time as well as to represent the thickness of the deposited paint on the

* 본 연구는 2006년도 한국전자통신연구원 가상현실융합 협력과제로 수행하였음

* 이 논문은 2007학년도 홍익대학교 학술연구진흥비에 의하여 지원되었음

* 이 논문은 2007 한국컴퓨터종합학술대회에서 '레이와 텍스처 기법을 이용한 실시간 스프레이 페인팅'의 제목으로 발표된 논문을 확장한 것임

† 학생회원 : 한국정보통신대학교 공학부

greatstone@icu.ac.kr

** 정 회 원 : 한국정보통신대학교 공학부 교수

jinah@icu.ac.kr

논문접수 : 2007년 9월 27일

심사완료 : 2008년 10월 20일

Copyright©2008 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 : 컴퓨팅의 실제 및 레터 제14권 제8호(2008.11)

surface. The Gaussian model is used for a painting deposition model to calculate the thickness of paints. For a painting simulation, rather than implementing particle systems, we propose a new heuristic algorithm for painting process based on a few number of rays. After we find the collision points of the rays with an environment, we compute the painted area using flood-fill searching method on the texture map and visualize paint effects. We analyzed time complexity of our method to verify that our system is suitable for real-time VR applications.

Key words : simulation, deposition model, virtual training, visualization, collision detection, spray painting

1. 서 론

본 논문에서는 유체가 일정한 힘을 받고 분사되어 이동하다 다른 물체에 부딪혔을 때, 그 물체의 표면에 입혀져 색을 변화시키는 경우에 대한 시뮬레이션을 다룬다. 이런 현상을 적용해 볼 수 있는 구체적인 상황은 스프레이를 이용해 벽면에 페인트를 칠하는 경우이다. 실제로 선박을 제조하는 산업 현장에서 제품의 벽면을 코팅할 때 스프레이 분사 도구를 이용해 도장한다. 선박 도장 작업은 전체 제조 공정 중 30%의 비중을 차지하는 중요한 과정인데, 페인트 원료의 높은 가격 때문에 초보 작업자들의 훈련이 원활히 이루어지지 않고 있다. 따라서 가상 페인트 도장 훈련을 위한 시뮬레이션을 구상하면서 실시간 스프레이 페인팅 시스템에 대한 연구를 진행하게 되었다.

유체가 공기 중에서 흩날리는 현상을 시뮬레이션 하는 방법 중 하나는 입자들을 개체로 생성한 뒤, 운동 모델을 정의하여 움직이도록 만드는 파티클 시스템을 사용하는 것이다[1,2]. 이 방법은 많은 입자를 가지고 있는 유체를 직접적으로 모델링하기 때문에 직관적이며, 유체의 움직임을 사실적으로 표현할 수 있다는 장점이 있다. 페인팅 시뮬레이션에도 파티클 시스템을 적용해 볼 수 있다[3]. 스프레이의 분사 구에서 다수의 파티클 입자들이 발사되어 날아갈 때, 파티클 입자가 물체의 표면에 부딪히게 되면 그 부분에 파티클 폴리곤을 위치시키거나 텍스처 맵을 수정함으로써 표면이 도색 되는 효과의 표현이 가능하다. 하지만 파티클의 수가 증가하면 충돌 검사를 수행하면서 시스템의 속도가 느려지기 때문에 파티클 시스템으로는 세밀한 페인팅 효과를 실시간으로 표현하기가 어렵다. 따라서 본 논문에서는 스프레이 분사를 통한 페인트 도색을 시뮬레이션 하기 위한 새로운 알고리즘을 제안한다. 페인트 입자를 몇 개로 샘플링 하여 이것을 이용해 물체와의 충돌 검사를 한 뒤, 페인트가 칠해질 부분을 플러드-필(Flood-Fill) 알고리즘을 적용해 영역을 찾아내는 방법이다. 이 방법을 이용하면,

파티클 시스템보다 빠른 속도로 시물레이션을 수행할 수 있다. 또한, 페인트 누적 모델을 이용해 페인트가 칠해진 부분의 두께 변화의 계산이 가능하다.

2. 관련 연구

Hanrahan[4]은 3D 환경에서 물체를 둘러보며 표면에 색칠을 할 수 있는 시스템을 개발하였는데 이는 물체 위에 텍스처를 입히고, 사용자의 브러쉬가 텍스처 위의 텍셀에 접근 하여 색을 변경하는 방법이다. 텍셀에 접근하는 방법으로 직접 텍스처 맵에 접근하는 Parameter-space brush와 스크린에 보이는 부분에 접근하는 Screen-space brush, 막대기 형태의 브러쉬가 가리키는 표면을 접근하는 Tangent-space brush를 제안하였다. 최근에는 직접적인 인터페이스를 이용해 3D 페인팅을 하는 연구가 발표되었다[5]. 위 연구들은 3D 환경에서 물체에 페인팅을 하는 방법을 제시하고 있다. 하지만 페인팅을 위한 인터페이스로는 모두 브러쉬를 사용하고 있다. 브러쉬는 직접적으로 물체에 닿아, 그 위치 주변을 색칠하는 도구이다. 페인팅을 하는 인터페이스로 스프레이를 사용하는 경우를 생각해 보면, 일정 부위가 아니라 광범위한 범위에 색칠을 하게 되기 때문에 다른 접근 방법이 필요하다.

3. 페인트 누적 모델

페인트 입자들이 분사될 때, 이들의 움직임에 대한 모델이 정의되어야 한다. 예를 들어 페인트가 한 지점 $P(x, y, z)$ 에서 분사되었을 때, $Q(x', y', z')$ 에 페인트 입자들이 도달할 확률이나 이 곳에 물체가 존재하여 표면에 달라붙었을 경우의 페인트의 두께를 계산하기 위한 모델이 필요하다. 이 모델을 페인트 누적 모델이라 정의하겠다. 페인트 누적 모델은 페인팅 로봇의 경로를 결정하는 논문에서 이미 연구된 바 있다. Freund[6]는 Gaussian model을 이용해 오차가 적은 페인트 누적 모델을 구축하였다. 이를 참고해 3D 환경에서 분사 각도와 거리에 따라서 표면에 누적되는 페인트 양을 측정할 수 있는 누적 모델을 다음과 같이 가우시안에 근거한 볼륨모델을 작성하였다.

$$f(x, y, z) = \frac{c}{z^2} \exp(-r^s(x, y, z))$$

$$r(x, y, z) = \sqrt{\left(\frac{x}{az}\right)^2 + \left(\frac{y}{bz}\right)^2}$$

$f(x, y, z)$ 는 분사구가 정점에 위치하고 z 축을 향해 페인트를 분사하였을 때, 계산되는 페인트 누적 모델이다. 변수 a 와 b 는 페인트가 분사될 때, 스프레이의 퍼짐을 결정한다. 변수 s 는 페인트 도막 두께의 평평함을 나타낸다. s 가 작을수록 가운데가 볼록한 형태가 되며, s

가 커지면 평탄한 누적 모델이 된다. c 는 전체의 유량을 결정하는 파라미터이다. 페인트 도막의 두께는 페인트의 성질과 이를 분사한 스프레이의 노즐 종류에 따라 다르다. 본 연구에서는 산업현장에서 실제로 사용되고 있는 페인트 두 종류와 대표적인 원형 노즐 두 종류, 네 가지 케이스로 실험하여 도막두께를 측정하였다. 이 측정 데이터를 이용하여 가우시안에 근접한 볼륨모델의 파라미터들 (a, b, c, s)을 추정하였다.

4. 도색 영역 찾기 알고리즘

페인트가 어떤 위치에서 분사된다면, 정면에 위치한 폴리곤들 중 페인트가 닿는 부분은 도색이 될 것이다. 이 때, 어느 부분이 페인트에 닿는가 닿지 않는가를 판단해야 한다. 페인트가 칠해져야 할 부분을 도색 영역이라 정의한다. 도색 영역의 범위는 페인트를 분사하는 도구나 페인트의 속성 등의 영향을 받아 변한다. 도색 영역을 찾기 위해 페인트 누적 모델을 이용하는 휴리스틱 알고리즘을 적용 하였다. 그림 1은 알고리즘의 흐름도이다. 첫째, 페인트 입자를 파티클 대신 레이(ray)로 대체하여, 이를 이용해 물체들과 충돌 검사를 수행한다. 둘째, 충돌 검사를 통해 찾은 점들에서 시작하여 주변 점들을 확장 탐색하여 한 면에서의 도색 영역을 텍셀 단위로 찾는다. 셋째, 도색 영역이 한 면을 넘어가게 되는 경우, 새로운 레이를 생성하여 이웃한 면을 찾는다. 넷째, 탐색된 도색 영역 중 다른 면에 가려져 도색이 되지 않아야 하는 부분을 제거 한다. 도색 영역을 모두 찾은 이후에 페인트 효과를 적용한다. 본 연구에서는 한 면의 기본 구조를 triangle이 아닌 plane 단위로 설정하였다. 텍스처를 적용하거나 탐색을 할 때, 한 단위로 계산하기 위해 연결된 평행한 triangle들을 한 집단으로 묶어 plane이라 정의하였다.

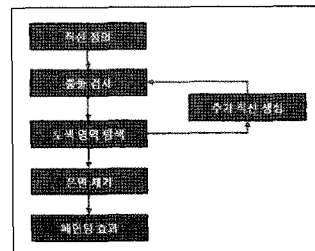


그림 1 알고리즘 흐름도

4.1 레이(Ray) 정의

폴리곤의 수가 증가 할수록 충돌 검사의 계산 시간도 증가한다. 따라서, 페인트가 분사되는 영역을 세밀히 찾기 위해 많은 파티클 입자들을 생성하면 수행 속도가 크게 느려지게 된다. 그렇기 때문에 오히려 충돌 검사를 위

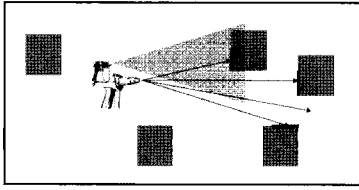


그림 2 페인트 분사 영역과 정의된 레이들

한 파티클의 수를 크게 줄이고, 다른 방법으로 분사 영역을 찾는 알고리즘을 구상하였다. 본 연구에서는 페인트 파티클이 일직선으로 이동한다고 가정하고, 분사 각 내의 레이들 인의로 선택하였다(그림 2). 이 레이들과 폴리곤과의 충돌 검사를 수행하면 몇 개의 점들을 찾아 낼 수 있다. 왼쪽 형으로 분사 되는 페인트 영역의 중심을 향하는 레이 하나와 위쪽, 아래쪽, 왼쪽, 오른쪽 가장자리 부분의 레이들 각각 하나씩 총 다섯 개를 선택하였다.

4.2 도색 영역 탐색

레이들과 물체들의 충돌 검사를 통해 찾은 점들로부터 그 면의 도색 영역을 탐색하기 시작한다. Flood-Fill 알고리즘을 이용해 충돌 지점을 중심으로 그 주변의 점들을 나선 방향으로 그림 3과 같이 검색해 간다. 이 탐색 방법은 주변의 점들을 큐에 넣고, 하나씩 꺼내가며 유효한 도색 영역인지를 판별하는 방법이다. 이 때, 페인트 효과가 적용되는 부분은 텍스처 맵의 한 점이 되므로 충돌 검사를 통해 발견한 기하학 좌표계 위의 점을 텍스처 좌표 계로 변환한 뒤, 텍셀 단위로 탐색을 한다. 한 점이 유효한 도색 영역인지를 판별하기 위해서, 페인트 누적 모델을 이용해 그 위치의 페인트 두께를 계산하고 그 결과가 일정 값 이상인 경우 유효하다고 판단하였다.



그림 3 Flood-Fill Algorithm

4.3 이웃 면 확장 탐색

충돌 검사를 통해 발견한 면에서 도색 영역 탐색을 진행할 때, 도색 영역이 가장자리에 닿게 되는 경우가 있다. 이런 경우 페인트는 그 면을 넘어 바깥 부분까지 칠해질 것이다. 따라서, 이 조건에서는 바깥쪽으로 도색 영역 탐색을 확장하는 방법을 적용한다. 도색 영역 탐색 작업이 면의 가장자리에 닿게 되었을 때, 그 면의 바깥쪽을 향하는 새로운 레이들 생성하여 환경의 물체들과 충돌 검사를 수행한 후, 새로 발견한 면들에 대해 도색 영역 탐색을 진행한다. 그림 4의 왼쪽 그림과 같이 페인트가 분사되는 영역의 안쪽에 다른 면들이 존재하지만 처음에 정의된 레이에 닿지 않아 도색 영역 탐색 작업을 수행하지 않던 면들이 이 과정에서 발견되고, 새로 발견된 점으로부터 도색 영역 탐색을 하여 페인트가 칠

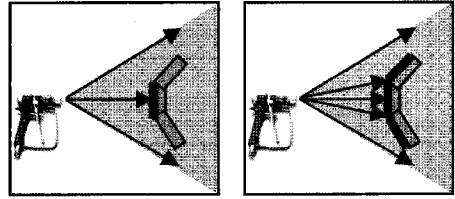


그림 4 이웃 면을 찾기 위한 새로운 레이 생성

해져야 할 부분들을 확장할 수 있다. 면의 가장자리에 도색 영역 탐색이 다다르게 되면 그림 4의 오른쪽 그림과 같이 면의 모서리 양끝의 두 점을 향하는 레이들 각각 생성한다. 즉, 두 점에서 면의 바깥쪽으로 조금 떨어진 곳을 향하는 벡터를 생성함으로써, 충돌 검사를 수행했을 때 새로운 면을 찾을 수 있도록 한다.

4.4 은면 제거

이웃 면 확장 탐색을 모두 마치고 나면 은면 제거를 수행한다. 도색 영역 탐색은 페인트가 칠해진다고 판단된 면 위에서 유효성만을 판단하며 도색 영역을 찾아가기 때문에 새로 찾은 영역이 다른 면에 의해 가려졌는지 여부를 판별할 수 없다. 따라서 다른 면에 가려져 페인트가 칠해지지 않아야 하는 부분임에도 불구하고 도색 영역으로 판별된 점들이 상당 수 존재하게 된다. 이 문제를 해결하기 위하여 다른 면에 의해 가려진 영역을 찾아내어 제거하는 작업을 수행한다. 도색 영역으로 판별된 점들 중 다른 면에 의해 가려져 있는 점들은 면을 확장할 때 생성한 두 개의 벡터를 이용해 알 수 있다. 이 두 벡터가 다른 면과 만났을 때, 두 점을 연결한 선분은 앞의 면에 의해 가려진 경계선을 나타낸다. 페인트가 칠해져야 할 영역이 이 경계선의 위쪽이라고 한다면, 아래 쪽의 점들을 유효 영역 집단에서 제거하는 것이다. 예를 들어 그림 5의 같은 경우, 한 면의 모서리가 다른 면과 만나고 있을 때 탐색을 통해 두 면의 도색 영역을 찾게 될 것이다. 이 때, 점선 부분은 다른 면에 의해 가려져 있지만 탐색 과정에서 유효 영역으로 판별하게 된다. 이것을 확장 탐색 중 생성된 가장자리를 향하는 레이들 이용해 도색 영역의 경계선으로 결정하고, 다른 면에 의해 가려진 영역을 제거한다.

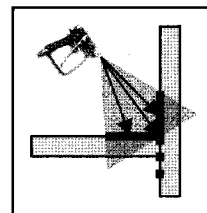


그림 5 은면 제거를 위해 레이들 이용

5. 페인팅 방법

도색 영역을 찾고 나면 텍스처 맵을 수정함으로써 페인트 효과를 준다. 페인트가 누적되는 효과를 주기 위해 텍스처 맵과 같은 크기의 페인트 두께 맵을 생성해두고, 페인트 누적 모델을 통해 계산된 두께 값을 두께 맵에 가산하면서 두께 맵의 값을 이용해 텍스처 맵의 색을 열고 진하게 조절함으로써 사실적인 페인트 효과가 가능하다. 특히 시각적인 효과를 극대화하기 위해 페인트가 분사될 때 파티클이 날려서 주변에 붙는 모습을 노이즈를 생성함으로써 표현이 가능하다.

6. 구현 결과

펜티엄 3-3.4 GHz, 3.12GB RAM, NVIDIA GeForce 6600, Windows XP의 Visual Studio, NET환경에서 OpenSG 라이브러리를 사용하였다.

6.1 3D 모델

페인팅 시뮬레이션을 위해 실제 모습, 실제 크기와 동일하게 모델링 한 선박 모델을 사용하였다. 텍스처 해상도는 모델의 1cm를 1pixel로 매칭하였다. 그림 6은 본 시스템에서 사용한 선박 모델과 분사 도구 모델이다.

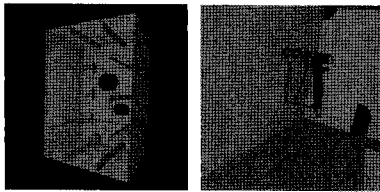


그림 6 선박 모델과 분사 도구 모델

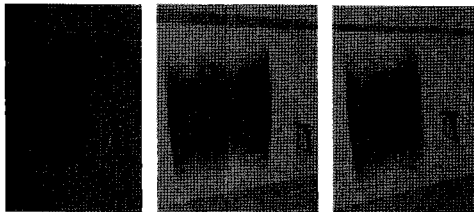


그림 7 페인트 효과 방법

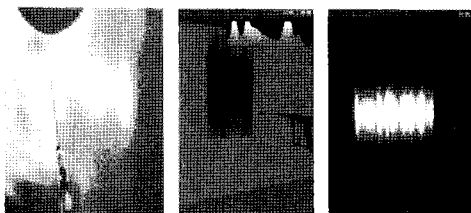


그림 8 페인트 두께 시각화 방법

6.2 페인트 효과

페인트가 칠해진 유무만을 시각화, 페인트가 누적되는 것을 계산해 농도를 색을 통해 시각화, 임의의 노이즈를 생성하여 시각화하는 등 다양한 방법으로 페인트 효과를 표현할 수 있다(그림 7).

6.3 두께 시각화

페인트 누적 모델을 이용해 페인트 효과를 시각화 할 때, 페인트 두께도 함께 계산된다. 하지만 페인트 효과만을 시각화 했을 때는 일정 이상 페인트를 계속 뿌려도 두께가 증가하는 것을 알 수가 없기 때문에 다른 방법으로 페인트의 두께를 시각화 할 필요가 있다. 따라서, 사용자가 원하는 한 점의 두께 정보를 텍스트로 나타내 주거나, 두 점을 선택했을 때 그 사이의 두께 변화를 그래프로 보여주거나, 페인트가 칠해진 면 자체에 두께의 수치에 따라 색을 다르게 표현하는 방법으로 도색된 페인트의 두께를 시각화 하였다(그림 8).

7. 수행 시간 분석

스프레이 페인팅 알고리즘의 목적은 도색 영역 탐색과 페인트 효과 적용이 실시간으로 이루어지는 것이다.

수행 시간 복잡도를 분석해 보면, 초기 정의된 레이의 수를 R, 탐색 과정 중 생성되는 레이의 수를 P, 충돌 검사를 하는 시간을 C, 전체 도색이 되는 텍셀의 수를 T라고 했을 때, 다음과 같다.

$$(R+P)C + 2T$$

R+P는 도색 되는 폴리곤의 수와 비례하고, C는 전체 환경의 폴리곤 수와 비례한다. 즉, 본 연구의 알고리즘은 도색 되는 폴리곤의 수와 전체 환경의 폴리곤의 수, 그리고 도색 되는 텍셀의 수에 의해 수행 시간이 변한다.

비교를 위해 각 파티클들이 충돌 검사를 수행한 뒤, 충돌하는 지점에 도색을 하는 시스템의 시간 복잡도를 분석한다. 이 시스템은 N개의 파티클이 생성될 때, NC의 시간 복잡도로 수행하게 될 것이다. 본 연구에서 초점을 맞춘 부분은 충돌 검사를 수행 하는데 많은 시간이 소모된다는 점이다. 시간 복잡도를 비교해 봤을 때, 파티클 시스템은 파티클의 수만큼 충돌 검사를 수행하고, 스프레이 페인팅 알고리즘은 페인트 되는 폴리곤의 수만큼 충돌 검사를 수행한다.

이와 같은 요소가 실제 수행 속도에 어떻게 영향을 미치는 지 파악하기 위해 파티클 시스템과 본 연구의 알고리즘을 구현하여 수행시간을 측정하였다. 수행 시간은 전체 2272개의 폴리곤 환경에서 측정하였다. 표 1은 파티클 시스템을 이용한 페인트 시뮬레이션의 수행 결과이다. 파티클의 수가 증가할수록 충돌 검사의 횟수가 증가하기 때문에, 초당 프레임 수가 급격히 감소한다.

파티클의 수의 변화는 페인트의 해상도라고 할 수 있

표 1 파티클 시스템의 파티클 수 변화에 따른 속도 변화

도색 되는 폴리곤 수	프레임 수 (텍셀 수: 500)	파티클 수	프레임 수
1	250	1	1000
20	166	100	100
100	52	500	19
200	31	1000	10
300	21	1500	7

표 2 텍셀 수의 변화에 따른 속도 변화

도색 되는 텍셀 수	프레임 수	
	폴리곤 수: 20	폴리곤 수: 100
100	250	58
300	166	52
1000	100	45
1500	71	40

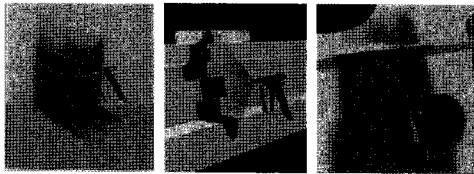


그림 9 구현 결과 화면

다. 파티클 하나가 부딪힌 곳에 일정 크기의 도장 효과를 표현한다고 할 때, 파티클의 수에 따라 실제 페인트 효과와 비슷해지기 때문이다. 스프레이 페인팅 알고리즘에서는 도색 영역의 텍셀의 해상도가 이와 관련된다. 표 2는 텍셀의 해상도 변화에 따른 스프레이 페인팅 알고리즘의 수행 속도의 결과이다. 역시 텍셀의 수의 증가에 따라 초당 프레임 수가 감소하지만 파티클 시스템에 비해 그 차이가 적다. 하지만 스프레이 페인팅 알고리즘은 도색 면의 폴리곤의 수에 따라서도 속도가 변하는 것을 볼 수가 있다. 표 3은 도색 면의 폴리곤 수의 변화에 따른 수행 속도의 결과이다. 은 파티클 시스템에서는 도색 되는 면이 하나의 폴리곤이거나 수 많은 폴리곤으로 구성된 곡면이거나 관계가 없지만, 스프레이 페인팅 알고리즘은 그 차이에 따라 초당 프레임 수가 급격하게 감소한다. 이 결과를 통해 스프레이 페인팅 알고리즘은 곡면으로 구성된 복잡한 폴리곤 환경에 적합하지 않다는 것을 알 수가 있다. 하지만 페인트 되는 영역의 폴리곤의 수가 200개를 넘는 매우 복잡한 부분이 아니라면 30 프레임 이상의 수행 속도를 보여주므로 실시간 페인트 시뮬레이션이 가능하다.

8. 결론 및 연구 방향

광범위한 영역으로 분사하는 페인트의 실시간 시뮬레이션은 아직 많은 연구가 이루어지지 않았다. 본 논문에서는 광범위로 분사되는 페인트가 표면에 도색 되는 시뮬레이션을 기존 파티클 시스템에 비해 빠른 속도로 수행할 수 있고, 사실적으로 시각화 할 수 있는 새로운 방법을 제시하였다. 또한 수행 속도를 측정하여 실제로 실시간 시뮬레이션이 가능하다는 것을 확인 하였다. 추후 연구 방향으로 현재는 페인트가 단순히 일직선으로 날아가는 상황을 고려하였지만, 바람이 불어 페인트가 날리는 경우나 날아갈 때 중력의 영향을 받는 경우에 대해 생각해 볼 수 있다. 그리고 페인트는 유체이기 때문에 벽에 너무 많은 양이 묻게 되면 흘러내리게 되는데 이를 표현할 수 있는 방법을 고민함으로써 더욱 사실적인 페인트 시뮬레이션이 가능할 것이다.

참 고 문 헌

- [1] Paul Adams, Douglas Domermuth, "Visualization of Steep Breaking Waves and Thin Spray Sheets around a Ship," Proc. of the 14th IEEE Visualization, pp. 555-559, 2003.
- [2] Alex Pang, Kyle Smith, "Spray Rendering: Visualization Using Smart Particles," Proc. of IEEE Visualization, pp. 283-290, 1993.
- [3] Maneesh Agrawala, Andrew C. Beers, Marc Levoy, "3D Painting on Scanned Surfaces," Proc. of Symposium on Interactive 3D Graphics, Monterey, California, pp. 145-150, April 1995.
- [4] Pat Hanrahan and Paul Haerberlit, "Direct WYSIWYG Painting and Texturing on 3D Shapes," Proc. of ACM SIGGRAPH, pp. 215-223, August 1990.
- [5] Takeo Igarashi, Dennis Cosgrove, "Adaptive unwrapping for interactive texture painting," Proc. of ACM Symposium on Interactive 3D Graphics, pp. 209-216, 2001.
- [6] E. Freund, D. Rokossa, and J. Rossmann, "Process oriented approach to an efficient off-line programming of industrial robots," Proc. of IEEE Industrial Electronics Society Conference, pp. 208-213, 1998.