



HDR 변환함수를 이용한 효과적인 과학 데이터 가시화

(Effective Visualization of Scientific Dataset using HDR Transfer Function)

이 중 연 (jylee@kisti.re.kr)

한국과학기술정보연구원
Korea Institute of Science & Technology Information

목차

1. 서론	1
2. HDR 과학 데이터	3
3. 변환 함수	7
가. 개요	7
나. 고차원 변환 함수	8
4. 기존 연구	10
5. HDR 변환 함수	18
가. 비선형 히스토그램 생성	19
나. HDR 변환 함수 생성	21
다. HDR 변환 함수 구현 및 적용	22
6. 결론	24
7. 참고문헌	25

그림 차례

[그림 2-1] Utah Torso 데이터	3
[그림 2-2] 테라 스케일 로터 시뮬레이션 데이터	4
[그림 2-3] 변환함수 에디터	5
[그림 3-1] 복셀 값 변화도와 1차, 2차 미분 값 변화도	8
[그림 3-2] 다차원 변환 함수 에디터와 렌더링 사례	9
[그림 4-1] 높은 동적 범위 이미지 톤 매핑	11
[그림 4-2] 다양한 톤 매핑 연산의 적용	12
[그림 4-3] Focus+context 변환 함수 인터페이스	13
[그림 4-4] Range Mapping	14
[그림 4-5] Range Mapping을 적용한 가시화 예	14
[그림 4-6] 변환 함수 블렌딩	16
[그림 4-7] 시변환 데이터에 대한 동적 변환함수 생성	17
[그림 5-1] HDR 변환 함수 생성 시스템	18
[그림 5-2] 히스토그램 재분배	19
[그림 5-3] 비선형 히스토그램 생성 방법	20
[그림 5-4] 비선형 히스토그램 생성 순서도	20
[그림 5-5] HDR 변환 함수 생성 순서도	22
[그림 5-6] 변환 함수 적용 결과	23

1. 서론

실제 세계에 존재하는 색과 명암은 매우 넓은 동적 범위를 가진다. 여기서 동적 범위(dynamic range)란 어떤 장면이나 이미지의 가장 큰 값과 가장 작은 값의 비율로 정의된다. 인간의 눈은 약 1,000,000:1이라는 매우 폭 넓은 동적 범위를 인지한다. 그러나 현재 디스플레이 기술로는 이러한 큰 동적 범위를 재현하는 것은 매우 어렵다[1]. 일반적인 LCD 디스플레이는 1,000:1 정도의 동적 범위만을 제공할 뿐만 아니라 채널당 8비트만을 지원하는 일반적인 그래픽스 하드웨어로는 이러한 동적 범위를 지원하지 못한다. 최근 50,000:1 정도의 높은 동적 범위를 지원하는 디스플레이가 출시되고 있지만 상대적으로 고가일 뿐만 아니라 현존하는 어떠한 디스플레이도 실제 세계의 높은 동적 범위를 지원하지는 못한다.

이러한 문제를 해결하기 위해서 탄생한 기술이 높은 동적 범위 이미징(high dynamic range imaging) 기술이다. Dubevec과 Malik은 [2]에서 여러 장의 서로 다른 노출을 준 낮은 동적 범위 이미지를 이용, HDR 휘도 맵을 생성하는 기술을 개발, 높은 동적 범위 이미징 기술을 처음으로 제안했다. 이러한 높은 동적 범위 이미징 기술은 컴퓨터 렌더링 분야에도 적용되기 시작했는데, Dubevec은 [3]에서 합성된 오브젝트를 실제 장면에 넣는 논문을 발표한 이후로 높은 동적 범위 렌더링 기술은 영화는 물론 컴퓨터 게임 등 여러 분야에 널리 활용되고 있다.

높은 동적 범위를 가지는 이미지를 낮은 동적 범위의 디스플레이에서 보이기 위해서는 톤 매핑(tone mapping)이 이루어져야 한다. 톤 매핑이란 높은 동적 범위의 색깔 값들을 컴퓨터 디스플레이에서 표현 가능한 낮은 동적 범위로 매핑하는 기법을 말한다. 물론 디스플레이 이전의 과정은 모두 높은 동적 범위로 계산된다. 일반적으로 톤 매핑은 비선형적으로 이루어지는데, 전체 색깔 범위에서 많이 나타나는 부분을 집중적으로 강조하는 방식이다. 일반적으로 톤 매핑은 크게 두 종류로 나눌 수 있다. 한 가지는 전역 톤 연산(global tone operator)이고 다른 하나는 지역 톤 연산(local tone operator)이다. 전역 톤 연산은 영상의 모든 픽셀의 색에 동일한 변환 함수를 적용하는 것이다[4,5,6,7]. 이때 변환 함수는 일반적으로 비선형적이 된다. 지역 톤 연산은 영상의 각 영역별로 서로 다른 변환 함수를 적용하는 것이다. 이 방법은 많은 경우 한 픽셀의 색깔은 주변 픽셀의 색깔과 거의 비슷하다는 사실에 기초해서 만들어진 방법이다[8,9,10,11]. 최근 그래픽스 하드웨어의 성능이 발달하면서 여러 종류의 톤 매핑 알고리즘이 하드웨어로 구현되고 있다. 이를 통해 실시간으로 높은 동적 범위 렌더링을 수행하는

것이 가능해지고 있다.

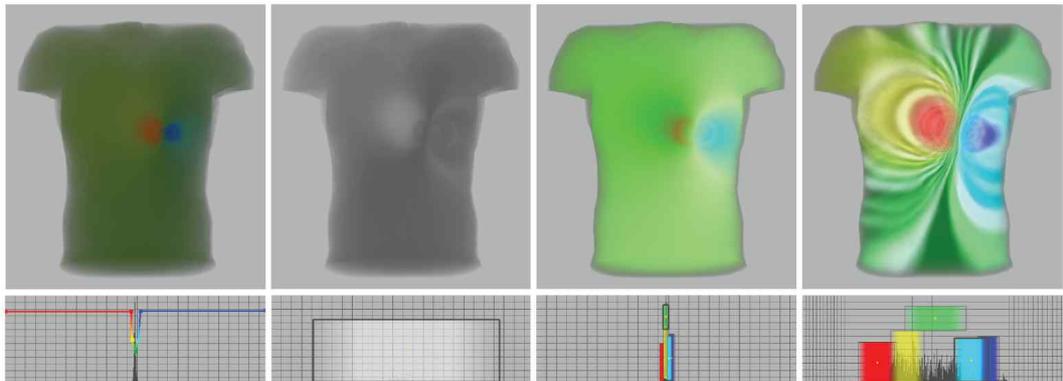
한편, 최근 들어 슈퍼컴퓨터를 비롯한 계산 자원의 성능이 비약적으로 발전하여 이를 통한 과학 데이터의 크기 또한 급격히 커지고 있다. 이로 인해 새로 생산되는 과학 데이터의 상당량은 기존 데이터에 비해 더 높은 동적 범위를 가지는 경우가 많다. 특히, 데이터가 시변환일 경우 전체 시간에 걸쳐 수치 값들이 지속적으로 변하는 경우가 많기 때문에 더욱 높은 동적 범위를 가지게 된다.

본 기술 문서에서는 이러한 높은 동적 범위 가시화 기술을 이용, 높은 동적 범위의 과학 데이터를 가시화하는 방법에 대해 살펴본다.

2. HDR 과학 데이터

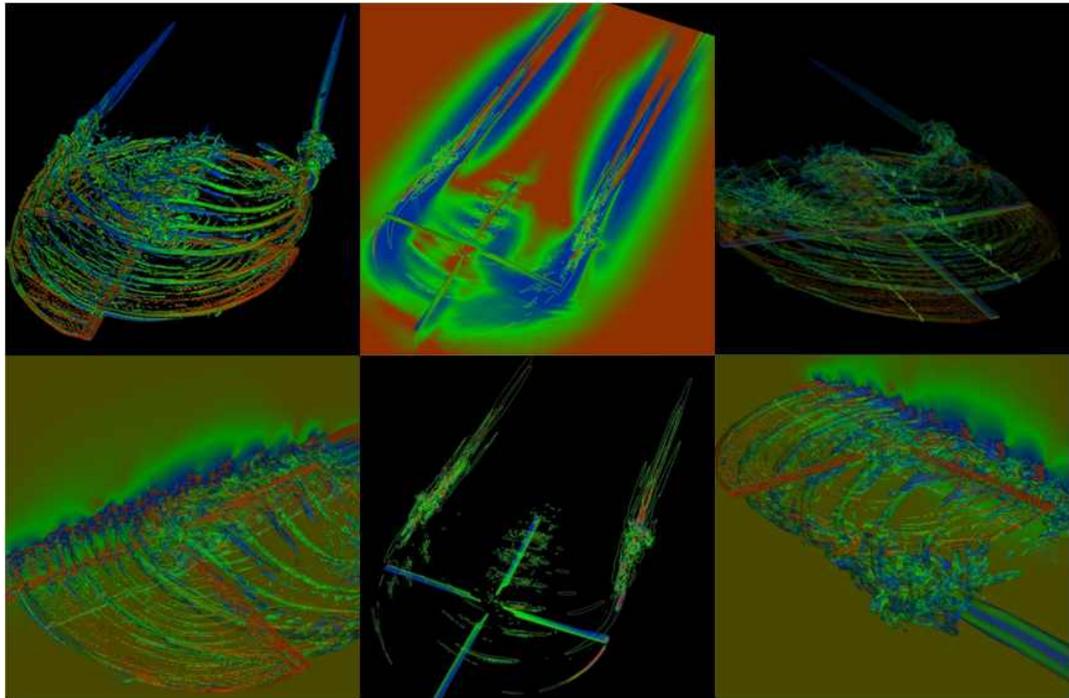
많은 종류의 과학 수치 데이터들은 일반적으로 높은 동적 범위를 가지고 있다. 수치계산을 통해 생산된 대부분의 과학 수치 데이터들은 32비트의 배정도 소수점으로 구성되어있다. 이러한 데이터의 전체 동적 범위를 가시화하기 위해서는 2^{32} 의 해상도를 가지는 고해상도의 히스토그램 또는 변환 함수가 필요하다. 그러나 일반적인 그래픽스 하드웨어의 프레임버퍼는 16비트로 구성되어 있으므로 이러한 고해상도의 히스토그램 또는 변환 함수의 적용이 불가능할 뿐만 아니라 이를 그래픽스 하드웨어의 텍스처 메모리로 저장하여 하드웨어 볼륨 렌더링을 시도할 경우 32비트의 데이터를 8비트로 줄여야 하므로 심각하게 품질이 저하되게 된다.

또, 높은 동적 범위의 과학 데이터의 경우 전체 값들 중 높은 비율이 전체 범위 중 좁은 범위에 집중되어 있는 경우가 많다. 예를 들어 Utah Torso 데이터의 경우 총 360 스텝, 스텝 당 5만개의 사면체(tetrahedron)로 구성되어있는데, 각 타입 스텝별로 봤을 때 전체 데이터 중 83%가 1%이 스칼라 범위에 들만큼 많은 부분이 좁은 범위에 집중되어 있다[12]. 그림 2-1은 Utah Torso 데이터의 가시화 예인데, 윗 부분은 실제 볼륨 렌더링 결과이고 아래 부분은 히스토그램과 변환 함수 에디터이다. 첫 번째 그림을 보면 히스토그램의 중간부분에 데이터가 집중되어 있는 것을 알 수 있다.



[그림 2-1] Utah Torso 데이터

마찬가지로 우리 연구원의 슈퍼컴퓨터를 이용해서 생성한 로터 시뮬레이션 데이터의 경우 총 98스텝, 스텝 당 140만개의 육면체(hexahedron)로 구성되어있는데, 전체 타입 스텝을 걸쳐 약 80%가 1%의 스칼라 범위에 든다.



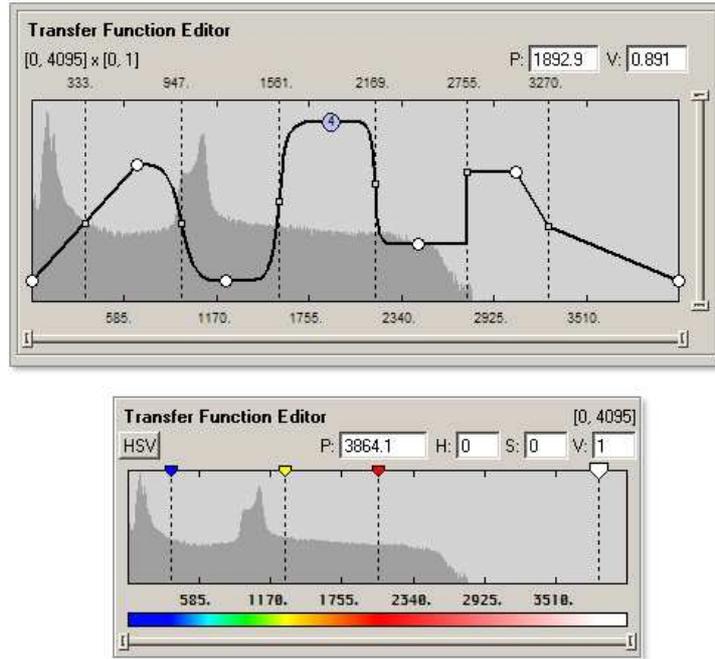
[그림 2-2] 테라 스케일 로터 시뮬레이션 데이터

이와 같이 좁은 범위에 데이터가 집중되어 있는 데이터의 경우 균일하게 해상도를 낮춘 저해상도 변환함수를 적용하면 좁은 범위의 정보가 손실될 수 있다. 이러한 문제를 해결하기 위해서는 크게 두 가지 방법이 존재하는데, 첫 번째는 고해상도의 변환함수를 사용하는 것이고 두 번째는 많은 데이터가 집중된 범위를 강조하는 변환함수를 생성하는 것이다.

여기서 첫 번째 고해상도 변환함수를 사용하는 방법은 고가의 컴퓨터 그래픽스 하드웨어와 고가의 높은 동적 범위를 지원하는 디스플레이가 필요하다. 또한 고해상도의 변환 함수는 해상도가 너무 커서 한 화면에 다 들어오지 않는 경우가 많아 조작하기가 불편한 문제가 있다. 마지막으로 아무리 높은 해상도의 변환함수를 사용한다 하더라도 그 보다 큰 동적범위의 데이터는 존재할 수 있고 이러한 데이터를 가시화할 때는 마찬가지로 문제가 발생한다.

따라서 이러한 문제를 해결하기 위해 데이터가 집중된 범위를 강조하는 변환함수를 생성할 필요가 있다. 종래의 기술에서는 데이터의 특정 영역을 강조하기 위한 방법으로 사용자의 상호작용을 받는 변환함수 에디터를 사용했다. 이러한 에디터를 이용해서 사용자는 중요한 정보가 포함된 영역을 직접 확장하여 변환 함수를 생성하도록 했다[13]. 그러나 이러한 사용자의 입력에 따른 변환 함수 생성 기법은 일반적으로 여러 번의 시행착오를 통해 최종 변환 함수를 생성하는 것으로 많은 시간을 필요로 한다. 특히, 데이터의 크기가 거대한 경우에는 더욱 많은

시간을 필요로 하는 문제가 있다. 또, 전체 값들 중 높은 비율이 전체 범위 중 좁은 범위에 집중되어있는 경우에 변환 함수 에디터의 경우에도 마찬가지로 매우 좁은 영역에서 위젯 등을 컨트롤해야 하는데, 이는 사용자에게 매우 불편한 환경이라는 문제도 존재한다.



[그림 2-3] 변환함수 에디터

한편, 슈퍼컴퓨터를 비롯한 계산 자원의 성능이 비약적으로 발전하여 이를 통한 과학 데이터의 크기 또한 급격히 커지고 있다. NCSA의 테라 그리드 클러스터를 통해 생산된 Piecewise-Parabolic Method(PPM) 가스 동적 코드는 2048³의 크기로 약 86억개의 복셀로 구성되어 있다. 이러한 거대 데이터를 대상으로 하드웨어 불륨 렌더링을 수행할 때에는 최소 2048개의 슬라이스를 컴포지션해야 하고 영상 품질을 높이기 위해서는 4096개의 슬라이스를 컴포지션해야 한다. 즉, 데이터의 크기가 커지면 커질수록 컴포지션해야 하는 슬라이스의 개수가 늘어나는데, 이럴 경우 컴포지션에 사용하는 알파 브렌딩(alpha blending)연산에서 각 슬라이스의 알파값이 0에 가까워지게 되면서 정밀도 문제가 발생하게 된다. 따라서 이러한 문제를 해결하기 위해서는 고정밀도 블렌딩 하드웨어를 필요로 한다.

이 장에서 설명한 높은 동적 범위의 수치 데이터를 가시화하기 위한 문제점들을 정리하면 다음과 같다.

-
1. 일반적인 컴퓨터 그래픽스 하드웨어는 높은 동적 범위의 수치 데이터를 가시화하기 위한 고해상도 변환함수를 지원하지 못한다.
 2. 좁은 영역에 수치가 집중된 데이터의 경우 균등하게 해상도를 낮춘 저해상도 변환함수를 사용하면 중요한 정보의 손실이 있을 수 있다.
 3. 데이터의 중요한 영역을 강조하기 위한 사용자 인터페이스는 매우 불편하다.
 4. 고해상도의 볼륨 데이터를 그래픽스 하드웨어를 이용해서 볼륨 렌더링할 경우 고정밀도 블렌딩이 지원되어야 한다.

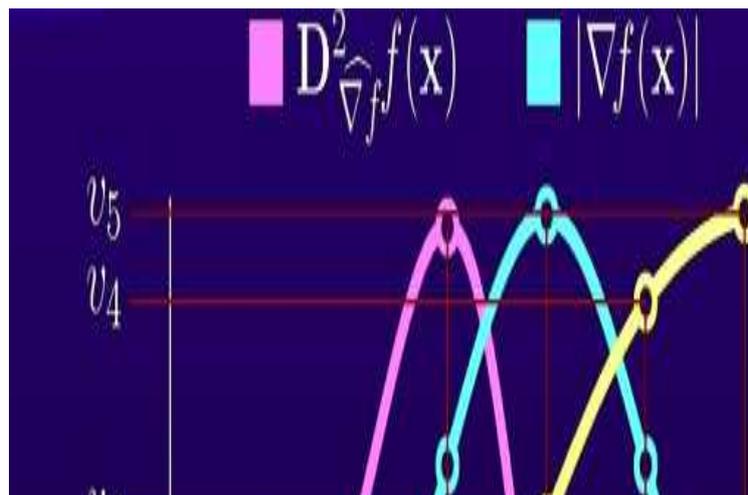
3. 변환 함수

가. 개요

변환 함수란 수치 데이터에서 사용자가 가시화하기를 원하는 영역을 분리해내는 역할을 하는 함수로, 과학 데이터 가시화에서 중요한 위치를 차지하고 있다. 대표적인 과학 데이터 중 하나인 볼륨 데이터는 각 복셀마다 밀도, 온도, 기압, 속도 등 다양한 의미를 가지는 변수들을 포함하고 있다. 볼륨 데이터의 전체 영역에 걸쳐 이러한 변수들이 조밀하게 밀집해 있고, 이 중 많은 수의 변수들이 특별한 의미를 가지고 있지 못하기 때문에, 전체 복셀들 중 유의미한 복셀을 찾아서 해당하는 변수를 사람이 쉽게 인지하도록 가시화하는 것은 어려운 문제이다. 예를 들어, 강우량이 100ml를 초과하는 영역을 가시화하고자 할 때, 강수량이 100 ml 미만인 지역은 이 시나리오에서는 특별한 의미를 가지지 못하기 때문에 가시화할 때 표시하지 않아야 한다. 이와 같이 복셀의 여러 변수들 중 유의미한 부분을 뽑아서 사람이 쉽게 정보를 인지할 수 있도록 색깔 및 투명도로 전이(transfer)하는 함수(function)를 변환 함수라고 하며, 이 변환 함수를 어떻게 생성했느냐가 데이터의 관심 영역을 충분히 잘 가시화했는지 여부가 결정될 만큼 과학 데이터 가시화에서 매우 중요한 위치를 차지한다. 좋은 변환 함수를 생성하는 것은 매우 어렵고 지루한 작업이다. 여기에는 여러 가지 이유가 있는데, 변환 함수 도메인에서 관심 영역을 유일하게 정의하는 것은 매우 어렵다는 사실이 첫 번째 이유이다. 관심 영역이 공간 도메인(spatial domain)에서 쉽게 정의된다 하더라도 변환 함수 도메인에서 feature가 되는 관심영역을 이것과 같은 데이터 범위를 가지는 관심이 없는 영역과 분리시키기는 쉽지 않다. 두 번째 이유는 변환 함수는 매우 다양한 자유도(degrees of freedom)를 가질 수 있다는 사실이다. 꺾이는 직선 모양의 간단한 1D 변환 함수조차도 2자유도를 필요로 한다. 세 번째 이유는 일반적으로 대부분의 유저 인터페이스에서는 사용자가 변환 함수를 생성할 때, 쉽게 관심 영역을 설정하고 색깔 및 투명도를 설정할 수 있도록 하는 데이터 자체를 바탕으로 한 정보를 제공하지 않는다는 것이다. 이러한 정보가 없으면 사용자는 여러 번에 걸쳐 일단 한번 변환 함수를 생성해보고 문제가 있으면 이를 지속적으로 개선하는 방법으로 변환 함수를 제작해야 한다. 이러한 변환 함수 제작 방법은 사용자들을 쉽게 좌절하게 만드는데, 변환 함수를 약간만 수정해도 종종 완전히 다른 가시화 결과가 나오기 때문에 변환 함수를 점진적으로 개선한다는 것이 쉽지 않은 작업이기 때문이다.

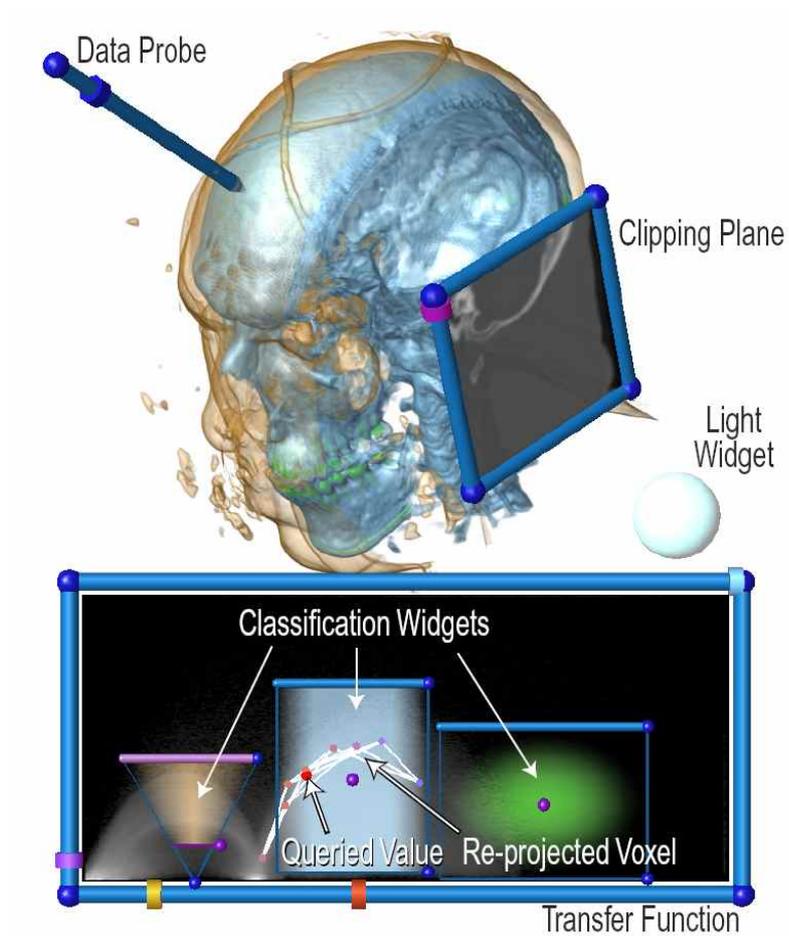
나. 고차원 변환 함수

하나의 복셀 값만으로 1차원 변환 함수를 생성해서 관심영역을 비관심영역과 분리하는 것은 매우 어려운 일이다. 이러한 이유로 관심영역과 비관심영역을 쉽게 분리시키는 변환 함수를 생성하는 연구가 여러 연구자들에 의해 행해지고 있는데, 그 대표적인 방법으로 2D 또는 3D의 고차원 변환 함수를 생성하는 방법이 있다. 즉, 하나의 복셀 값만으로 색깔 및 투명도를 결정하는 것이 아니라, 복셀 값을 1차 미분한 그라디언트(gradient) 값이나, 2차 미분한 값, 또는 다른 복셀 값들의 조합으로 색깔 및 투명도를 결정하는 것이다. 이렇게 여러 값들을 이용해서 변환 함수를 생성하면 데이터의 구조를 보다 세분할 수 있어 비교적 용이하게 관심영역과 비관심영역을 분리해서 가시화할 수 있다. 즉, 이 값들이 다차원 변환 함수의 각 축이 되는 것이다. 특히 1차 미분한 그라디언트 값이나 2차 미분한 값을 변환 함수의 도메인으로 사용하면 복셀 값이 급격히 변하는 물질의 경계부분이나 그다지 변함이 없는 균일(homogeneous) 부분을 쉽게 찾을 수 있다. 그러나 변환 함수의 차원을 높이면 높일수록 변환 함수의 자유도가 높아져서 점진적으로 개선하는 방식으로 변환 함수를 생성할 경우, 생성 시간이 기하급수적으로 늘어난다는 단점이 있다.



[그림 3-1] 복셀 값 변화도와 1차, 2차 미분 값 변화도

그림 3-2는 인체영상 볼륨 데이터에 대한 2차원 변환 함수의 적용 예이다[24]. 위쪽 그림은 렌더링 결과와 여러 종류의 인터페이스이고 아래쪽은 변환 함수 인터페이스이다.



[그림 3-2] 다차원 변환 함수 에디터와 렌더링 사례

변환 함수 인터페이스를 보면 삼각형과 사각형 등 여러 종류의 위젯이 존재한다. 4각형으로 가시화한 하늘 색 및 녹색 부분은 실제 렌더링 결과에 보면 불투명하게 보임을 알 수 있다. 삼각형으로 가시화한 부분은 실제 인체의 피부 부분으로 그라디언트가 높은 부분을 강조하는 역삼각형 모양이기 때문에 렌더링 시 피부와 다른 매질과의 경계 부분이 강조됨을 알 수 있다.

4. 기존 연구

2장에서 설명한 높은 동적 범위의 과학데이터를 가시화하기 위해 여러 가지 방법이 제안되어왔다.

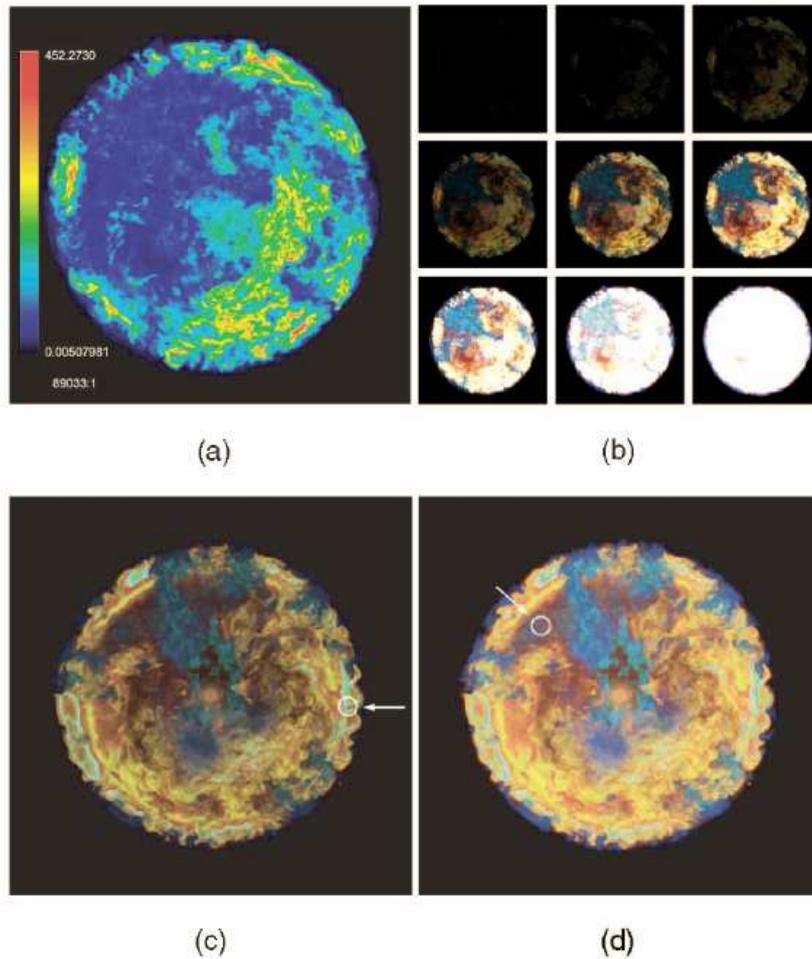
[17]에서 Ghosh 등은 높은 동적 범위 디스플레이 기술을 볼륨 렌더링에 적용했다. 이 논문에서 Ghosh 등은 높은 동적 범위 디스플레이에서 지원 가능한 넓은 영역의 변환 함수를 생성했다. 이 변환 함수는 일반적인 선형 함수로써 일반적인 디스플레이에서는 적용하지 못하는 문제가 있었다.

[18]에서 Kniss 등은 가우시안 변환 함수(Gaussian Transfer Function)를 적용함으로써 높은 동적 범위의 데이터에 대해 저해상도의 색인표(Lookup Table)를 이용해서 정확한 가시화를 가능하게 했다.

또, [14,15,16]에서는 로그 스케일 변환 함수를 이용해서 좁은 영역에 많은 수치가 몰리는 데이터를 가시화하고자 했다. 그러나 이 방법들은 모두 수치가 0에 근접하게 몰려있는 데이터에만 가능한 문제가 있다.

Yuan 등은 높은 동적 범위의 볼륨 데이터에 대해 일반적인 컴퓨터 그래픽스 하드웨어와 디스플레이를 이용해서 가시화하기 위한 방법을 제안했다[19]. 이 방법은 2장에서 소개한 문제들 중 1, 3, 4번의 해결방안을 제시하고 있다. 이 논문에서는 우선 고해상도의 볼륨 데이터를 가시화하기 위해 고정밀도 연산을 지원하는 높은 동적 범위 하드웨어를 사용해서 볼륨 렌더링을 수행한다(4번 문제). 이 논문에서 사용한 고정밀도 연산을 지원하는 그래픽스 하드웨어는 채널당 32비트의 연산을 지원하지만 속도를 빠르게 하기 위해 24비트만을 사용했으며 2048^3 크기의 볼륨 데이터를 성공적으로 렌더링할 수 있었다.

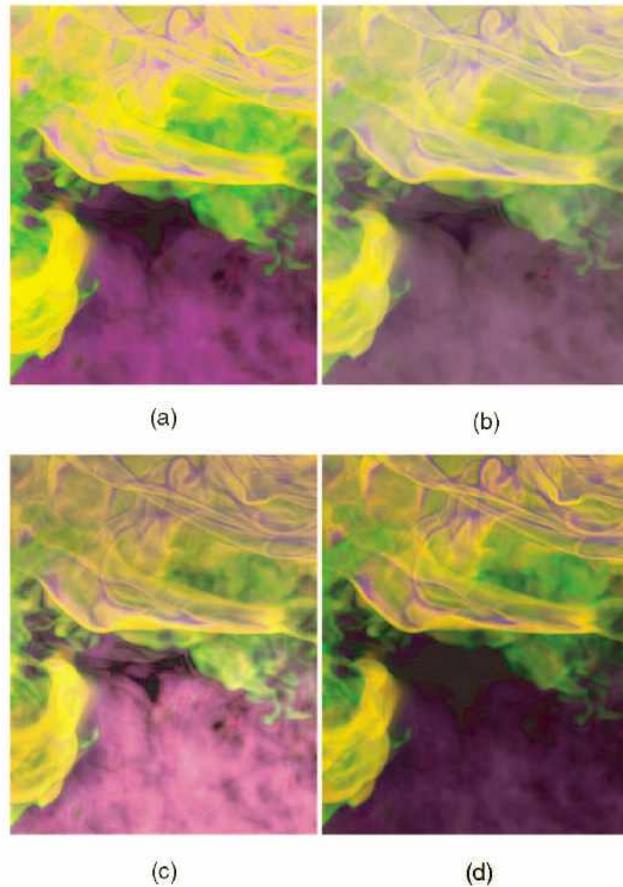
이렇게 렌더링한 이미지는 높은 동적 범위의 이미지로 일반적인 디스플레이에서는 가시화하는 것이 불가능하다. 이를 위해 이 논문에서는 일반 8비트 디스플레이에 가시화하기 위해 기존 높은 동적 범위 이미징에서 사용한 다양한 노출을 준 이미지를 이용, 톤 매핑을 통해 낮은 동적 범위 디스플레이에 가시화하는 방법을 사용했다(1번 문제). 그림 4-1은 높은 동적 범위를 가지는 데이터에 대한 톤 매핑 예이다. (a)는 높은 동적 범위 데이터에 선형 변환 함수를 적용한 예로서 세부적인 정보를 잃어버린 것을 알 수 있다. (b)는 이 데이터에 다양한 노출을 주면서 가시화한 예이고 (c)와 (d)는 특정 복셀값을 중심으로 톤 매핑을 수행한 결과이다. 이 예에서는 adaptive logarithmic 톤 매핑을 사용했다.



[그림 4-1] 높은 동적 범위 이미지 톤 매핑

그림 4-2는 높은 동적 범위 볼륨 렌더링 결과에 다양한 톤 매핑 연산을 수행한 겨로가이다. (a)는 Dargo 등이 제안한 adaptive logarithmic 매핑[20]이고 (b)는 Reihard 등이 제안한 photographic 톤 재생성 방법[21], (c)는 Fattal 등이 제안한 그라디언트 도메인 높은 동적 범위 압축[22], (d)는 Durand와 Dorsey가 제안한 빠른 쌍방 필터링 결과[23]이다.

그림 4-2의 가시화 결과를 보면 밝은 영역에 대해서는 (c)와 (d)가 좋은 대조 (contrast)를 보였으나 (d)는 어두운 영역에서 세부적인 정보를 잃어버리는 것을 알 수 있었다. 이 논문에서는 전반적으로 봤을 때 (c) 그라디언트 도메인 높은 동적 범위 압축 기법의 적용 결과를 가장 좋은 결과로 판단했다.



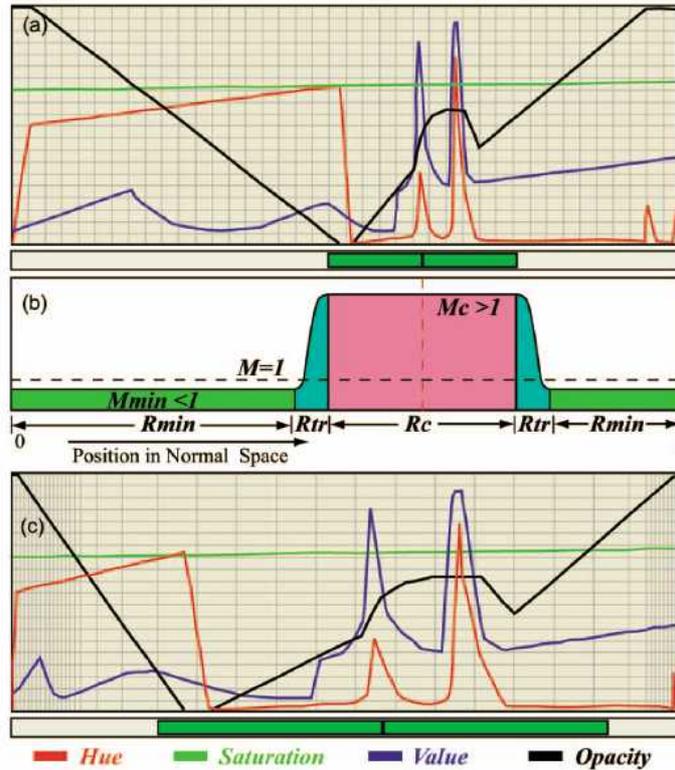
[그림 4-2] 다양한 톤 매핑 연산의 적용

현존하는 대부분의 톤 매핑 기법은 사진을 위한 기법으로 현재까지는 볼륨 렌더링과 같은 과학 데이터 가시화를 위한 기법은 전무하다. 사진을 위한 톤 매핑 기법은 사진을 충실히 인지할 수 있도록 하는 기법인데 비해 과학 데이터 가시화를 위한 톤 매핑 기법은 최대한 세부 사항을 보존해서 가시화해야 한다. 앞으로 이러한 과학 데이터 가시화를 위한 톤 매핑 기법이 지속적으로 연구되어야 할 것이다.

Yuan 등의 논문에서는 1번과 4번 문제뿐만 아니라 3번 문제에 대한 해결방안도 제시했는데, 그림 4-3에서 보는 것과 같이 수치가 집중되는 영역을 확대해서 표시하는 인터페이스를 제공하는 것이다.

이 논문에서는 제안한 인터페이스를 Focus+context 변환 함수 인터페이스라 이름 지었는데, (a)에서 보는 것과 같이 대부분의 정보가 몰려있거나 중요한 부분이라고 판단되는 영역을 아래쪽 초록색 영역 표시와 같이 사용자가 표시하면 (c)와 같이 그 부분을 확대해서 보여주게 된다. 이러한 인터페이스 방법은 중요한 부분을 강조해서 보여줌으로써 사용자가 간편하게 변환 함수 조작을 가능하

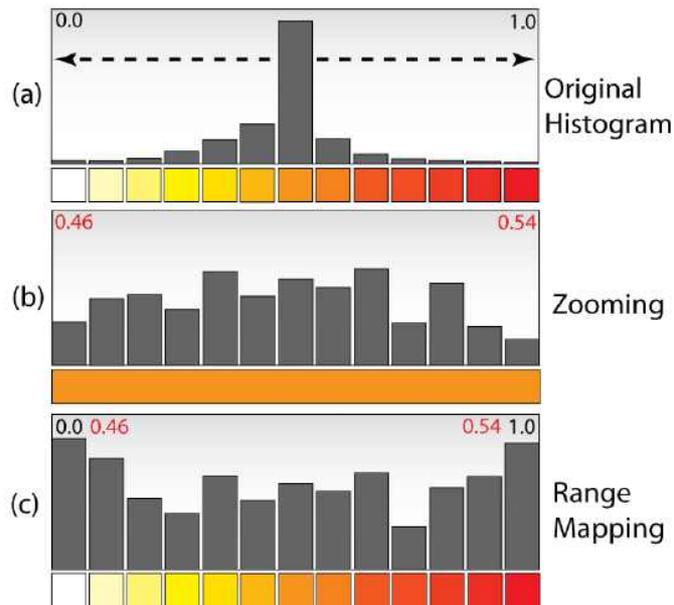
도록 한다. 이 인터페이스는 또한 3장에서 설명한 다차원 변환 함수의 생성 또한 지원한다. 이를 위해 그림 2-1의 아래 부분과 같은 사각형 모양의 변환 함수 위젯 뿐만 아니라 그림 3-2와 같은 삼각형 모양의 변환 함수 역시 지원한다.



[그림 4-3] Focus+context 변환 함수 인터페이스

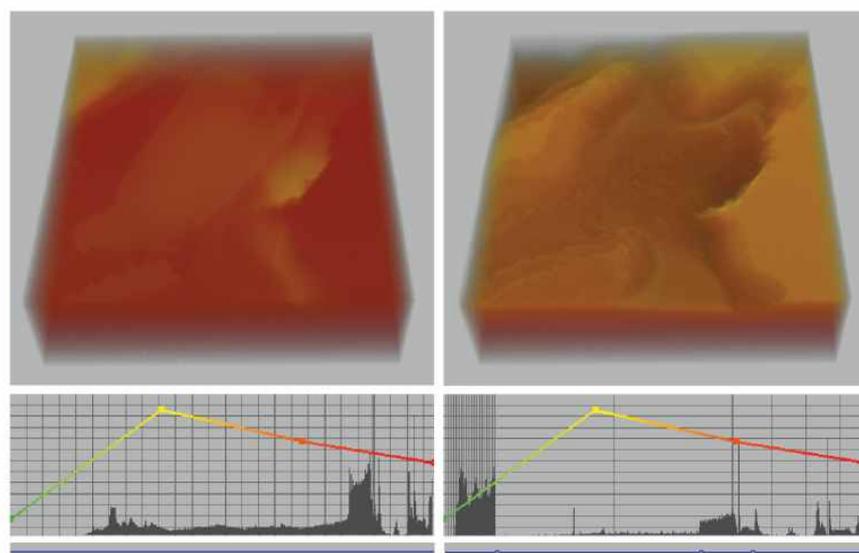
Bernardon 등은 [12]에서 2장에서 제시한 문제점들 중 1, 2, 3번에 대한 해결책을 제시했다. [19]에서 Yuan 등은 Focus+context 변환 함수 인터페이스를 통해 많은 수치가 집중되는 좁은 영역을 확대, 변환 함수를 컨트롤할 수 있는 인터페이스를 제안했다. 그러나 이 방법은 실제 변환 함수 자체는 기존의 선형 함수로써 높은 동적 범위를 보장하는 고정밀도 그래픽스 하드웨어를 사용하는 경우에만 적용이 가능하고, 일반적인 저정밀도 그래픽스 하드웨어에서도 사용이 가능한 저해상도 변환 함수에 적용할 경우 많은 수치가 집중되는 좁은 영역에서 정보를 손실할 가능성이 높다. 이러한 문제를 해결하기 위해 Bernardon 등은 히스토그램의 평탄화를 통해 히스토그램 영역을 재분배하는 Range Mapping 기능을 제안했다. 이 방법은 그림 4-4에서와 같이 (a)의 원 히스토그램에서 많은 수치가 집중되는 [0.46, 0.54] 구간을 늘려서 비선형 히스토그램을 생성한 후 변환 함수에 적용하는 방법으로, 전체 히스토그램의 해상도를 균등하게 낮추지 않고 수치가

집중되는 부분은 살리고 수치가 적은 부분의 해상도를 낮추는 등 불균등한 방법을 통해 저해상도 변환함수에서도 정보의 손실을 최소화할 수 있도록 했다.



[그림 4-4] Range Mapping

그림 4-4에서 (a)는 원 히스토그램이고 (b)는 [12]에서 제안한 확대 인터페이스이며 (c)는 [19]에서 제안한 Range Mapping 기법이다. 그림 4-5는 변환 함수 자체는 변경하지 않은 채 히스토그램의 변경만으로 렌더링 결과가 달라지는 예이다. 왼쪽의 Range Mapping 기법을 적용한 결과의 세부 정보가 더 많이 살아 있음을 알 수 있다.



[그림 4-5] Range Mapping을 적용한 가시화 예

이러한 히스토그램 평탄화 기법은 기존의 사진 영상의 후보정에서 많이 사용되는 기법으로 수학적으로 정규화된 스칼라 입력에 대해 probability density function(PDF)들의 합으로써 cumulative density function(CDF)를 적용함으로써 구현이 가능하다.

$$CDF(x_i) = \sum_{x_j < x_i} PDF(x_j)$$

따라서 정규화된 스칼라 입력 x 에 대해 새로운 출력 y 를 매핑하는 다음과 같은 간단한 함수가 생성된다. 이때, y 는 균등하게 분배되고 정규화된 출력이다.

$$y = CDF(x)$$

일반적인 경우 변환함수를 생성할 때 속도와 간단함 때문에 분리된 (discrete) 히스토그램 평탄화를 사용하고 이를 색인표(lookup table)에 적용하는 방법을 사용한다. 이러한 방법은 자동적이지만 사용자에게 변환함수 생성을 컨트롤할 여지를 주지 않고 각 스칼라 영역간의 연속성이 깨진다는 문제가 있다. 이 논문에서 제안한 Range Mapping 기법은 구분된 선형 매핑 함수(piecewise linear mapping function)를 기반으로 한 방법으로 스칼라 영역의 연속성을 유지하면서 사용자가 컨트롤할 여지를 준다. 입력 스칼라 영역 $[x_0 \dots x_n]$ 에 대해 새로운 스칼라 영역 $[y_0 \dots y_n]$ 을 매핑하는 Range Mapping 함수는 입력 스칼라 영역에 대한 구분된 선형 매핑 함수 f 로써 다음의 조건을 만족한다. : f 는 $f(x_0) = y_0$ 와 $f(x_n) = y_n$ 을 만족하는 단조 증가 함수(monotonically increasing function)이다. 히스토그램 평탄화와 비슷하게 새로운 스칼라 값 y 는 다음과 같이 계산된다.

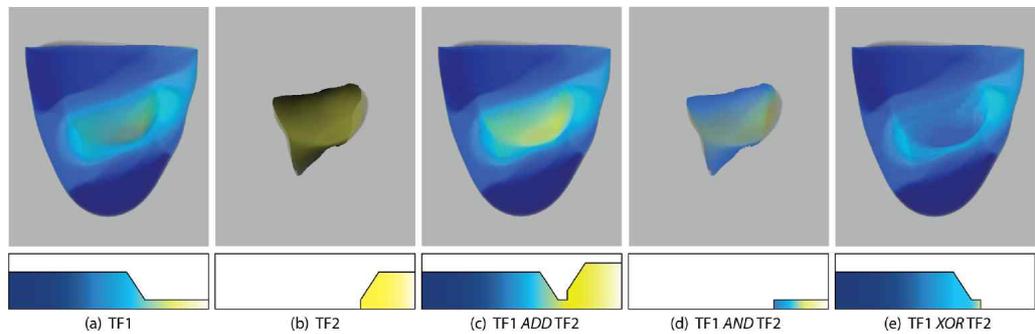
$$y = f(x)$$

실제로 스칼라 영역 $[x_i \dots x_{i+1}]$ 를 $[y_i \dots y_{i+1}]$ 로 매핑하는 선형 Range Mapping 함수로 다음 함수가 많이 사용된다.

$$y = \frac{x - x_i}{x_{i+1} - x_i} (y_{i+1} - y_i) + y_i$$

선형 보간은 매우 간단한 연산이기 때문에 사용자가 영역을 변경할 때마다 실시간으로 바로 계산이 가능하다.

한편, 이 논문에서는 여러 종류의 변환 함수를 블렌딩(blending)하여 새로운 변환 함수를 만드는 기법을 소개했다. 그림 4-6을 보면 (a) TF1와 (b) TF2 변환 함수를 정의한 뒤 TF1 ADD TF2나 TF1 AND TF2 그리고 TF1 XOR TF2 등의 예를 소개했다.

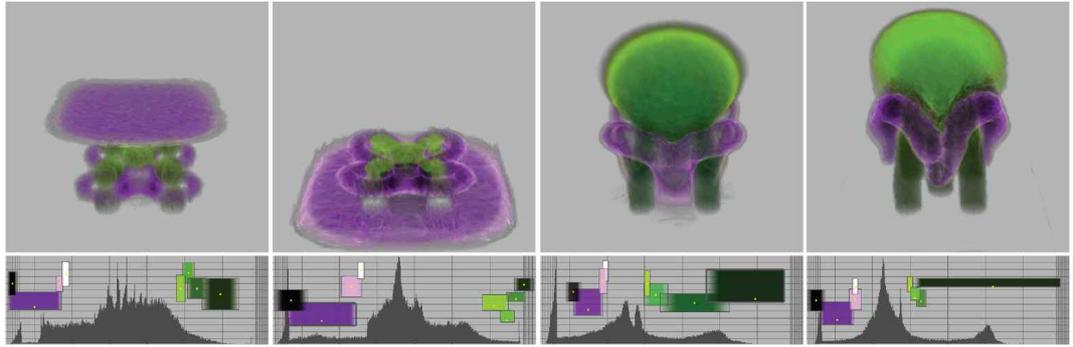


[그림 4-6] 변환 함수 블렌딩

여기서 TF1 ADD TF2는 TF1과 TF2의 가시화를 합치는 효과를 보여주고 TF1 AND TF2는 TF1과 TF2에서 공통적으로 가시화되는 영역을 보여주며 TF1 XOR TF2는 TF1의 가시화 결과 중 TF2와 겹치지 않는 영역만을 보여준다.

이러한 변환 함수 블렌딩 기법을 이용하면 히스토그램이 동적으로 변경되는 시변환 데이터를 위한 시변환 변환 함수의 생성이 가능하다.

그림 4-7의 각 그림은 동일한 데이터에 대한 서로 다른 타임 스텝에서의 히스토그램 변화와 가시화 결과이다. 아래쪽 히스토그램을 보면 시간의 흐름에 따라 데이터의 분포가 동적으로 변경됨을 알 수 있다. 따라서 해당 타임 스텝의 중요한 현상을 가시화하기 위해서는 서로 다른 변환 함수를 적용해야 하는데, 이렇게 되면 매 타임 스텝마다 변환 함수가 변경되어 애니메이션 시 가시화 결과가 끊어져서 보여지게 된다. 이러한 문제를 해결하기 위해 이 논문에서는 여러 타임 스텝 중 키 스텝을 설정하고 변환 함수를 생성한 후 중간 스텝의 경우 앞 뒤 스텝의 변환 함수를 블렌딩해서 보여주는 방법으로 애니메이션 가시화의 끊어짐을 최소화했다.



[그림 4-7] 시변환 데이터에 대한 동적 변환함수 생성

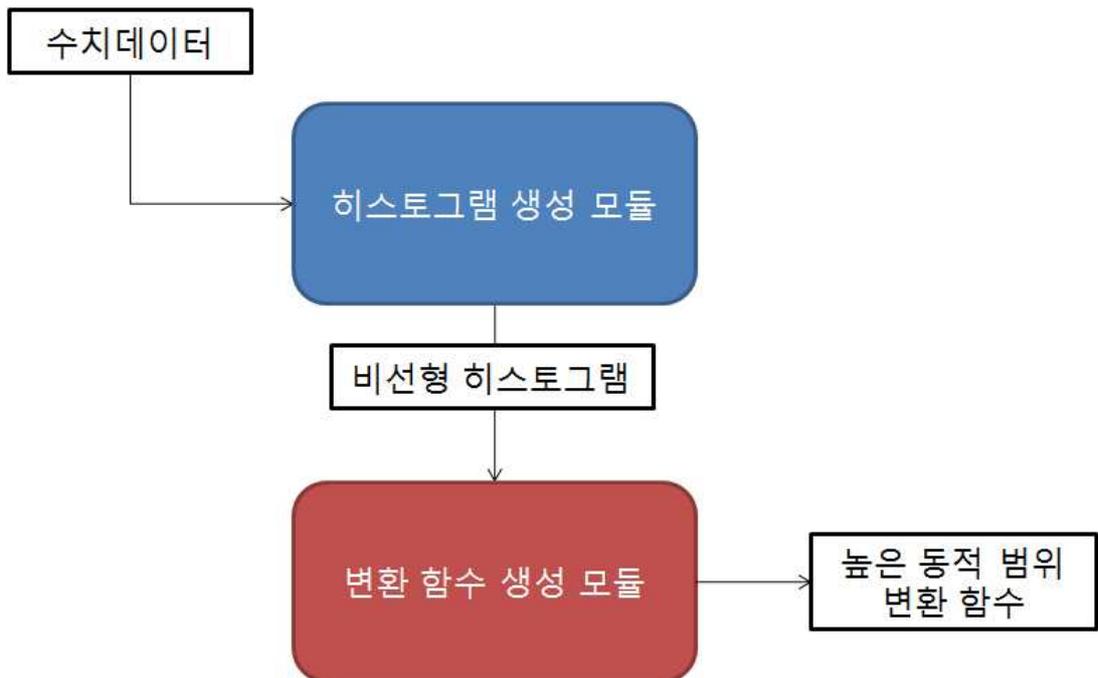
이 장에서 살펴본 기존 연구들은 모두 높은 동적 범위의 수치 데이터를 효과적으로 가시화하기 위한 변환 함수 생성 및 가시화 기법을 소개한 것으로 사진이나 게임 등에서의 높은 동적 범위 데이터 처리 기법들에 비해 연구 사례가 많지 않다. 각 논문들은 높은 동적 범위 수치 데이터를 처리하기 위해 많은 제안을 내놓았지만 아직까지 모든 문제를 완벽히 해결한 방법은 제시하지 못했다.

5. HDR 변환 함수

데이터의 정밀도가 높을수록 수치 값의 범위와 히스토그램의 해상도는 커진다. 8 비트 데이터의 경우 2^8 의 해상도를 가지는 히스토그램이면 전체 데이터 값을 표현하는데 문제가 없다. 그러나 32비트 데이터의 경우에는 2^{32} 의 해상도를 가지는 고해상도의 히스토그램을 필요로 한다. 이러한 히스토그램을 이용해서 생성한 변환 함수 역시 고해상도여야만 함은 물론이다. 그러나 범용 컴퓨터 하드웨어에서는 프레임 버퍼의 정밀도가 16비트로 낮기 때문에 이런 해상도의 변환 함수를 이용해서 렌더링하는 것은 불가능하다.

이를 위해 본 기술문서에서는 높은 동적 범위의 수치 데이터에 대해 비선형 히스토그램 생성과 이를 이용한 변환 함수 생성을 통해 저해상도의 변환 함수로도 품질의 하락 없이 데이터를 가시화할 수 있는 방법을 제안하고자 한다. 이 방법은 높은 동적 범위의 수치 데이터를 위한 변환 함수를 사용자의 개입없이 완전히 자동으로 변환 함수를 생성하는 방법으로 2장에서 언급했던 문제점들 중 1, 2, 3번 문제의 해결 방안이 될 수 있다.

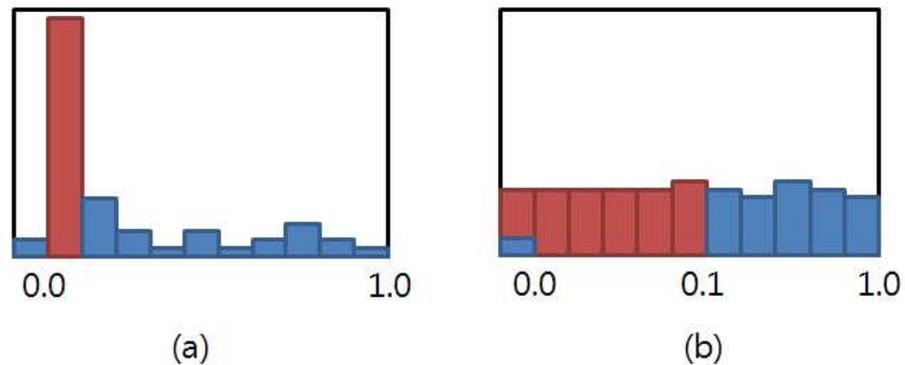
본 기술문서에서 제안하고자 하는 방법의 전체적인 구성도는 그림 5-1과 같다.



[그림 5-1] HDR 변환 함수 생성 시스템

가. 비선형 히스토그램 생성

대부분의 높은 동적 범위 데이터는 데이터 값이 선형으로 분포되지 못하고 특정 데이터 범위에 값들이 집중되는 현상이 있다. 이러한 현상을 이용해서 그림 5-2와 같이 히스토그램의 데이터 값을 재분배해서 데이터 값이 없거나 적은 영역을 압축하면 전체 데이터를 낮은 해상도의 히스토그램으로 표현하는 것이 가능해진다.

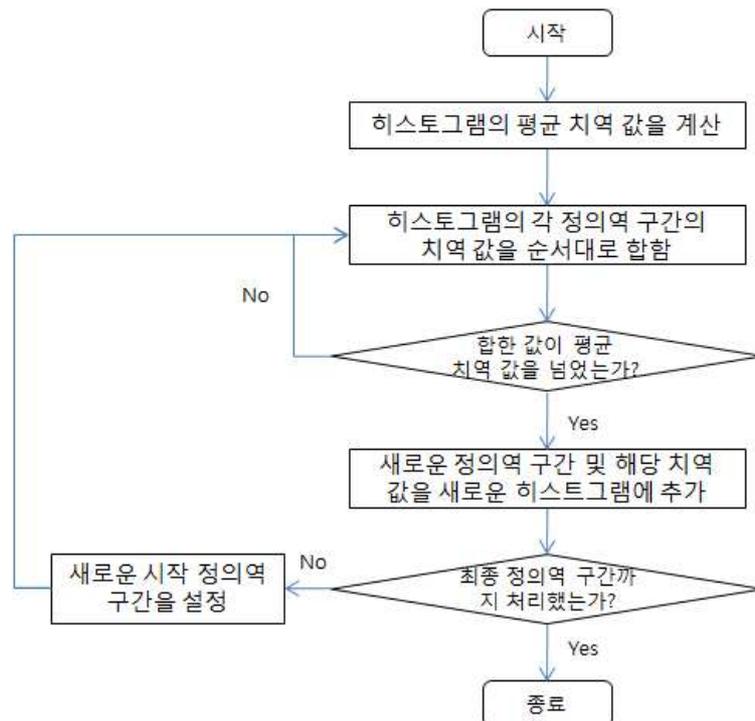


[그림 5-2] 히스토그램 재분배

히스토그램의 데이터 값을 재분배할 때는 그림 5-2의 (b)와 같이 히스토그램의 각 정의역 구간들의 치역 값들이 전 정의역 구간에 걸쳐 균등하게 되도록 해야 한다. 이를 위해서 치역 값이 작은 구간은 합하고 큰 구간은 나누어서 정의역이 선형으로 증가하지 않고 비선형으로 증가하도록 한다. 이때, 치역 값이 큰 구간을 나눌 때는 그 구간이 몇 개의 구간으로 나뉠 것인지를 계산하고 나뉠 구간의 개수에 맞춰 선형 보간법으로 나누면 된다. 치역 값이 작은 구간을 합칠 때는 뒤에 변환 함수에서 복원될 것을 감안해서 전체적인 데이터 값의 분포를 선형, 로그, 지수 함수 중 하나로 분석, 저장한다. 히스토그램의 데이터 값을 재분배해서 비선형 히스토그램을 생성하는 상세한 방법은 그림 5-3과 같고, 이를 위한 순서도는 그림 5-4와 같다.

1. 히스토그램의 평균 치역 값을 구한다. 이 평균 치역 값은 각 정의역 구간의 치역 값을 균등화하는 값이 된다. 이 값을 mean이라 하자.
2. 히스토그램의 각 정의역 구간의 치역 값을 순서대로 합하며 평균 치역 값 mean을 넘는지 확인한다. 시작하는 정의역 구간의 시작 값을 s1, 치역 값의 합을 sum이라 하자.
 - 2-1. $sum < mean$ 이면 계속 2를 실행한다.
 - 2-2. $sum \geq mean$ 이면 3을 실행한다.
3. 2에서 마지막으로 합한 정의역 구간의 시작 값을 es, 종료 값을 ee, 해당 구간의 치역 값을 range라고 하자. 새로운 히스토그램에 치역 값을 mean으로 하는 정의역 구간 $[s1, es+(ee-es)*(sum-mean)/range]$ 을 추가한다. 새로운 히스토그램의 해당 정의역 구간에 es와 시작 구간 부터 마지막 직전 구간까지의 데이터 값 분포 함수를 선형, 로그, 지수 함수 중 하나로 저장한다.
 - 3-1. 3에서 $(sum-mean)/range$ 가 0.1 이하이면 마지막 바로 직전 구간까지만 합해서 새로운 히스토그램에 추가한다.
 - 3-2. 3에서 $(sum-man)/range$ 가 0.9 이상이면 마지막 구간 끝까지 모두 더해서 새로운 히스토그램에 추가한다.
4. 시작하는 정의역 구간의 시작 값을 $es+(ee-es)*(sum-mean)/rang$ e로 한 뒤 전체 히스토그램의 마지막 영역까지 2를 반복해서 수행한다.

[그림 5-3] 비선형 히스토그램 생성 방법



[그림 5-4] 비선형 히스토그램 생성 순서도

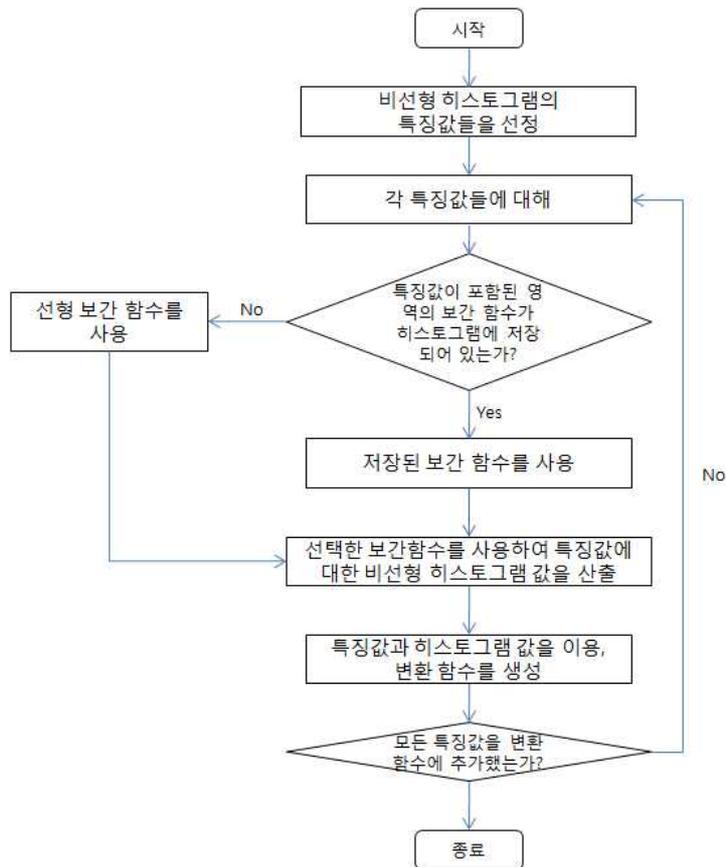
나. HDR 변환 함수 생성

이렇게 생성한 비선형 히스토그램을 이용하면 높은 동적 범위 변환 함수의 생성이 가능하다. 변환 함수를 생성하는 방법은 우선 비선형 히스토그램에서 특징값들을 샘플링한 뒤 각 값들을 보간하는 함수를 결정하는 것으로 시작한다. 보간 함수는 선택한 특징값들이 치역 값이 커서 나눈 정의역 영역에 포함되면 선형 함수를, 그렇지 않으면 각 구간에 저장된 데이터 값 분포 함수를 사용한다. 이는 이전 히스토그램 생성과정에 각 히스토그램에 저장되어있으므로 이것을 사용하면 된다. 특징값은 비선형 히스토그램의 정의역 값을 사용하되, 변환 함수의 정확도를 위해 충분한 개수를 사용해야 한다. 샘플링하는 특징값의 개수에 따라 변환 함수의 정확도가 높아지기 때문이다. 일반적으로 시각화 과정에서 변환 함수는 샘플링한 특징값의 사이를 선형 보간하므로 선형 보간으로 복원하지 못하는 치역 값이 작아서 합친 정의역 값들을 사용하는 것이 좋다. 선택한 특징값(A)을 선택한 보간함수를 이용, 비선형 히스토그램에 적용하여 치역값(B)를 구하고 여러 개의 (A, B) 쌍을 이용하여 변환 함수를 생성한다.

이때, 특징값 A의 집합을 $\{a_0, a_1, \dots, a_n\}$ 이라 하고 히스토그램 값 B의 집합을 $\{b_1, b_2, \dots, b_n\}$ 이라 할 때, 변환 함수 $f(x)$ 는 다음과 같은 구분된 선형 함수(piecewise linear function)로 정의된다.

$$f(x) = \frac{b_i - b_{i-1}}{a_i - a_{i-1}}x + f(0) \quad (\text{단, } 1 \leq i \leq n)$$

변환 함수 생성을 위한 순서도는 그림 5-4와 같다.

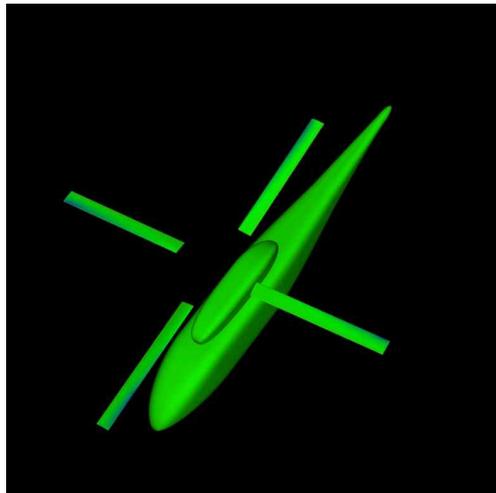


[그림 5-5] HDR 변환 함수 생성 순서도

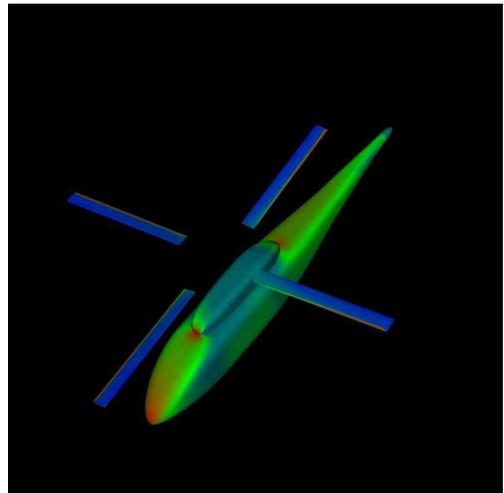
다. HDR 변환 함수 구현 및 적용

지금까지 설명한 방법을 이용해서 각 복셀이 32비트로 구성된 로터 시뮬레이션 데이터에 대한 가시화를 수행했다. 실험에 사용한 로터 시뮬레이션 데이터는 두 종류로, 각각 30과 90스텝으로 구성되었으며 전체 시간에 걸친 히스토그램을 생성한 뒤 이에 대한 비선형 히스토그램을 생성했다. 전체 시간에 대한 히스토그램을 생성함으로써 전체 시간을 위한 단일한 변환 함수의 생성이 가능했다.

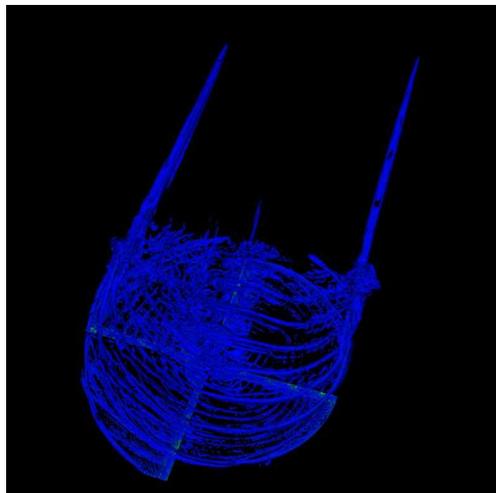
그림 5-3은 로터 시뮬레이션에 대해 다양한 가시화 작업을 수행한 뒤 일반적인 선형 히스토그램으로부터 생성한 변환 함수와 본 논문에서 제안한 비선형 히스토그램으로부터 생성한 변환 함수를 적용한 예이다. 그림 5-3에서 (a), (c), (e)는 선형 히스토그램을 적용한 예이고 (b), (d), (f)는 비선형 히스토그램을 적용한 예이다. 그림에서 볼 수 있듯이 선형 히스토그램으로부터 생성한 변환 함수를 사용했을 때 구분할 수 없었던 영역이 비선형 히스토그램으로부터 생성한 변환 함수를 사용했을 때는 명확히 구분됨을 알 수 있다.



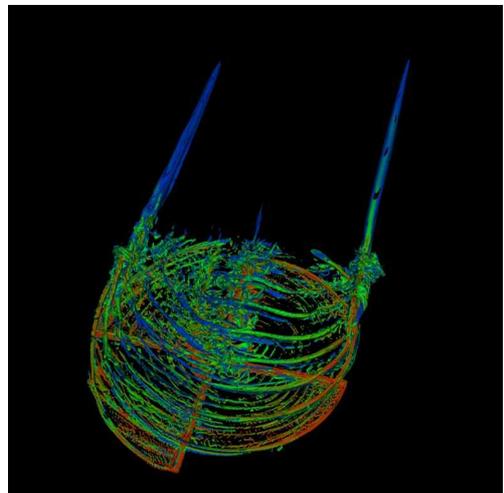
(a)



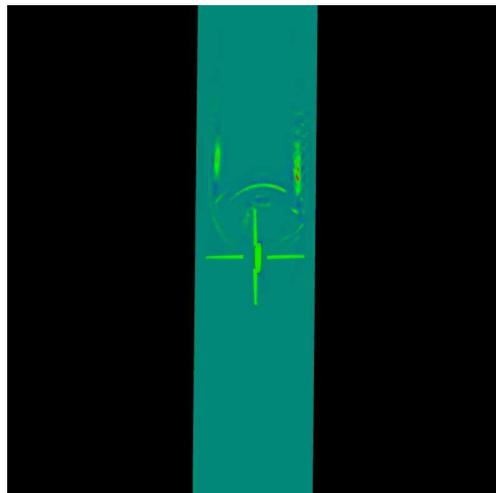
(b)



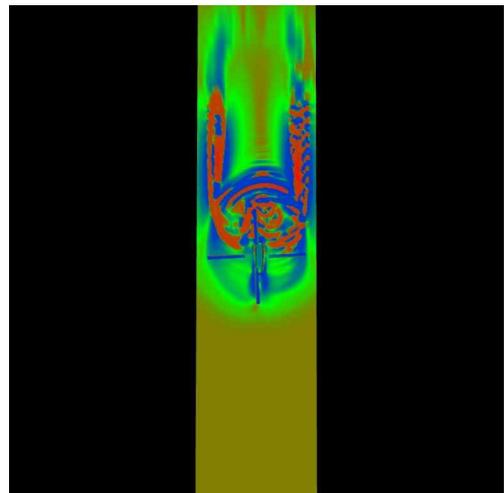
(c)



(d)



(e)



(f)

[그림 5-6] 변환 함수 적용 결과

6. 결론

본 기술문서에서는 높은 동적 범위를 갖는 수치 데이터의 가시화의 문제점과 이를 해결하기 위한 다양한 방법들에 대해 알아보고 새로운 저해상도 변환 함수 생성 방법을 제안했다. 현재까지 앞서 언급했던 4가지 문제점들을 모두 해결하는 방안은 제안되지 않았다. 그러나 일부 문제점들을 해결하고자 하는 방안들은 제시되고 있다.

본 기술문서에서 제안한 방법은 사용자의 개입 없이 자동으로 입력한 수치 데이터에 최적화된 높은 동적 범위 변환 함수를 생성함으로 정보의 손실을 최소화한 가시화를 가능하게 했다. 이 방법은 좁은 영역에 대부분의 수치 데이터가 집중되는 높은 동적 범위 수치 데이터에 대해 비선형 히스토그램을 생성함으로써 고해상도 변환함수에서 데이터가 많이 몰리는 부분을 살리는 방식으로 불균등하게 해상도를 낮춤으로써 저해상도 변환 함수를 이용해서도 정보의 손실을 최소화한 가시화를 가능하게 했다. 본 기술문서에서 제안한 알고리즘은 데이터가 전체 영역에 걸쳐 균등하게 퍼져있는 경우에도 기존의 균등하게 해상도를 낮추는 방식을 사용하는 경우와 같은 방식으로 변환 함수의 해상도를 낮출 수 있기 때문에 일반적인 높은 동적 범위 수치 데이터에 대해 적용이 가능하다.

본 기술문서에서는 높은 동적 범위를 갖는 수치 데이터의 가시화를 위한 변환 함수 생성 기법에 대해 조사했지만 시변환 데이터를 위한 시변환 변환 함수에 대한 설명은 미흡했다. 향후 높은 동적 범위의 시변환 데이터를 위한 변환 함수 생성 기법에 대한 추가적인 조사와 연구가 필요할 것이다.

7. 참고문헌

- [1] Wikipedia, http://en.wikipedia.org/wiki/High_dynamic_range_rendering
- [2] Paul E. Debevec and Jitendra Malik. "Recovering high dynamic range radiance maps from photographs", In SIGGRAPH '97: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, pages 369~378, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [3] Paul E. Debevec, "Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-Based Graphics with Global Illumination and High Dynamic Range Photography", In SIGGRAPH. '98
- [4] Christophe Schlick, "Quantization techniques for visualization of high dynamic range pictures", pages 7~20. Springer-Verlag, 1994.
- [5] Fredo Durand and Julie Dorsey, "Interactive tone mapping", In Proceedings of the Eurographics Workshop on Rendering Techniques 2000, pages 219~230, London, UK, 2000. Springer-Verlag.
- [6] F. Drago, K. Myszkowski, T. Annen, and N. Chiba, "Adaptive logarithmic mapping for displaying high contrast scenes", Computer Graphics Forum, 22: 419~426, 2003.
- [7] Kate Devlin, "Dynamic range reduction inspired by photoreceptor physiology", IEEE Transactions on Visualization and Computer Graphics, 11(1):13~24, 2005. Member-Reinhard, Erik.
- [8] Erik Reinhard, Michael Stark, Peter Shirley, and James Ferwerda, "Photographic tone reproduction for digital images", ACM Trans. Graph., 21(3):267~276, 2002.
- [9] Fredo Durand and Julie Dorsey, "Fast bilateral filtering for the display of highdynamic-range images", In SIGGRAPH '02: Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, pages 257~266, New York, NY, USA, 2002. ACM.
- [10] Prasun Choudhury and Jack Tumblin, "The trilateral filter for high contrast images and meshes", In SIGGRAPH '05: ACM SIGGRAPH 2005 Courses,

page 5, New York, NY, USA, 2005. ACM.

[11] Raanan Fattal, Dani Lischinski, and Michael Werman, "Gradient domain high dynamic range compression", In SIGGRAPH '02: Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, pages 249~256, New York, NY, USA, 2002. ACM.

[12] F. Bernardon, L. Ha, S. Callahan, J. Comba and C. Silva, "Interactive Transfer Function Specification for Direct Volume Rendering of Disparate Volumes", Technical paper UUSCI-2007-007, University of Utah

[13] Kitware Paraview, <http://www.paraview.org>

[14] S. Potts and T. Möller. "Transfer functions on a logarithmic scale for volume rendering", Proceedings of Graphics Interface, pages 57-63, 2004.

[15] M. Kraus, W. Qiao, and D. S. Ebert. "Projecting tetrahedra without rendering artifacts", Proceedings of IEEE Visualization 2004, pages 27.34, 2004.

[16] W. Qiao, D. S. Ebert, A. Entezari, M. Korkusinski, and G. Klimeck, "Vold: Direct volume rendering of multi-million atom quantum dot simulations", Proceedings of IEEE Visualization 2005, pages 319.326, 2005.

[17] A. Ghosh, M. Trentacoste, and W. Heidrich, "Volume Rendering for High Dynamic Range Displays", Proc. EG/IEEE VGTC Workshop Volume Graphics '05, pp. 91-98, 2005.

[18] J. Kniss, S. Premoze, M. Ikits, A. Lefohn, C. Hansen, and E. Praun. Gaussian transfer functions for multi-field volume visualization. In IEEE Visualization, pages 497~504, 2003.

[19] Xiaoru Yuan, Minh X. Nguyen, Baoquan Chen and David H. Porter, "HDR VolVis: High Dynamic Range Volume Visualization", IEEE Transactions on Visualization and Computer Graphics, pp. 433~445, July/August, 2006

[20] F. Drago, K. Myszkowski, T. Annen, and N. Chiba, "Adaptive Logarithmic Mapping for Displaying High Contrast Scenes", Computer Graphics Forum, vol. 22, no. 3, pp. 419~419, 2003.

[21] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, "Photographic Tone Reproduction for Digital Images," Proc. SIGGRAPH '02, pp. 267~276, 2002.

[22] R. Fattal, D. Lischinski, and M. Werman, "Gradient Domain High Dynam

ic Range Compression,” Proc. SIGGRAPH ’02, pp. 249~256, 2002.

[23] F. Durand and J. Dorsey, “Fast Bilateral Filtering for the Display of High-Dynamic-Range Images,” Proc. SIGGRAPH ’02, pp. 257~266, 2002.

[24] J. Kniss, G. Kindlmann, and C. Hansen, “Multidimensional Transfer Functions for Interactive Volume Rendering”, IEEE Transactions on Visualization and Computer Graphics 8, 3, 2002