

3차원 애니메이션을 위한 회화적 스트로크의 동적 관리 기법

이효근[†], 류승택^{**}, 윤경현^{***}

요 약

본 연구에서는 3차원 비사실적 애니메이션을 위한 프레임간 유사성 유지 기법을 제공한다. 비사실적 애니메이션 분야에서는 결과 영상이 매 프레임마다 임의적으로 변경되는 것을 방지하기 위하여 프레임간 유사성(frame-to-frame coherence)을 유지하는 것이 매우 중요하다. 본 연구에서는 이러한 유사성을 유지하기 위하여 3차원 파티클 시스템을 적용하였다. 각각의 파티클들은 결과 영상에서 하나의 브러시 스트로크를 나타낸다. 본 연구에서는 파티클들을 물체의 표면에 고정함으로써 물체가 움직이거나 회전할 때 프레임간 유사성을 유지한다. 물론 카메라가 Zoom in/out 될 때에도 프레임간 유사성은 유지된다. 하지만 물체 표면의 브러시 스트로크도 또한 동일하게 Zoom in/out 될 것이다. 이러한 결과로서 생성되는 브러시 스트로크들은 지나치게 커지거나 지나치게 작아지는 경우, 사람이 그린 듯한 브러시 스트로크라는 느낌을 주기 어렵다. 따라서 본 연구에서는 카메라의 Zoom in/out 시에 브러시 스트로크를 동적으로 관리하여 화면 공간상에서는 그 크기가 일정하게 유지되도록 함으로써 결과 영상을 자연스럽게 유지하면서 프레임간 유사성을 유지하도록 하였다.

A Study on the Dynamic Painterly Stroke Generation for 3D Animation

Lee, Hyo Keun[†], Ryoo, Seung Taek^{**}, Yoon, Kyung Hyun^{***}

ABSTRACT

We suggest the dynamic stroke generation algorithm that provides frame-to-frame coherence in 3D non-photorealistic animations. We use 3D particle system to eliminate the visual popping effect in the animated scene. Since we have located particles on the 3D object's surface, the coherence is maintained when the object or the camera is moving in the scene. Also, this algorithm maintains the coherence when camera is zooming in/out. However, the brush strokes on the surface also zoom in/out. This result(too large or too small brush strokes) can not represent hand-crafted brush strokes. To remove this problem, we suggest stroke generation algorithm that dynamically maintains the number of brush stroke and its size during camera zoom in/out.

Key words: Non-Photorealistic Rendering(비사실적 렌더링), Stroke Generation(스트로크 생성), Frame-to-Frame Coherence(프레임간 유사성), 3D Animation(3차원 애니메이션)

※ 교신저자(Corresponding Author): 류승택, 주소: 경기도 오산시 양산동 411번지(447-791), 전화: 031)370-6439, FAX: 02) 820-5301, E-mail: stryoo@hs.ac.kr
접수일: 2004년 5월 24일, 완료일: 2004년 11월 1일
[†] 삼성전자 TN 총괄 무선 사업부 연구원
(E-mail: xcrosslee@daum.net)

^{**} 정회원, 한신대학교 소프트웨어학과 조교수
^{***} 정회원, 중앙대학교 컴퓨터 공학부 교수
(E-mail: khyoon@cau.ac.kr)

※ 본 논문은 과학기술부 국가지정 연구소 사업(No. 2000-N-NL-01-C-285)의 지원을 받아 수행되었습니다.

1. 서론

전통적 그래픽스는 사물을 사실적으로 표현하는 방법을 연구하였고, 광선추적법(Ray Tracing)이나 레디오시티(Radiosity)와 같이 사물의 물리적인 특성을 시뮬레이션 하여 사실적인 영상을 만들어 냈다. 그러나 기존의 렌더링 기술은 초현실적인 영상을 만드는 결과를 초래하였고 이러한 문제 때문에 비사실적 렌더링에 대한 연구가 시작되었다.

비사실적 렌더링(Non-photorealistic rendering)이란 사람이 그린 듯한 영상을 창조해 내는 기법이다. 대표적인 비사실적 렌더링 연구 분야에는 회화적 렌더링(Painterly Rendering)[1,6,9,12], 펜/잉크 렌더링(Pen-and-Ink Rendering)[2,3], 툰렌더링(Toon Rendering), 테크니컬 스케치(Technical Sketch)[19], 모자이크(Mosaic), 수묵화 등이 있다. 본 연구에서는 회화적 렌더링 기법을 적용한 3차원 애니메이션에서 스트로크의 프레임간 유사성을 유지하는 기법을 제안한다. 회화적 렌더링 기법을 3차원 애니메이션에 적용하기 위해서는, 스트로크의 위치 및 방향, 길이, 폭 등이 결정되어야 한다. 본 연구에서는 3차원 다각형 모델을 기반으로 하며, 스트로크의 위치는 3차원 물체 표면에 고정시켰으며 스트로크들은 물체의 굴곡을 따라서 진행하도록 방향을 결정하였다. 스트로크의 길이도 마찬가지로 그 굴곡에 따라 스트로크들이 자연스러운 곡선을 이루도록 결정하였고, 스트로크의 폭은 스트로크의 분포에 따라 표면을 모두 덮을 수 있는 폭으로 결정하였다. 스트로크의 프레임간 유사성을 유지하기 위한 방법으로는 파티클 시스템과 동적 스트로크 관리 기법을 제시하였다.

2. 관련 연구

스트로크 기반의 3차원 렌더링은 그 방식에 따라 크게 두 가지로 구분할 수 있다. 한 가지 방식은 스트로크를 하나의 집합으로 생성하는 것이다. 이 방식은 스트로크들을 하나의 텍스처로 모아서 3차원 물체의 표면에 맵핑하는 방식이다. 2001년 Emil Praun은 텍스처를 이용한 3차원 실시간 해칭(Hatching) 기법을 선보였다[7]. 그는 스트로크들의 집합을 하나의 텍스처로 표현하여 스트로크 처리에 효율성을 기하고, 3차원 물체에 하드웨어 기반 텍스처 맵핑을 수행함으로써 실시간 처리를 가능하게 하였다. 스트로크들을

명암과 밀도 차에 따라 일정한 단계를 적용하여 일련의 텍스처 맵인 TAM(Tornal Art Map)을 생성하였는데, TAM은 생성 시에 인접한 텍스처들을 포함시키도록 생성된다. 따라서 명암의 변화에 따라 새로운 텍스처를 선택하여 렌더링하게 되면, 이전의 텍스처의 특성을 그대로 반영한 상태로 렌더링되므로 프레임간의 유사성을 유지할 수 있게 된다.

스트로크 기반의 3차원 렌더링의 다른 방식은 하나 하나의 스트로크를 개별적으로 생성하여 물체의 표면에 직접 렌더링하는 방식이다. 1996년 Babara J. Meier는 개개의 스트로크를 직접 생성하여 3차원 물체를 렌더링하는 기법을 선보였다[9]. Meier는 원기둥, 원뿔과 같은 단순한 모델을 다각형화 한 모델을 사용하였다. 이러한 다각형 모델의 표면 위에 3차원 파티클들을 배치하였으며, 각각의 파티클들은 스트로크의 위치를 나타내게 하였다. 따라서 물체가 이동하거나 회전할 때, 물체 표면에 고정되어 있는 파티클들도 동일하게 이동하거나 회전하게 됨으로써 프레임간의 유사성을 유지할 수 있었다. 반면, Meier의 시스템은 물체를 Zoom in/out 할 때에 스트로크의 크기가 커지거나 작아지게 된다. 이는 스트로크가 지나치게 커지거나 지나치게 작아지는 경우 자연스럽게 스트로크 렌더링의 결과를 낳게 된다. 본 연구에서는 Meier와 같이 개개의 스트로크를 렌더링하는 시스템을 기반으로 한 렌더링 기법을 이용하여 물체의 이동 및 회전 시뿐만 아니라, 물체의 Zoom in/out 시에도 스트로크의 형태를 유지하며 프레임간의 유사성을 유지하는 기법에 대해서 소개한다.

3. 스트로크 기반 3차원 렌더링의 시스템 구조

본 연구에서 제안한 스트로크 기반 3차원 렌더링 시스템은 그림 1과 같이 3차원 모델의 관리, 스트로크 모델링, 스트로크 렌더링, 스트로크의 동적 관리, 네 부분으로 구성된다.

3차원 모델의 관리 단계에서는 3차원 모델을 입력받아 본 연구에서 사용하게 될 다각형 그물 구조(Polygon Mesh)의 형태로 가공하게 된다. 3차원 모델의 입력은 3DS MAX로 모델링 된 3차원 모델을 ASE(ASCII SCENE EXPORT)형태의 텍스트 파일 형태로 추출하여 입력받게 된다. 본 연구에서 사용한 모델은 정점 리스트와 선분 리스트, 그리고 각 정점에 할당된 색상 리스트로 구성된다.

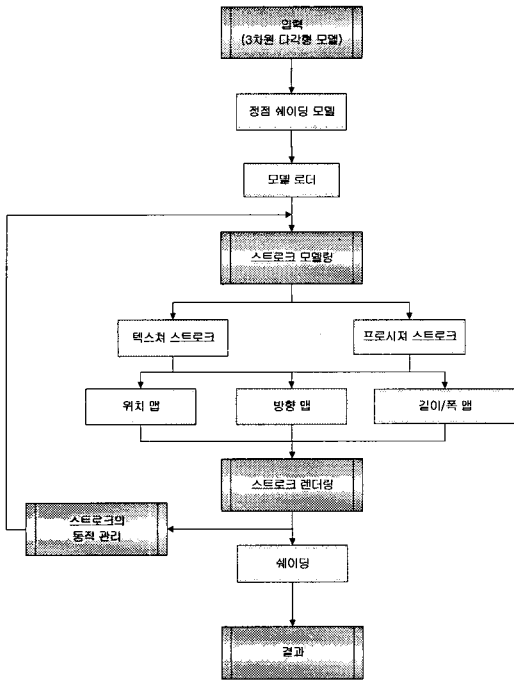


그림 1. 스트로크 기반 3차원 렌더링 시스템 구조

스트로크 모델링 단계에서는 각 스트로크들의 속성들을 결정하게 된다. 본 연구에서 사용하는 스트로크는 유화적 기법의 스트로크로써, 폭이 있고 긴 유선형의 형태를 띄게 된다. 스트로크 모델링 단계에서는 이러한 스트로크를 생성하기 위하여 스트로크의 위치, 방향, 길이, 폭, 색 등의 속성을 결정하게 된다.

스트로크 렌더링 단계에서는 스트로크 모델링 단계에서 결정된 스트로크의 속성들을 기반으로 스트로크를 화면상에 보여주는 단계이다. 개개의 스트로크들은 폭이 있고 긴 유선형의 형태로 렌더링된다.

마지막 단계는 스트로크를 동적으로 관리하는 단계이다. 이 단계에서는 물체의 Zoom in/out 시에 스트로크의 형태를 자연스럽게 유지하면서 프레임간 유사성을 유지하도록 동작한다.

4. 스트로크 생성

4.1 스트로크 모델링

4.1.1 스트로크의 위치 결정

본 연구에서 사용하는 3차원 물체는 삼각형으로 이루어진 다각형 모델이다. 이러한 모델에 스트로크를 위치시키기 위해서는 각 삼각형의 표면에 스트로

크의 시작 위치를 위치시키고 삼각형들로 이루어진 표면을 따라 스트로크를 진행시켜야 한다. 따라서 우선 물체를 이루고 있는 모든 삼각형 표면상에 스트로크들의 시작 위치들을 결정해야 한다. 이러한 스트로크의 시작위치들을 나타내기 위해서 본 연구에서는 3차원 파티클 시스템을 사용하였다. 본 연구에서는 삼각형의 중점과 중선을 이용하여 일정한 패턴으로 파티클들을 분포시킨다. 그림 2는 삼각형의 중점과 중선을 이용하여 파티클들을 배치하는 모습이다. 물체를 이루는 하나의 삼각형에 대해서 스트로크의 첫 시작 위치는 그 삼각형의 중점이 된다. 그 다음 시작 위치는 삼각형의 세 중선 상에서 중점으로부터 정해진 거리만큼 떨어진 지점에 3개의 새로운 파티클이 생성된다. 다음으로는 마찬가지로 중선 상에 3개의 위치에 생성하게 되는데, 이때는 바로 앞 단계와 반대 방향으로 3개의 파티클을 생성한다. 같은 방법으로 이전의 여섯 개의 위치보다 중점으로부터의 거리가 두 배인 지점에 새로운 여섯 개의 파티클들을 생성한다. 이러한 과정은 생성되는 파티클들이 삼각형의 내부에 배치될 수 있을 때까지 계속 진행된다.

파티클들, 즉 스트로크의 시작위치가 일정 패턴으로 배치되어 스트로크들도 마찬가지로 패턴화 될 가능성이 있으나 이는 다음 4.1.3 절의 스트로크의 길이를 확장하는 과정에서 보완된다.

4.1.2 스트로크의 방향 결정

3차원 물체는 원근감, 명암, 색상 등을 이용하여 2차원 상에서 입체감 있게 보일 수 있다. 하지만 본 연구에서와 같이 스트로크를 이용하여 3차원 물체를 렌더링 할 때는 이러한 특성들만으로는 3차원 정보를 효과적으로 표현할 수 없다. 3차원 물체를 스트로크로 렌더링하기 위해서는 각 스트로크의 방향을 결정하는 방법에 따라 그 형태를 효과적으로 전달할

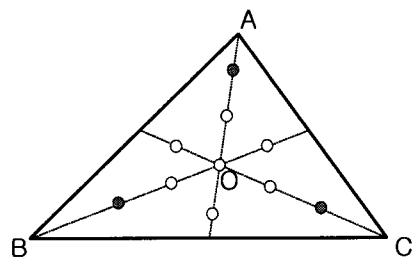


그림 2. 삼각형의 중점과 중선을 이용한 파티클의 배치

수 있다[4,10,11]. 각각의 스트로크 방향은 임의로 지정하거나 사용자가 지정한 한 방향으로 모두 일률적으로 주어질 수도 있다. 하지만 이들 방법을 사용했을 때는 물체의 외곽이나 뚜렷한 경계가 존재하는 부분에서는 문제가 없겠지만 정면으로 보이는 곡면의 정보를 잘 전달하지 못한다. 따라서 물체의 모양을 잘 전달할 수 있는 스트로크 방향 결정 방법이 필요하다. 심리학자들은 여러 연구를 통하여 3차원 물체를 스트로크로 표현할 때 제1주방향(First Principal Direction) 또는 제2주방향(Second Principal Direction)으로 스트로크들을 배치시키면 곡면을 이루는 3차원 물체의 정보를 보다 효과적으로 전달할 수 있음을 보여주었다. 일반적인 3차원 물체 표면의 한 지점에 대해서, 굴곡이 제일 심한 방향을 제1주방향이라 하고 제1주방향에 직각인 방향을 제2주방향이라고 한다. 이러한 제1주방향 및 제2주방향은 그 지점에 접해있는 평면(Tangent Plane)상에 존재한다. 그림 3에서 N 은 곡면의 한 지점에서의 법선 벡터이다. 또한 T_1 및 T_2 는 각각 그 지점에서의 제1주방향 및 제2주방향이다. 이 T_1 과 T_2 는 N 에 수직인, 접면상에 존재하게 된다. 본 연구에서는 3차원 물체의 형태를 효과적으로 전달하기 위하여 제1주방향 및 제2주방향의 스트로크를 사용한다.

앞서 설명한 주방향은 일반적인 3차원 물체에 대한 주방향을 의미한다. 하지만 본 연구에서 사용하는 물체는 삼각형 평면들의 집합으로 이루어진 다각형 그물망 형태의 모델이다. 이러한 모델을 이루고 있는 각 삼각형은 평면이기 때문에 앞에서 설명한 주방향의 개념을 그대로 적용할 수가 없다. 따라서 다각형 그물망 구조의 모델에 적합한 주방향 결정 방법이 필요하다. 이러한 다각형 그물망 구조의 주방향 결정 방법은 여러 가지 방법이 연구되었다. 하지만 이 방법들은 모두 오차를 가지고 있다[13]. 이러한 오차들은 곡률이 사방으로 거의 비슷한 완만한 곡면

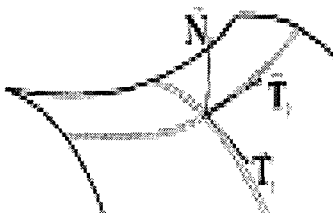


그림 3. 곡면에서의 주방향[11]

(Umbilics)이나 평면에서 나타난다.

본 연구에서는 정규 굴곡 근사법(Normal Curvature Approximation Method)에 의해 각 삼각형의 꼭짓점의 주방향을 검출하였다. W 를 지역 좌표계 L 을 이루는 정점 p 에 대한 Weingarten 행렬이라고 하자. 그리고 p 에 인접한 n 개의 정점들 중 i 번째 정점을 q_i 라 하자(그림 4). 이때, 점 p 에 접한 평면(점 p 에서의 노말 벡터(N_p))에 의해 생성되는 평면을 말함)상에 선분 pq_i 를 투영시킨 벡터를 구하고 그 벡터를 정규화 한 벡터를 y_i 라 하자(pq_i 는 지역 좌표계 L 상에서 정의된다)

이때 y_i 방향으로의 정규 굴곡 k_{y_i} 는 다음 식 1과 같다. 하지만 본 연구에서는 정규 굴곡 k_{y_i} 를 구하기 위하여 Weingarten matrix를 이용하지 않고 식 2와 같은 근사화 방법에 의하여 정규 굴곡을 구한다.

$$k_{y_i} = y_i^T W y_i \tag{식 1}$$

$$k_{y_i} = 2 \frac{(p - q_i) \cdot N_p}{(p - q_i) \cdot (p - q_i)} \tag{식 2}$$

식 2의 의미는 다음과 같다. $(p - q_i)$ 는 q_i 에서 p 로 가는 벡터를 나타내고 점 p 에서의 법선 벡터인 N_p 와의 내적은 N_p 와 벡터 $q_i p$ 가 이루는 각을 의미한다. 이때, 벡터 N 벡터와 $q_i p$ 벡터의 내적은 N 벡터에 투영한 $q_i p$ 벡터의 길이로써 정규 굴곡 근사화에 대한 가중치를 의미한다. 물체의 주방향은 그 지점의 주변 정규 굴곡 중 제일 큰 굴곡을 향하는 방향이 된다. 따라서 p 의 인접 정점들과 그 정점들을 잇는 선분을 따라서 각 정규 굴곡을 검출하고 이 중 제일 큰 정규 굴곡을 구하면, 이 지점으로 향하는 방향이 그 지점에서의 주방향이 된다. 이렇게 각 꼭지점의 주방향을

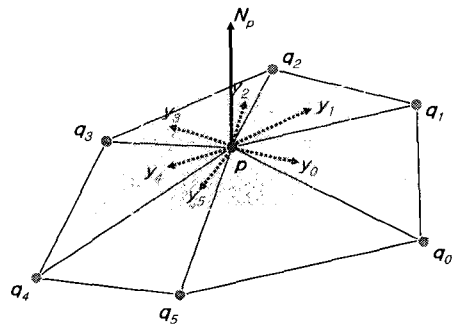


그림 4. 정규 굴곡 근사법

결정한 뒤 삼각형 내부 지점에 대해서는 삼각형의 3개 꼭지점의 주방향으로부터 삼중 선형 보간(Trilinear Interpolation)을 하여 주방향을 얻게 된다. 그림 5는 정규 굴곡 근사법에 의하여 주방향을 검출한 결과를 보여준다.

4.1.3 스트로크의 길이 결정

본 연구에서 사용하는 스트로크는 다양한 길이를 갖는 긴 스트로크이다. 이러한 긴 스트로크는 물체의 표면을 따라서 위치하며 부드러운 곡선의 형태를 보인다. 이를 위하여 물체의 표면을 따라서 스트로크를 길게 확장하는 알고리즘과 생성된 스트로크를 부드러운 곡선으로 만드는 방법이 필요하다.

4.1.2 절에서 제시한 방법으로 스트로크를 생성하면 스트로크의 시작위치로부터 제1주방향(혹은 제2주방향)으로 향하는 하나의 직선 스트로크가 생성된다. 스트로크의 길이를 연장하기 위해서는 이 직선의 스트로크의 끝점에서 다시 스트로크의 주방향을 결정한 후 새로운 직선의 스트로크를 생성한다. 이러한 과정을 반복하면 원하는 길이의 스트로크를 생성할 수 있다. 이러한 과정을 스트로크 확장이라고 명명한다. 이때, 본 연구에서 사용하는 모델은 삼각형들로 구성된 다각형 그물망 구조의 형태이므로 삼각형이 서로 맞물린 지점에서 불연속지점이 생긴다. 따라서 이러한 모델에 적합한 몇 가지 규칙에 의해 스트로크를 확장한다. 다음은 스트로크 확장을 위한 규칙들이다.

- 이전 스트로크의 끝점에서 새로 스트로크를 확장할 때는 그 끝점에서 새로운 주방향을 구함으로써 방향을 결정한다. 이때 앞선 스트로크의

방향과 새로 구한 스트로크의 방향의 사이각이 미리 결정된 문턱치(Threshold) 값을 만족하는 경우에만 스트로크를 확장한다.

- 스트로크는 미리 정의한 스트로크의 최대 길이에 도달하면 확장을 멈춘다.
- 확장하는 동안에 삼각형의 경계를 가로지르는 경우는 삼각형의 경계까지만 확장하고 그 지점에서 다시 새로운 방향을 결정하여 확장한다.
- 기존에 생성된 스트로크를 가로지르거나 너무 근접하는 경우는 스트로크 확장을 멈춘다.

그림 6은 이러한 규칙에 의거하여 스트로크를 확장하는 과정을 보여주고 있다. 그림 6(a)를 보면 최초 스트로크의 위치 P_1 으로부터 그 지점의 주방향 Dir_1 으로 스트로크를 한 단계 확장한 지점이 P_2 가 된다. P_2 에서 스트로크를 새로 확장해야 하는데 이때 P_2 에서의 주방향 Dir_2 를 다시 구하여 그 방향으로 확장한다. 이러한 과정을 반복하며 미리 정의된 길이에 도달할 때까지 스트로크들을 확장하게 된다. 그림 6(a)의 경우는 하나의 삼각형 내에서 스트로크를 확장하는 가상적인 경우를 예로 든 것이다. 또한 그 삼각형 내에는 기존의 다른 스트로크가 없다는 가정 하에 스트로크를 확장했으므로 스트로크의 확장에 특별한 제한 없이 정의된 길이에 도달할 때까지 스트로크를 확장할 수 있었다. 하지만 이러한 경우는 실제로는 거의 일어나지 않는다. 따라서 실제 상황에서 스트로크 확장에 영향을 끼치는 여러 가지 상황을 살펴 보도록 하겠다.

우선 스트로크의 확장은 그 방향이 유사성을 가져야만 계속해서 확장된다. 즉, 그림 6(a)에서 Dir_1 과 Dir_2 는 그 방향이 서로 유사하고, 마찬가지로 Dir_2 와 Dir_3, \dots, Dir_{n-2} 와 Dir_{n-1} 의 방향이 유사하다. 여기에서 방향이 유사하다는 말은 두 방향이 급격하게 변하지 않고 완만하게 변화한다는 것을 의미한다. 본 논문에서는 이러한 방향의 유사성을 검사하기 위하여 두 벡터의 스칼라 곱(Dot Product)를 이용하였다. 즉 Dir_1 과 Dir_2 의 방향의 유사성을 검사하기 위하여 두 벡터 Dir_1 과 Dir_2 의 스칼라 곱을 구한 후 그 사이각이 미리 정의한 각 이상 벌어져야만 두 벡터의 방향이 완만하게 변한다고 정의한 것이다. 따라서 스트로크를 확장함에 있어 앞뒤의 방향이 급격하게 변화하는 부분에서는 스트로크의 확장을 멈추게 된다. 다음으로 물체를 이루는 삼각형들의 경계에서 스트로크의

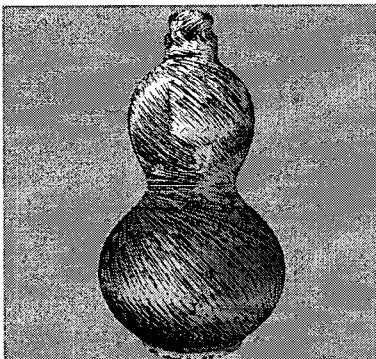


그림 5. 정규 굴곡 근사법에 의한 주방향 검출

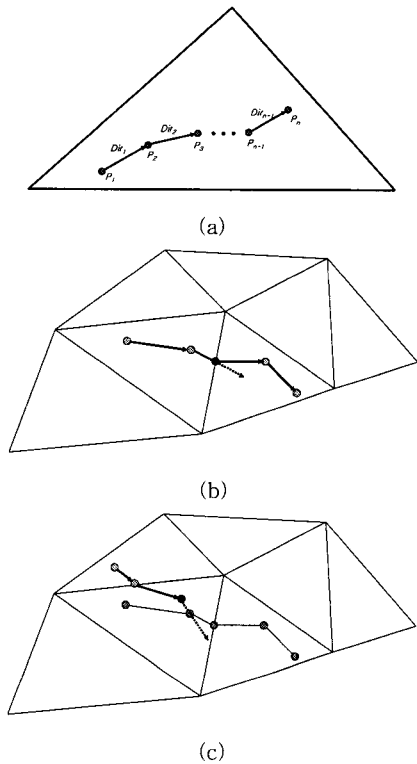


그림 6. 스트로크의 확장

확장에 제어를 받게 된다. 스트로크를 확장하는 과정에서 확장된 스트로크가 삼각형의 경계를 가로지르는 경우에는 그림 6(b)와 같이 그 교점까지만 스트로크를 확장하고 다시 그 교점에서 새로운 방향을 구하여 스트로크를 확장하게 된다. 마지막으로 스트로크들은 서로 교차할 수 없도록 한다. 즉 그림 6(c)와 같이 확장하는 스트로크가 기존의 스트로크를 가로지르는 경우에는 스트로크의 확장을 멈추게 된다. 스트로크의 위치를 생성하는 부분을 언급한 4.1.1절에서 파티클들을 규칙적으로 배열함으로써 스트로크도 규칙적인 모양으로 생성될 것이라고 문제를 제기했다. 이 문제는 스트로크의 확장과정에서 해소될 수 있다고 언급한 바 있다. 스트로크 확장은 스트로크간에 서로 겹치지 않으면서 길게 확장되기 때문에 앞에서 생성한 모든 파티클들이 스트로크를 생성하지는 않는다. 또한 각 스트로크들은 모두 같은 길이와 일률적인 방향으로 생성되지 않는다. 따라서 초기에 스트로크의 위치들을 규칙적으로 배열하더라도 최종적으로 생성되는 스트로크들은 자연스러운 분포를 갖도록 생성되는 것이다.

4.1.4 스트로크의 폭 결정

본 연구에서 사용하는 스트로크는 펜이나 연필 스트로크와 같은 단순한 선형 구조가 아닌 유화에서 나타나는 폭이 있는 붓 스트로크이다. 따라서 일정한 폭을 가지는 스트로크를 생성해야 한다. 스트로크의 폭은 사용자가 원하는 폭으로 미리 정의를 한다. 이 크기는 세계 공간상에서의 크기가 된다. 따라서 각 스트로크들을 렌더링한 후의 결과는 스트로크를 보는 각도에 따라 그 폭의 간격에 약간의 차이가 생긴다. 스트로크의 폭을 적용하기 위해서는 스트로크의 좌우로 정해진 폭의 1/2만큼씩 늘려주게 된다. 이 폭을 늘리는 방법은 그림 7에 나타나 있다. 제어점 P_1 을 포함하는 삼각형의 법선 벡터를 N_1 이라 하자. 다음 P_1 에서의 주방향을 Dir_1 이라 하자. 이때 N_1 과 Dir_1 의 외적을 구하면 Dir_1 에 직각인 벡터들을 구할 수 있다. 이렇게 구한 벡터들을 단위화하여 점 P_1 에서부터 구해진 단위 벡터를 따라 미리 정의된 폭의 1/2만큼 떨어진 지점 L_1 (혹은 R_1)을 구하면 이 지점이 P_1 의 폭을 늘린 지점이 된다. 반대편의 폭은 L_1 (혹은 R_1)과 반대 방향으로 폭의 1/2만큼 떨어진 지점을 구하면 된다. 마찬가지로 방법으로 나머지 제어점 P_2, P_3, P_4 에 대해서 원하는 폭만큼 벌어진 제어점들을 구한다. 이렇게 구한 $L_1 \sim L_4$ 점들과 $R_1 \sim R_4$ 점들이 폭을 가지는 스트로크의 새로운 제어점이 된다.

4.2 폭이 있는 긴 곡선의 스트로크 렌더링

스트로크의 렌더링 단계에서는 스트로크 모델링 단계에서 결정된 스트로크의 위치 및 방향, 길이, 폭에 따라 렌더링을 하게 된다. 이때 폭이 있는 긴 곡선의 스트로크를 렌더링하기 위하여 특별한 방법이 필요하다. 우선 곡선의 스트로크로 렌더링하기 위하여

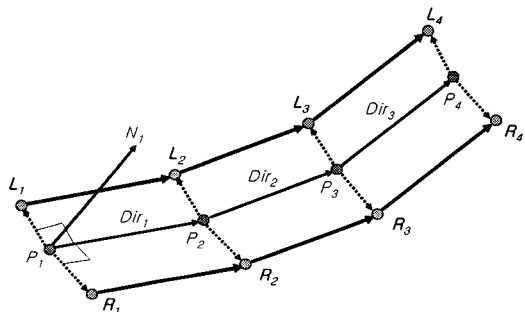


그림 7. 스트로크의 폭 결정

직선들로 이루어진 스트로크를 베지어(Bezier) 곡선으로 평가해야 한다[12]. 그림 8은 베지어 곡선을 이용하여 곡선의 스트로크를 생성한 모습을 보여준다. 네 개의 점은 긴 스트로크를 구성하는 제어점이 되고, 점선은 직선으로 이루어진 긴 스트로크를 나타낸다. 이러한 제어점에 대해 베지어 곡선을 생성한 것이 실선으로 표시한 곡선이다.

앞서 스트로크의 폭을 결정할 때, 스트로크를 구성하는 제어점들을 좌우로 확장하여 새로운 제어점들을 생성하였다. 따라서 폭이 있는 긴 곡선의 스트로크를 렌더링할 때는 이렇게 좌우로 생성된 제어점들을 이용하여 렌더링해야 한다. 또한 베지어 곡선을 실제 화면상에 그릴 때는 베지어 곡선을 작은 간격으로 잘라서 직선들을 이어서 그리게 된다. 그림 9는 폭이 있는 긴 곡선을 렌더링하는 과정을 보여준다. 우선 좌, 우의 각각의 제어점들에 대해서 베지어 곡선을 생성하고(그림 9(a)), 각 베지어 곡선을 등 간격으로 나눈다(그림 9(b)). 다음으로 두 베지어 곡선을 등 간격으로 나눈 점들을 삼각형 띠(Triangle Strip) 형태(그림 9(c))로 구분하고 이를 삼각형화하여 렌더링하면 폭이 있는 긴 곡선의 스트로크가 형성된다(그림 9(d)). 이때 스트로크의 색상은 스트로크 시작점의 색상을 사용하며, 스트로크 시작점의 색상은 스트로크 시작점을 포함하는 삼각형의 세 꼭짓점의 색상으로부터 이중 선형 보간하여 얻는다. 삼각형의 세 꼭짓점의 색상은 이미 입력 자료로 읽어 들인 자

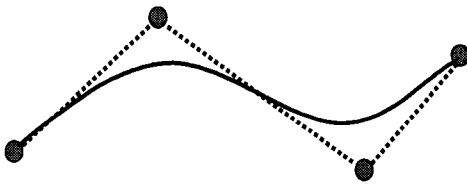


그림 8. 곡선의 스트로크 생성

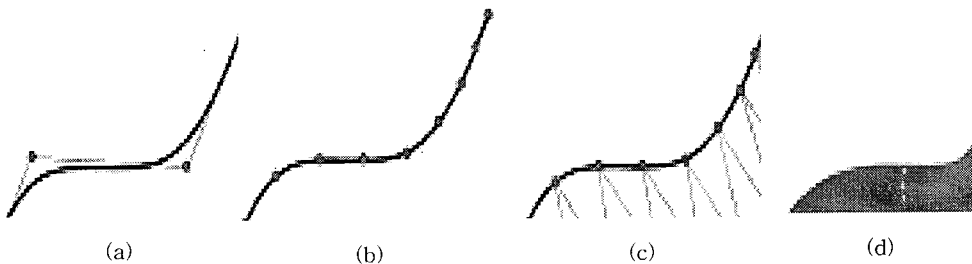


그림 9. 폭이 있는 긴 곡선의 스트로크 렌더링

료이므로 이용이 가능하다.

5. 프레임간 유사성 유지 기법

5절에서는 프레임간 유사성 유지 기법에 대해서 설명한다. 스트로크에 기반을 둔 3차원 렌더링에서 프레임간 유사성 유지 기법은 두 가지 상태에 따라 다르게 된다. 한 가지 경우는 물체가 회전하거나 이동하는 경우이다. 이때에는 스트로크도 물체와 함께 따라서 회전하거나 이동해야 한다. Meier는 이를 위하여 스트로크를 물체의 표면에 고정시켰다[9]. 이의 실제 구현은 개개의 스트로크를 나타내는 파티클(Particle)들을 이용한 파티클 시스템의 형태로 구현되었다. 다른 한 가지 경우는 물체(또는 카메라)가 Zoom in/out 할 때이다. 이때에는 보통 스트로크도 Zoom in/out 시에 같이 커지거나 작아지게 된다. 또한 스트로크의 폭도 같이 두꺼워지거나 얇아지게 된다. 이는 Zoom in/out이 심할 경우에는 스트로크의 크기가 지나치게 커지거나 지나치게 작아짐으로써 부자연스러운 결과를 초래한다. 이의 해결 방안으로 본 연구에서는 스트로크의 동적 관리 기법을 소개한다.

5.1 파티클 시스템

Meier는 자신의 연구에서 프레임간의 유사성을 유지하기 위한 방법으로 파티클을 사용했다[9]. Meier는 원기둥 위에 원뿔을 올려놓은 모델을 스트로크들로 렌더링하여 건축도미를 만들어냈다. Meier는 원기둥과 원뿔을 다각형화 한 후 각 다각형 표면에 임의의 위치와 임의의 개수로 파티클들을 심었다. 이 파티클들은 렌더링 과정에서 스트로크들의 위치가 되었으며 파티클들을 물체 표면에 고정 배치했기 때문에 물체의 회전 및 이동시에 프레임간의

유사성을 유지할 수 있었다. 본 연구에서는 Meier가 사용한 3차원 파티클 시스템을 사용하여 스트로크의 위치를 유지한다. 4.1.1절에서 생성된 각 스트로크 위치들이 파티클이 되며 이러한 파티클을 계속 유지함으로써 물체의 회전 및 이동시에 스트로크의 프레임 간 유사성을 유지하게 된다. 그림 10은 본 연구에서 사용한 모델의 표면에 파티클들을 배치한 모습을 보여준다.

5.2 스트로크의 동적 관리

Meier가 스트로크를 표현하기 위하여 제안한 파티클 시스템에는 단점이 존재한다. 즉, 물체(또는 카메라)가 Zoom in/out 시에 스트로크가 같이 확대 혹은 축소되는 경우의 문제이다. Meier가 제안한 시스템은 물체가 Zoom in/out 되면 물체에 고정되어 있는 스트로크도 같이 확대되거나 축소된다. 이때 스트로크가 지나치게 확대되거나 지나치게 축소되면 화면상에 부자연스러운 스트로크들을 보여지게 된다. 본 연구에서는 이러한 문제를 해결하는 방법으로 스트로크를 동적으로 관리하는 시스템을 제안한다. 동적 스트로크 관리의 기본 아이디어는 물체가 Zoom in/out 될 때 스트로크의 크기를 동적으로 조절하는 것이다. 즉, 물체가 가까워지거나 멀어지는 것과 상관없이 화면상의 스트로크의 크기는 일정하게 유지되도록 하는 것이다. 그림 11(a)는 관찰자와 물체의 초기 상태를 나타낸다. d_0 는 관찰자와 물체간의 최초 거리이고 w_0 는 사용자가 지정한 물체의 스트로크의 초기 폭을 나타낸다. 이때 그림 11(b)처럼 물체가 관

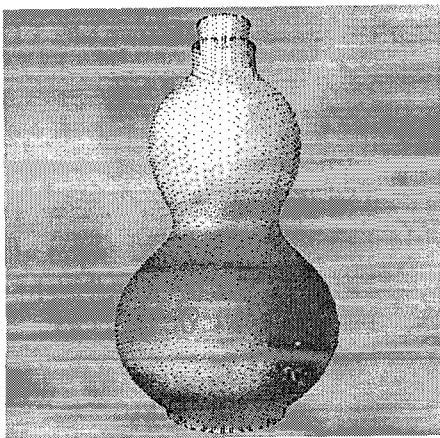


그림 10. 파티클 배치 모습

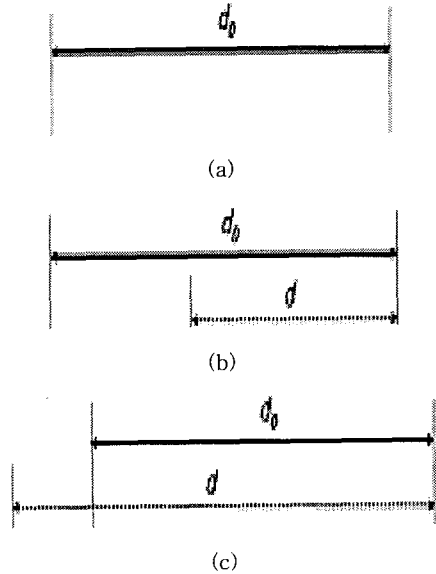


그림 11. 동적으로 스트로크의 폭을 결정

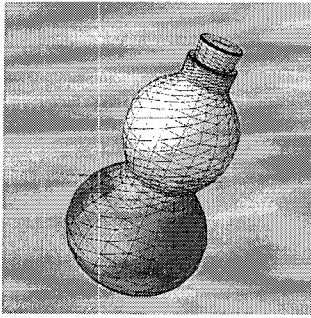
찰자에게 가까워지거나 그림 11(c)처럼 물체가 관찰자로부터 멀어지는 경우, 이때의 물체와 관찰자간의 거리를 d , 새로 결정되는 스트로크의 폭을 w 라 하자. 물론 w_0, w, d_0 및 d 는 모두 세계 공간상에서의 단위이다. 이때 물체가 최초의 위치로부터 이동해서 관찰자로부터의 거리가 변경된 경우 새로운 위치에서의 스트로크의 폭 w 는 식 3으로 구한다.

$$w = \left[\frac{d}{d_0 / w_0} \right] + 1 \tag{식 3}$$

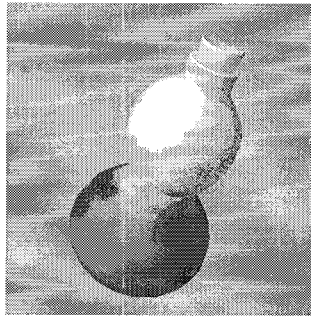
6. 구현결과 및 검토

본 논문에서 제안된 알고리즘을 구현 및 검토를 하기 위해 Pentium III 700 MHz를 사용하였고 총 960 개의 다각형으로 이루어져 있는 모델을 사용하였다. 또한, 회화적 렌더링을 위하여 2152 개의 스트로크를 사용하였으며 그 렌더링 시간은 약 2초가 소요된다.

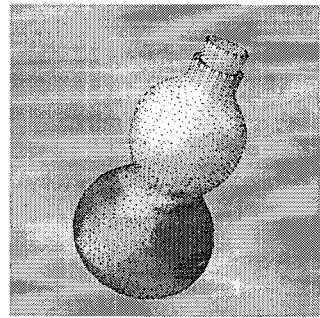
그림 12는 폭이 있는 긴 유선형의 스트로크를 이용하여 3차원 물체를 렌더링하는 과정을 보여준다. 그림 12의 왼쪽 그림은 위에서부터 순서대로 모델이 회전하는 모습을 보여주고 있다. 각 스트로크들의 위치 및 방향은 모델의 표면을 따라서 고정되어 있기 때문에 물체가 이동하거나 회전할 때, 스트로크들도



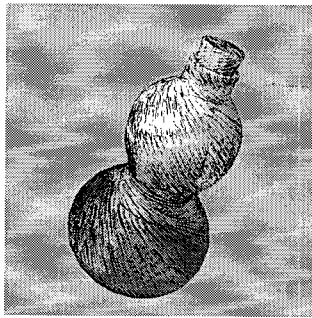
(a) 3차원 다각형 모델



(b) 정점 셰이딩



(c) 파티클 배치



(d) 스트로크 방향 검출



(e) 스트로크 렌더링

그림 12. 폭이 있는 긴 곡선의 스트로크 렌더링

자연스럽게 이동 및 회전하기 때문에 프레임간 유사성을 유지하는 모습을 보여주고 있다. 그림 13은 프레임간 유사성을 유지하는 결과를 보여주고 있다. 그림 13의 오른쪽 그림은 물체의 Zoom in/out 시에 프레임간 유사성을 유지하는 모습을 보여주는 결과이다. 그림 13의 오른쪽 그림은 위에서부터 순서대로 물체가 Zoom in 되는 모습을 보여주고 있다. 이 연속적인 프레임들을 살펴보면 각 스트로크들의 폭은 화면상에서 일정하게 유지되면서 Zoom in이 진행되면서 그 사이사이에 새로운 스트로크들이 나타남으로써, 스트로크들이 프레임간 유사성을 유지함을 볼 수 있다.

표 1에서는 스트로크에 기반한 대표적인 렌더링 기법 중 Praun의 실시간 해칭 기법과 Meier의 회화적 스트로크 렌더링 기법, 그리고 본 연구에서 제시하고 있는 기법들의 특징 및 장단점의 비교하였다. Praun의 실시간 해칭 기법에서도 물론 물체의 Zoom in/out 시에 프레임간 유사성을 유지하고 있지만 Praun의 해칭 기법은 개개의 스트로크에 색, 폭의 속성을 줄 수 없는 단점이 있다. 표 1에서 보면 알

수 있듯이 본 연구에서 제시한 폭이 있는 곡선의 긴 스트로크를 적용한 렌더링 방법은 모든 경우에 대해서 프레임간 유사성을 유지한다는 데에 의의가 있다. 또한 개개의 스트로크마다 색, 폭, 길이와 같은 특성을 정의해 줄 수 있으며 스트로크를 주방향으로 렌더링함으로써 3차원 물체의 정보를 보다 효과적으로 전달할 수 있다.

7. 결론 및 향후 연구

본 연구에서는 회화적 렌더링을 위해 3차원 다각형 모델을 기반으로 사용하였다. 스트로크의 위치는 3차원 물체 표면에 고정시켰으며 스트로크의 방향은 물체의 굴곡을 따라서 진행하도록 하였다. 또한, 스트로크의 길이도 마찬가지로 그 굴곡에 따라 스트로크들이 자연스러운 곡선을 이루도록 결정하였고, 스트로크의 폭은 스트로크의 분포에 따라 표면을 모두 덮을 수 있는 폭으로 결정하였다. 본 연구에서는 스트로크 기반 회화적 렌더링 방법에서 물체의 애니메이션시에 프레임간의 유사성을 유지하기 위한 해결

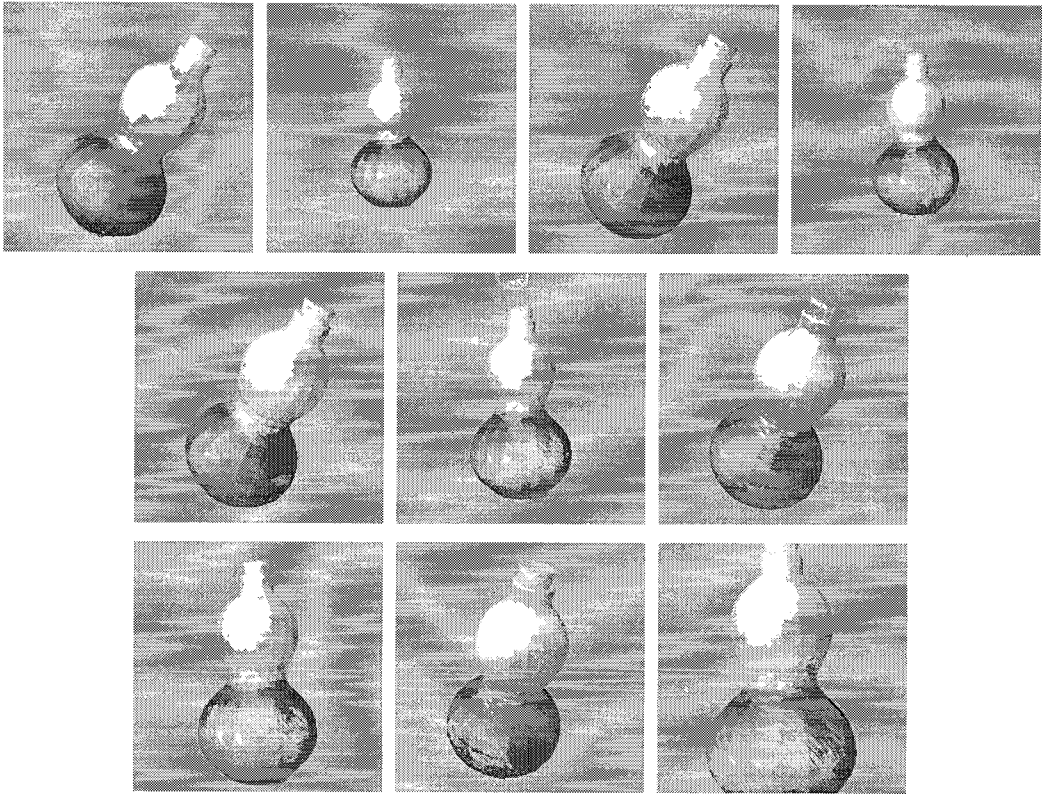


그림 13. 프레임간 유사성 유지 결과

책으로 3차원 파티클 시스템과 동적 스트로크 관리 기법을 제시하였다. 제안된 방법을 사용하면 카메라의 Zoom in/out 시에 브러쉬 스트로크를 동적으로 관리하여 화면 공간상에서는 그 크기가 일정하게 유지되도록 할 수 있다. 이러한 결과로 영상의 튀는 현상(popping effect)을 제거하여 자연스러운 애니메이션 영상을 생성할 수 있다.

하지만 본 연구에서 제시한 방법은 이러한 장점이

외에 해결해야할 문제가 남아있다. 첫 번째 문제는 시스템의 수행 속도에 있다. 현재의 시스템은 하나의 물체에 대해서 수많은 스트로크들을 생성한다. 이는 스트로크 렌더링 시에 빈 공간이 생기는 것을 방지하기 위한 것으로써 시스템 수행 과정 중에 스트로크를 생성하는 부분이 제일 많은 수행 시간을 갖는 결과를 낳고 있다. 이 문제의 해결 방법으로는 스트로크의 분포를 고르게 하는 방법이 있다[14]. 스트로크의 분

표 1. 기존 방법과 제안된 방법과의 비교

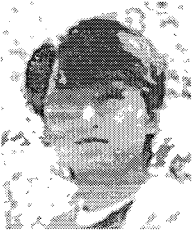
	Praun	Meier	제안 방법
스트로크 특성	펜화, 점묘화 단색의 스트로크	개개의 스트로크가 색상을 표현	
스트로크 렌더링	스트로크 집합의 텍스처 사용	하나의 텍스처 스트로크 사용	긴 곡선의 스트로크 생성
스트로크 방향	사용자 지정	사용자 지정	물체의 주방향
프레임간 유사성 (이동/회전)	○	○	○
프레임간 유사성 (Zoom in/out)	○	×	○
실시간 처리	○	×	×

포를 일정 간격으로 고르게 함으로써 빈 공간 생기는 것을 방지함과 동시에 스트로크의 수를 최적화할 수 있다.

또한, 본 연구에서 더 나아가 연구되어질 부분은 유화적 기법의 적용 부분이다. 현재의 스트로크 렌더링 결과는 스트로크에 질감이나 유화적 색조의 특성을 보이지 않는다. 따라서 보다 유화적인 방향으로 결과를 유도해 내기 위해 물감의 질감을 나타내는 임파스토(Impasto) 기법[15]이나 글레이징(Glazing) 기법과 같은 연구들이 진행되어야 하고, 스트로크 색조의 결정을 유화적 색조로 변환하기 위한 연구들이 더 필요하겠다.

참 고 문 헌

- [1] P. Haeberli, "Paint By Numbers: Abstract Image Representations", in SIGGRAPH '90 Conference Proceedings, August 1990, pages 207-214.
- [2] G. Winkenbach and D. H. Salesin, "Computer-Generated Pen-and-Ink Illustration", in Proceedings of SIGGRAPH '94, July 24-29, 1994, pages 91-100.
- [3] M. P. Salisbury, S. E. Anderson, R. Barzel, and D. H. Salesin, "Interactive Pen-and-Ink Illustration", in Proceedings of SIGGRAPH '94, July 24-29, 1994, pages 101-108.
- [4] Aaron Hertzmann and Denis Zorin. "Illustrating smooth surfaces", Proceedings of SIGGRAPH 2000, pages 517-526, July 2000.
- [5] Aaron Hertzmann. "Paint by relaxation", In Computer Graphics International 2001, pages 47-54, July 2001. ISBN 0-7695-1007-8.
- [6] Aaron Hertzmann and Ken Perlin. "Painterly Rendering for Video and Interaction", In Proceedings of the First Annual Symposium on Non-Photorealistic Animation and Rendering, June 2000.
- [7] E. Praun, H. Hoppe, M. Webb, and A. Finkelstein. "Real-time hatching", In SIGGRAPH '01 Conference Proceedings, August 2001.
- [8] G. Gorla, V. Interrante, and G. Sapiro, "Texture Synthesis for 3D Shape Representation", IEEE Transactions on Visualization and Computer Graphics, 2002.
- [9] Barbara J. Meier, "Painterly Rendering for Animation", In SIGGRAPH 96 Conference Proceedings, pages 477-484, August 1996.
- [10] Victoria Interrante and Chester Grosch, "Visualizing 3D Flow", IEEE Computer Graphics and Applications, 18 (4):49-53, 199.
- [11] Ahna Girshick, Victoria Interrante, Steven Haker, and Todd Lemoine. "Line direction matters: An argument for the use of principal directions in 3d line drawings", In NPAR 2000 : First International Symposium on Non-Photorealistic Animation and Rendering, pages 43-52. ACM SIGGRAPH / Eurographics, June 2000.
- [12] Aaron Hertzmann. "Painterly Rendering with Curved Brush Strokes of Multiple Sizes", In SIGGRAPH 98 Conference Proceedings, pages 453-460, July 1998.
- [13] J. Goldfeather and V. Interrante, "Understanding Errors in Approximating Principal Direction Vectors", ACM Transactions on Graphics, 2003.
- [14] B. Jobard and W. Lefer, "Creating Evenly-Spaced Streamlines of Arbitrary Density", in Proceedings of the 8th Eurographics Workshop on Visualization in Scientific Computing, 1997, pages 45-55.
- [15] Aaron Hertzmann. "Fast Paint Texture", In NPAR 2002: Proceedings of the Second Annual Symposium on Non-Photorealistic Animation and Rendering, June 2002.
- [16] Bruce Gooch, Peter-Pike Sloan, Any Gooch, Peter Shirley and Richard Riesenfeld. "Interactive Technical Illustration", Proceedings of ACM Symposium on Interactive 3D Graphics, 1999.



이 호 근

2001년 2월 중앙대학교 컴퓨터 공학과 공학사
2003년 2월 중앙대학교 영상공학과 컴퓨터그래픽스 전공 공학석사
2003년 3월~현재 삼성전자 TN 총괄 무선 사업부 연구원

관심분야 : Non-Photorealistic Rendering



윤 경 현

1991년 3월 University of Connecticut CS&E 박사
1983년 9월~1985년 8월 한국전기연구소 연구원
1991년 9월~현재 중앙대학교 컴퓨터 공학부 교수

관심분야 : Rendering, Global Illumination, Image-Based Rendering, Non-Photorealistic Rendering



류 승 택

1996년 2월 중앙대학교 전자계산학과 공학사
1998년 2월 중앙대학교 컴퓨터 공학과 컴퓨터 그래픽스 전공 공학석사
2002년 8월 중앙대학교 영상공학과 공학박사

2002년 9월 중앙대학교 정보통신 연구원 연구 전담 교수
2004년 3월~현재 한신대학교 소프트웨어학과 조교수
관심분야 : Image Based Rendering, Realtime Rendering, NPR