

석사학위논문

상용 게임 엔진을 이용한
의료시뮬레이션 구축

2019년

한 성 대 학 교 대 학 원

정 보 시 스 템 공 학 과

정 보 시 스 템 공 학 전 공

하 태 준

석사학위논문
지도교수 계획원

상용 게임 엔진을 이용한
의료시뮬레이션 구축

Construction of medical simulation using
commercial game engine

2018년 12월 일

한성대학교 일반대학원

정보시스템공학과

정보시스템공학전공

하 태 준

석사학위논문
지도교수 계획원

상용 게임 엔진을 이용한
의료시뮬레이션 구축 시스템

Construction of medical simulation using
commercial game engine

위 논문을 공학 석사학위 논문으로 제출함

2018년 12월 일

한성대학교 일반대학원

정보시스템공학과

정보시스템공학전공

하 태 준

하태준의 공학 석사학위 논문을 인준함

2018년 12월 일

심사위원장 _____(인)

심 사 위 원 _____(인)

심 사 위 원 _____(인)

국 문 초 록

상용 게임 엔진을 이용한 의료시뮬레이션 구축

한 성 대 학 교 일 반 대 학 원
정 보 시 스 템 공 학 과
정 보 시 스 템 공 학 전 공
하 태 준

상용 게임 엔진의 기능이 향상되면서 범용 연구에 사용하려는 시도가 계속되고 있다. 본 연구는 임상 수술과 같은 의료 시뮬레이션 개발에 게임 엔진을 이용하려 한다. 게임 엔진을 사용하는 경우, 가상현실 하드웨어 지원이나 반투명한 물체의 깊이 정렬과 같이, 응용 프로그램 개발을 도울 수 있는 기능이 제공되어 편리하다. 한편 응용프로그램 구조를 게임 엔진의 특성에 적합하도록 수정해야하는 제약사항이 있다. 따라서 게임 엔진의 구조에 맞춰 의료 볼륨 데이터를 가시화하는 방법에 대해 설명하고 사용자의 요구에 즉각적으로 반응하여 관찰을 도울 수 있는 방법을 제안한다. 구체적인 방법으로, 인체의 주요한 부분을 사용자가 선택하면, 선택한 부분의 가시화 파라미터를 두드러지게 변경하여 복합적인 효과를 보이는 초점 배경 가시화를 볼륨 데이터에 적용하려 한다. 그 과정에서 본 연구는 누적 기반 초점 배경 볼륨 가시화 방법을 제안하며, 제안 방법을 통해 배경 부분은 초점 부분과 자연스럽게 투명해져 융합된다. 또한 제안 방법은 기존 볼륨 가시화 방법과 잘 결합되기 때문에 볼륨의 절개와 같은 가상

수술 기능이 원활하게 수행된다. 그 결과, 의료 볼륨 데이터와 표면 데이터가 함께 구성된 가상현실 장면을 생성 하였으며, 입력 이벤트에 반응하는 응용 프로그램 개발이 가능하였다. 다양한 기능을 내장한 게임 엔진은 의료 시뮬레이션 개발에 도움이 된다.

【주요어】 의료 가시화, 게임 엔진, Unity3D, 볼륨 가시화, 초점 배경 가시화, 가상현실

목 차

제 1 장 서 론	1
제 1 절 가상현실 산업	1
제 2 절 게임 엔진을 이용한 범용 연구	1
제 3 절 연구 내용	2
제 4 절 논문 구성	3
제 2 장 관련 연구	4
제 1 절 Unity3D	4
1) Unity3D 소개	4
2) 게임 오브젝트 및 셰이더 프로그램	4
제 2 절 볼륨 가시화	6
제 3 절 초점 배경(focus+context)가시화	7
1) 이차원 기반 방법	8
2) 삼차원 기반 방법	8
제 3 장 렌더링 시스템 구현	11
제 1 절 모델 설정	11
제 2 절 좌표계와 변환과정	12
제 3 절 광선 투사법 구현	14
제 4 절 텍스처 생성	16
제 4 장 초점-배경 볼륨 가시화	17
제 1 절 샘플 기반 초점 배경 볼륨 가시화	17
제 2 절 누적 기반 초점 배경 볼륨 가시화	18

제 3 절 볼륨데이터 절단 기능	22
제 5 장 실험	26
제 1 절 실험 환경	26
제 2 절 표면 물체와 복합 장면 구성	26
1) 다중 데이터 사용	26
2) 표면 물체와 복합 장면 구성	28
3) 절단된 볼륨 가시화	29
제 3 절 초점 배경 가시화 결과	29
1) 기법별 가시화 결과	29
2) 초점 배경 가시화 + 볼륨 절단	32
3) 초점 형태 변경	33
제 6 장 결론	35
참 고 문 헌	36
ABSTRACT	40

표 목 차

[표 2-1] 유니티 셰이더 프로그램 예시	6
[표 4-1] BFCVR 알고리즘	20
[표 4-2] 각 샘플링 포인트의 상태변수 값	22
[표 4-3] 절단 기능을 적용한 광선 투사법 코드	24
[표 5-1] 볼륨데이터 정보	26

그 립 목 차

[그림 2-1] 광선 투사법	7
[그림 2-2] 이차원 기반 방법(가상 렌즈): 볼륨 확대 렌더링	8
[그림 2-3] 삼차원 기반 방법(VML): 의료 데이터 세트 검사	10
[그림 3-1] 의료 영상과 게임 객체의 상호작용	11
[그림 3-2] Unity3D 세 가지 좌표계	13
[그림 3-3] 물체에 회전을 가했을 경우 광선 방향의 오류	14
[그림 4-1] FCVR과 SFCVR 도식 비교	18
[그림 4-2] 광선 진행시 샘플링 예시	19
[그림 4-3] SFCVR과 BFCVR 도식 비교	21
[그림 4-4] BFCVR 초점 경계면 판단 방법	21
[그림 4-5] 전역 좌표계 메쉬 생성과정	23
[그림 4-6] 각 메쉬별 광선의 진행	25
[그림 5-1] 다양한 볼륨데이터 가시화	27
[그림 5-2] 볼륨 및 표면데이터 복합 장면 구성	28
[그림 5-3] 서로 다른 시점에서 관찰한 분할된 볼륨 데이터	29
[그림 5-4] 기법별 가시화(초점 : 근육, 배경 : 골격)	30
[그림 5-5] 기법별 가시화(초점 : 골격, 배경 : 피부)	31
[그림 5-6] 절단된 메쉬의 초점 배경 가시화	32
[그림 5-7] 구 형태 초점 배경 가시화	33

제 1 장 서론

제 1 절 가상현실 산업

가상현실이란 컴퓨터로 만든 가상의 세계에서 사용자가 실제와 유사한 체험을 할 수 있도록 하는 기술이다. 시장조사업체 트렌드포스에 따르면 가상현실 시장 규모는 2020년이 되면 약 700억 달러에 이를 것으로 전망했다. 세계적인 기업인 Google, Facebook, Microsoft도 가상현실 산업에 적극적으로 투자 하고 있다. 국내 시장에서도 이러한 영향을 받아 VR시장은 점점 가속화 될 것으로 보이며, 이미 주요 업체에서는 개발 사업에 착수하였다.

가상현실은 시장이 크고 다양한 산업분야에 접목 가능한 기술로써 주목 받고 있다. 그 중 가상훈련분야는 사용자의 전문성을 보다 사실적인 환경을 제공함으로써 몰입감 높은 교육을 시행하고, 실제 훈련보다 더 안전하고 저렴한 비용으로 훈련을 시행할 수 있어 많은 관심을 받고 있다. 특히, 의료산업의 경우 의료인의 전문성은 환자의 생명과 직접적인 상관관계를 보이므로 반복적인 훈련이 필요하다. 전통적인 의학교육은 실제 환자를 대상으로 진행하여 안전성 보장과 동일한 환경제공 등이 어려움이 있다. 반면 가상현실을 이용한 교육은 의료인이 환자에게 해가 되지 않고, 동일한 환경을 제어하여 반복적인 시술 경험이 가능하기 때문에 안정적으로 의학적인 전문성을 기를 수 있다. 본 연구는 의료 시뮬레이션을 제작하는 것을 목표로 하는 기초연구이다.

제 2 절 게임 엔진을 이용한 범용 연구

게임 산업이 발전하면서 비디오 게임 제작을 위한 게임 엔진들이 다수 개발되었다. Unity3D, Unreal Engine 등으로 대표되는 게임 엔진을 통해

많은 게임이 개발되었다. 게임 엔진이 갖은 다양한 그래픽스 기능을 무료로 사용될 수 있게 되면서, 이를 범용 연구에 사용할 수 있는 가능성이 확대되었다. 실제 생물학에서 분자구조의 관찰이나, 대용량 생체 분자구조(biomolecular) 해석, 유전학 정보 가시화에도 Unity3D가 사용되었다(Lv et al., 2013; Le et al., 2015; Khalifa et al., 2015). 또한 다양한 개발도구를 사용하여 연구자는 필요한 핵심 기능 개발에 집중할 수 있다. 예를 들면, 게임 엔진은 간단한 충돌처리 기능을 자체적으로 지원하고, 가상현실 기기를 기본적인 설정으로 사용할 수 있어, 몰입도 높은 그래픽스 콘텐츠의 빠른 제작이 가능하다.

한편, 범용 연구에 사용하기 위해서는 게임 엔진의 구조적 특성에 맞추어 제약 사항을 준수해야 하며, 지원하지 않는 기능에 대해서는 연구의 제한점이 있다.

제 3 절 연구내용

본 연구는 대표적인 게임 엔진인 Unity3D(<https://unity3d.com/kr>)를 이용하여 가상현실 기반의 의료 진단 시스템을 구축하고자 한다. 그 초기 단계로서 의료 시뮬레이션에 적합한 모델로 의료데이터를 표현하고, 볼륨 절단 기능을 결합하는 방법을 제안한다. 또한 의료 시스템의 주요 기능인 초점 배경(focus+context) 볼륨 가시화를 구현한다(Wang et al., 2005). 초점 배경 가시화는 사용자가 지정한 부분인 초점 부분은 두드러지도록, 나머지 배경 부분은 더 투명하게 조작하는 방법이다. 즉, 사용자는 모델의 전체적인 구조를 파악할 수 있으며 동시에 관심 있는 부분을 설정하여 초점을 부각하는 복합적인 영상을 관찰할 수 있다. 본 연구는 Unity3D의 셰이더 프로그램(shader program)으로 볼륨 가시화 알고리즘을 구현하며, 배경부분의 투명도를 일관성 있게 제어하기 위해 누적 기반 초점 배경 가시화 방법을 제안한다.

또한 의료 시뮬레이션에 사용될 의료 데이터는 관찰뿐 아니라 타 물체와의 상호작용이 빈번하게 일어난다. 이러한 조건을 만족하기 위해 타 물

체와의 상호 작용이 가능하도록 Unity3D의 게임 오브젝트로 의료 데이터를 대응하였다. 이러한 방법을 사용하면 게임 엔진에서 기본적으로 제공하는 도형생성 및 조작, 충돌 처리 기능 등 기본 게임엔진의 객체가 사용하는 기능 대부분을 그대로 사용할 수 있어 기능을 추가하기 용이하며, 다른 표면 데이터와 복합 장면 구성 및 이벤트 처리가 용이하다.

제 4 절 논문 구성

본 논문의 구성은 2장에서 관련연구에 대해 소개하며 게임 엔진을 이용한 방법과 일반적인 가시화 방법의 차이를 분석한다. 3장에서 렌더링 시스템 구축에 대해 설명한다. 4장에서 초점 배경 가시화를 제안하고 이전 연구와의 차이점을 설명한다. 5장에서는 실험 결과를 보이고, 6장에서 결론을 맺는다.

제 2 장 관련연구

제 1 절 Unity3D

1) Unity3D 소개

Unity3D는 멀티 플랫폼 게임 엔진으로 개발자가 손쉽게 게임을 개발할 수 있는 저작 도구이다. 여러 배포 버전 중 개인용 버전을 사용하면, 비 영리용 응용 프로그램은 물론, 수익이 연간 10만 달러 이하의 상업용 소프트웨어를 개발할 수 있고, 배포 또한 무료로 이용할 수 있다.

Unity3D의 중요한 장점으로 다양한 SDK가 제공되어 기능 확장을 간단하게 할 수 있는 점이 있다. 또한 멀티 플랫폼으로 배포를 지원하여 PC, 모바일 환경뿐만 아니라, 가상현실 콘텐츠 제작도 용이하다. 예를 들어 Google SDK와 Oculus SDK가 제공 되어 가상현실 콘텐츠를 간단한 설정만으로 개발할 수 있다. 본 연구에서 개발한 의료 영상 가시화 시스템은 설정 변경만으로 Oculus SDK를 사용하는 가상현실 시스템으로 이식할 수 있다.

2) 게임 오브젝트 및 셰이더 프로그램

셰이더 프로그램이란 객체의 픽셀(fragment) 위치와 색상을 계산하는 함수다. 즉, 색조, 명암, 조명등을 블렌딩 하여 최종 색상을 결정하는 것을 의미한다. 셰이더 프로그램은 크게 두 가지 종류로 나뉜다. 첫 번째는 고정 함수셰이더(fixed function shader)이고, 두 번째는 프로그램 가능한 셰이더(programmable shader)이다. 고정 함수 셰이더는 프로그래머가 제어할 수 있는 부분이 없지만, 프로그래밍 가능한 셰이더의 경우 프로그래머가 원하는 계산식을 이용하여 정점의 정보와 픽셀의 색상 등을 결정 할

수 있다. 프로그램 가능한 셰이더는 정점 셰이더와 픽셀 셰이더로 나뉜다. 정점 셰이더는 물체의 정점의 정보를 정의하고 정보의 값을 이용하여 물체를 이동시키거나 텍스처를 조작하는 작업을 수행하는 함수이다. 픽셀 셰이더는 최종적으로 픽셀이 갖게 될 색상 값을 계산하는 함수이다. 이는 구현에 따라 간단한 단색을 나타 낼 수도 있지만, 다양한 함수를 포함하는 복잡한 계산식으로 표현될 수 있다. 본 연구에서는 픽셀 셰이더 부분에 광선 투사법을 구현함으로써 픽셀의 색상을 볼륨 데이터의 값과 조명을 적절한 수식에 의해 혼합된 색상으로 결정한다.

한편, 셰이더 프로그램은 GPU에서 수행되기 때문에 SIMD(single instruction multiple threads)구조에 맞춰 프로그래밍 하는 것이 효율적이다. 하나의 프로세스는 배정받은 픽셀의 계산만 하도록 하여야 하며, 셰이더 프로그램 안에 복잡한 조건 분기 문을 사용하면 조건에 부합하지 않는 프로세스에 한해서 유향 프로세스가 발생하기 때문에 이를 회피하기 위한 특수한 공식을 만드는 등의 연구가 필요하다.

Unity3D는 셰이더 스크립트 언어로 'ShaderLab'을 사용하는데 내부로 Cg/HLSL 언어로 작성 할 수 있다. Cg로 구현한다면 셰이더 스크립트 Pass블럭 내부에 지시어 'CGPROGRAM'과 'ENDCG'을 사용하여 블럭을 선언하고, 내부에 코드를 작성한다. [표 2-1]은 간단한 셰이더 코드 예시이다. 정점 셰이더에서 좌표 변환을 수행하고, 보간 된 정점 속성이 픽셀 셰이더에서 변환 없이 출력되는 것을 확인할 수 있다.

Unity3D를 이용하여 가시화하는 방법은 다음의 과정을 거친다. 구성하고자 하는 장면 내부에 육면체나 구 등의 게임 오브젝트를 생성하고, 이에 적용할 재질속성을 부여한다. 재질 속성은 게임에서 사용하는 일반적인 3차원 텍스처일 수도 있고, 연구자가 직접 생성한 계산식일 수도 있다. 직접 생성한 계산식은 그래픽스 파이프라인에서 사용하는 프로그램 가능한 셰이더로 표현되는데, 본 연구는 셰이더 프로그램으로 광선 투사법을 구현하여 볼륨 가시화를 수행한다. 즉 게임 오브젝트의 표면에서 셰이더 프로그램이 수행되고 의료 데이터를 이용하여 가시화를 수행한다.

[표 2-1] 유니티 셰이더 프로그램 예시

```
CGPROGRAM

#pragma vertex vert
#pragma fragment frag
#include "UnityCG.cginc"

struct v2f {
    float4 pos : SV_POSITION;
    float3 uvw : TEXCOORD0;
};

v2f vert(appdata_base v) {
    v2f o;
    o.pos = mul(UNITY_MATRIX_MVP, v.vertex);
    o.uvw = v.vertex + 0.5f;
    return o;
}

float4 frag(v2f i) : COLOR{
    return float4(i.uvw,1);
}

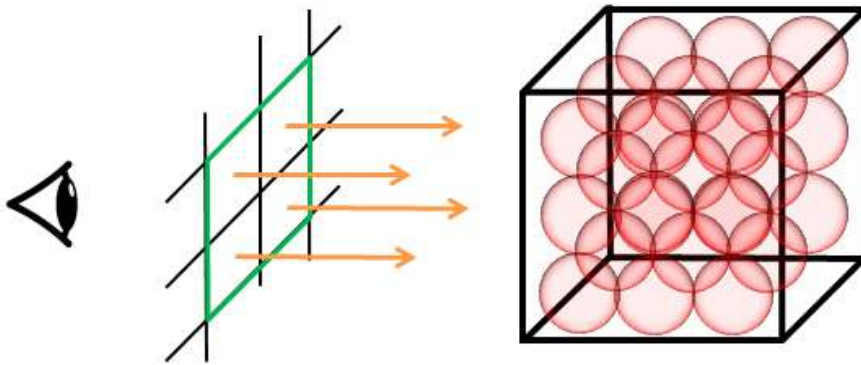
ENDCG
```

제 2 절 볼륨 가시화

볼륨 가시화란 삼차원 배열 데이터를 가시화 하는 수단이다(Sabella, 1998; Engel et al., 2006). 본 연구의 삼차원 데이터는 인체 CT 데이터로, 인체에 X선을 투영하여 획득한 많은 수의 단면 영상이다. 신체의 장기 및 골격은 밀도가 높을수록 더 많은 X선을 흡수하는 점을 이용하여 가시화를 수행한다. 볼륨 가시화 방법은 다음과 같다, 삼차원 배열 데이터를 입력으로 삼아, 사용자가 부여한 전이함수(Transfer Function; TF)를 이용하여 색상과 투명도로 변환한다(Levoy, 1988). 예를 들면 X선 흡수도

가 높은 뼈는 불투명한 흰색으로, X선 흡수도가 약간 낮은 근육은 반투명한 붉은색으로 변환하는 TF를 생성할 수 있다. 그리고 사용자의 관찰 방향으로 색상과 투명도를 누적하여 출력 영상을 생성한다.

본 연구에서는 [그림 2-1]과 같이 많이 이용되는 광선 투사법(ray casting)을 이용한다(Levoy, 1988). 광선 투사법은 관찰 지점에서 가상의 평면을 생성하고, 관찰 방향으로 각 픽셀마다 가상의 광선을 발사한다. 각 광선은 일정한 간격마다 볼륨 데이터에서 샘플링을 통해 배열 값을 얻는다. 배열 값은 TF를 통해 색상과 투명도로 변환되고, 이들의 누적된 값이 출력 픽셀이 된다.



[그림 2-1] 광선 투사법

제 3 절 초점 배경(focus+context)가시화

초점 배경 가시화란 사용자가 중점적으로 관찰할 영역을 초점으로 정의하고, 나머지 영역을 배경으로 정의하여 초점과 배경을 함께 가시화하는 방법이다. 초점을 부각하여 관찰하기 위해서 일반적으로 초점과 배경에 서로 다른 TF를 사용한다.

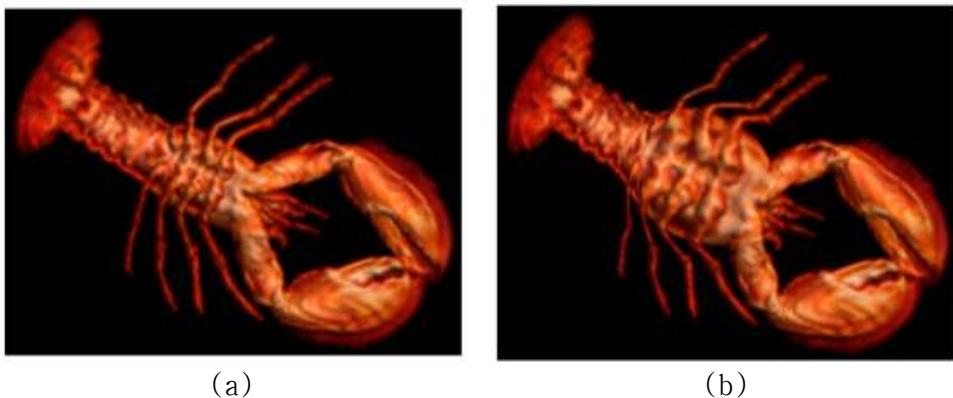
예를 들어, 인체의 다리 데이터에서 무릎을 초점으로 정의하고, 나머지를 배경으로 정의할 수 있다. 초점은 골격을 보고, 배경 부분은 피부를 보고 싶다면 초점 부분은 X선 흡수도가 높은 부분만 보이도록 TF를 생성하고, 배경은 반대가 되도록 TF를 생성한다. 초점 내부인지 여부를 판단하

여 서로 다른 TF를 적용하여 가시화하면 원하는 영상이 출력된다.

사용자가 초점 부분을 지정하는 방법은, 출력 영상에서 영역을 결정하는 이차원 기반 방법과 삼차원 공간상에서 원하는 부분을 지정하는 삼차원 기반 방법으로 나눌 수 있다.

1) 이차원 기반 방법

이차원 기반 방법은 물체가 초점 속하는지 배경에 속하는지 판단이 간단하다는 장점이 있다. 마술 볼륨 렌즈(magic volume lens)는 사용자가 출력 영상의 관심 부분에 가상의 렌즈를 위치시키고, 렌즈에 효과를 부여하여 다시 가시화를 수행하는 방법이다(Wang et al., 2005). 렌즈의 위치에 따라 광선의 경로를 변경하여, 초점 부분이 확대된 영상을 생성한다. 가상렌즈(virtual lens) 방법은 화면 좌표계 기준으로 사각형 영역을 설정하여, 서로 다른 TF로 가시화한다(Gallo and Placitelli, 2013).



[그림 2-2] 이차원 기반 방법(가상 렌즈): 볼륨 확대 렌더링
(a) 렌즈 없음, (b) 원형 렌즈 (Wang et al., 2005)

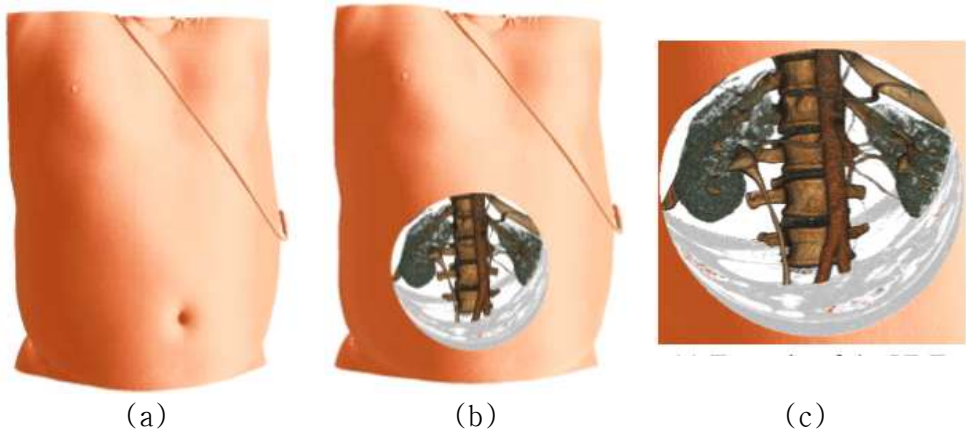
2) 삼차원 기반 방법

삼차원 기반 방법은 사용자가 볼륨 데이터 내부에 초점에 해당하는 삼

차원 영역을 정의한다. 초점 영역이 구나 육면체라면 내부/외부 판정을 간단하게 계산할 수 있으며, 원뿔이나 사용자 지정의 복잡한 도형을 사용할 때는 전처리가 필요한 경우도 있다.

ClearView 기법은 간단하게 초점과 배경을 결합하는 방법으로, 초점과 배경을 별개로 각각 렌더링한 이후에 두 영상을 합성한다(Kruger et al., 2006). 사용자가 지정한 중심 위치에 따라 가중치를 두어 합성하여, 초점이 잘 관찰되도록 설계하였다. Monclus 등은 원뿔 모양의 가상 손전등(lantern)을 사용하였는데, 각 샘플 위치가 원뿔 내부인지 판단하여 조건 분기를 통해 물체의 색상을 결정한다(Monclus et al., 2009). 양방향렌즈(interactive lenses)는 사용자가 정의한 렌즈 내부영역을 부각하는 방법이다(Ropinski and Hinrichs, 2004). 렌즈 영역 내부에 특별한 효과를 부여하기 위해 깊이 테스트의 설정을 바꿔 여러 번 가시화한다. 중요성 중심 시각화(importance driven rendering)는 객체에 부여하는 색상과 불투명도 외에 중요성(importance)이라는 새로운 속성을 추가하였다(Viola et al., 2006). 중요성 속성은 중심부와 배경에 전처리로 차등 적용되며, 가시화 단계에서 중요성이 높은 물체가 부각되도록 합성한다.

본 연구는 삼차원 기반 초점 배경 가시화를 수행하되, 초점 경계면에서 효과를 부여하는 새로운 방법을 제안한다. 앞서 말한 선행 연구는 초점과 배경 부분을 동시에 관찰 가능하지만, 겹치는 부분에서 서로 상호관계에 따른 복합적인 관찰이 어렵다. 예를 들어 초점이 배경의 안쪽에 완전히 둘러싸여있다면 초점 관측을 위해 배경을 없애거나 전체적으로 반투명하도록 설정한다. 하지만 이러한 방법은 배경의 색상을 손실하게 되어 초점과 배경의 뚜렷한 혼합 영상을 얻기 어렵다. 이에 본 연구에서는 초점에 가려지는 부분에 한해서 투명도를 조절하여 복합적인 영상을 얻을 수 있도록 하였다.



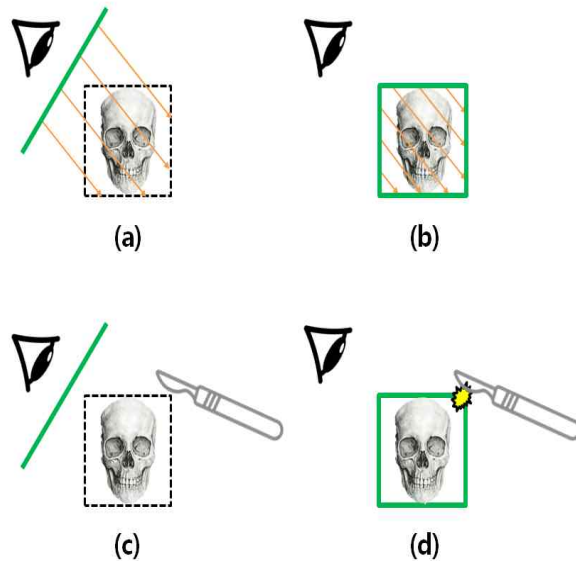
[그림 2-3] 삼차원 기반 방법(VML): 의료 데이터 세트 검사

(a) Original model, (b) Virtual Magic Lantern(VML),
(c) Zoom-in of the VML region of interest (Monclus et al., 2009)

제 3 장 렌더링 시스템 구현

제 1 절 모델 설정

일반적인 볼륨가시화는 각 픽셀을 채우기 위해 많은 연산이 필요하여, 사용자가 지정한 관찰자 위치에 가상의 평면을 생성하고 평면상에 영상을 입히는 것이 일반적이다. 그러나 본 연구에서는 의료 시뮬레이션을 특성을 고려하여 Unity3D의 게임오브젝트 중 부피가 있는 객체를 사용해 각 겉면에 영상을 생성하는 방식을 사용한다.



[그림 3-1] 의료 영상과 게임 객체의 상호작용

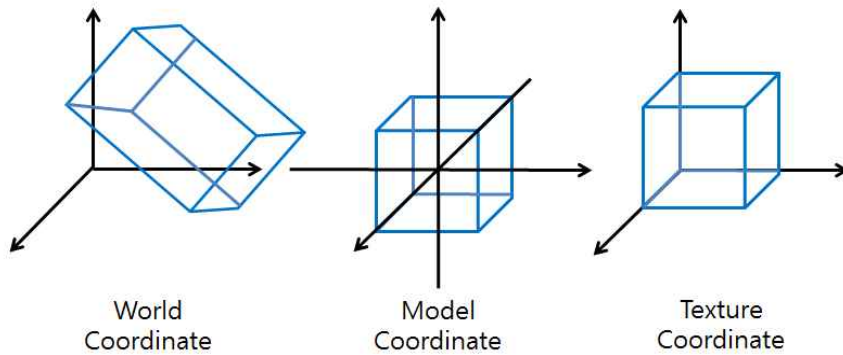
먼저 두 방식의 비교를 그림에 보이고 있다. (a), (b)에 붉은 선으로 표시한 부분은 Unity3D에서 생성한 객체의 표면이며, 객체의 충돌연산이 발생하는 기준이 된다. 볼륨 가시화한 결과, 기존 방법 (a)에 상이 땀하면 관찰자는 (c)와같이 영상 뒤쪽에 물체가 존재하는 것처럼 가상적으로 느

끼게 된다. 한편, 제안 방법 (b)의 경우 육면체에서 관찰자에게 가까운 3개의 면에서 가시화가 수행되며, 겹면에 영상이 대응된다. 그 결과 관찰자는 육면체 내부에 물체가 존재하는 것처럼 느끼게 된다. 결과적으로 관찰자가 느끼는 영상은 차이가 없다.

이 때, 사용자가 인체를 폴리곤으로 생성한 가상 수술도구로 접촉하였을 경우, (c)의 경우는 접촉이 발생하지 않았다고 판정하게 된다. 이는 사용자가 관찰하고 있는 입체적인 영상에 비해 충돌 검사가 이뤄지는 객체는 부피를 갖지 않기 때문이다. 한편 본 연구의 제안 모델은 (d)와 같이 충돌 검사의 주체가 육면체가 되므로 올바른 충돌 판정을 얻게 된다. 즉, 제안 방법을 사용하면 Unity3D에서 제공하는 충돌 검사 기능을 게임 오브젝트와 같이 적용 가능한 장점이 있다. 제안된 모델을 사용하기 위해서는 Unity3D에 존재하는 좌표계를 변환하여 일치시키는 작업이 필요하다.

제 2 절 좌표계와 변환과정

텍스처를 사용하는 일반적인 그래픽스 응용프로그램은 전역 좌표, 물체 좌표, 텍스처 좌표와 관찰자 좌표로 구성된다. OpenGL에서 모델 변환과 관찰자 변환이 하나의 Model-View 행렬에 반영 되어 있듯이, 일반적인 광선 투사법 알고리즘은 볼륨 데이터가 전역 좌표에 고정되어 있다고 가정하고, 관찰자의 위치와 방향만을 이동한다. 즉, 모델 변환은 항등변환이며, 관찰자 변환만 존재한다. 그러나 실제 의료 시뮬레이션에서 의료 영상을 의미하는 볼륨 데이터는 의료 기구 등의 타 물체와의 상호 작용이 일어난다. 따라서 볼륨 데이터 전역 좌표 변환이 가능한 모델설정은 중요하다.



[그림 3-2] Unity3D 세 가지 좌표계

Unity3D에는 제어 가능한 세 종류의 좌표계가 있다. 첫 번째는 여러 게임 오브젝트를 추가 할 수 있는 전역 좌표계가 있고, 두 번째는 전역 좌표계에 존재하는 각 객체의 물체 좌표계가 있으며, 세 번째는 각 오브젝트에 적용할 텍스처 좌표계가 있다. 관찰자 변환은 카메라의 위치와 속성을 지정하여 제어하며, 사용자가 직접 계산을 수행하지 않는다.

본 연구는 Unity3D에서 제공하는 게임 오브젝트인 육면체를 배치하고, 이 육면체가 볼륨 데이터를 의미하도록 구성하였다. 육면체를 렌더링 하면, 재질 속성으로 연구자가 부여한 셰이더 프로그램이 수행되며, 육면체 표면에 광선 투사법이 적용된 의료영상이 맷히도록 구성하였다. 이 방법은 게임 오브젝트를 추가로 생성하여 복잡한 장면구성이 편리하고, 자유로운 움직임이 가능한 장점이 있다.

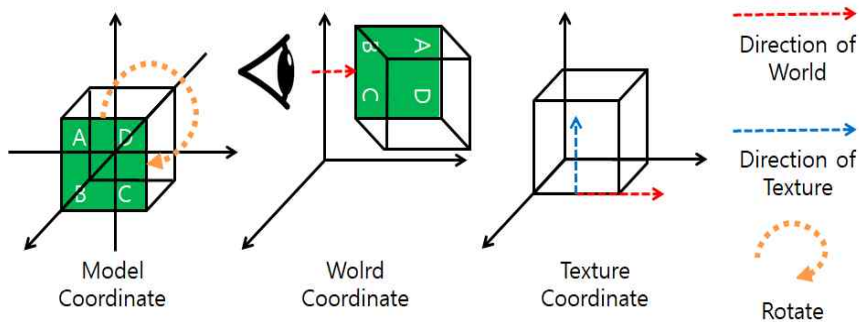
즉, 전역 좌표는 변환이 자유롭고 텍스처 좌표는 $[0, 1]$ 범위의 정육면체 좌표계로 설정되므로 물체 좌표를 편리하게 설정하여 변환을 쉽게 하는 것이 중요하다. 본 연구는 볼륨 데이터를 의미하는 원점을 무게중심으로 하는 단위 육면체로 모델링 하였다. 제안 방법을 사용하면 전역 좌표계와 물체 좌표계 사이에는 일반적인 모델 변환으로 상호 변환이 가능하며, 원점을 중심으로 설정하여 회전 변환도 쉽게 대응된다. 한편, 텍스처 좌표와 모델 좌표는 평행이동으로 일치되므로, 간단하게 변환이 수행되는 장점이 있다.

제 3 절 광선 투사법 구현

광선 투사법은 텍스처 공간에서 광선을 진행하면서 샘플링, 색상 부여, 누적을 반복한다. 본 절에서는 Unity3D의 속성 값으로부터 텍스처 좌표를 생성하는 방법을 설명한다. 각 광선을 진행하기 위해서는 광선의 진행하는 방향과 시작점이 필요하다.

먼저 광선의 시작점에 대해 계산한다. 본 연구는 육면체에 의료 데이터를 모델링 할 때, 물체 좌표의 각 정점에 공간 좌표 $[-0.5, 0.5]$ 와 텍스처 좌표 $[0, 1]$ 를 대응해 놓는다. 픽셀 셰이더가 시작할 때, 텍스처 좌표가 보간되므로, 육면체 표면의 각 픽셀에 대해 대응되는 시작 텍스처 좌표를 입력 파라미터로 얻을 수 있다. 다른 방법으로는 물체 좌표에 0.5를 더하는 연산을 통해 텍스처 좌표를 계산할 수도 있다.

광선의 진행 방향을 계산하는 것은 주의가 필요하다. 광선의 진행 방향은 육면체 표면의 한 점과 관찰자의 위치를 잇는 직선으로 결정된다. 육면체의 정점은 텍스처 좌표계 또는 물체 좌표계에서 결정되고, 관찰자의 위치는 전역 좌표에서 얻을 수 있다. 따라서 [그림 3-3]과 같이 모델 변환이 가해지면, 전역 좌표계에서 얻은 광선 방향은 그대로 사용할 수 없다.



[그림 3-3] 물체에 회전을 가했을 경우 광선 방향의 오류

이 문제를 해결하는 간단한 방법은, 전역 좌표의 카메라 위치 ($CamPos_w$)를 물체 좌표로 변환(WtoM)하고, 물체 좌표에서 육면체 표면

상 각 점($FragPos_m$)을 이어 광선의 진행방향($RayDir_m$)을 (1)과 같이 결정하는 것이다.

$$RayDir_m = FragPos_m - WtoM(CamPos_w) \quad (1)$$

또는 정점 셰이더에서 전역 좌표계를 기준으로 카메라 위치와 육면체 표면의 점을 계산하는 (2)로 변형하는 방법을 사용할 수 있다.

$$RayDir_m = WtoM(MtoW(FragPos_m) - CamPos_w) \quad (2)$$

본 연구는 (2)를 사용하였는데, 이는 전역 좌표계에서 광선 시작점을 얻고 이후 전역 좌표계 상에서 일어나는 충돌처리 및 가림 등에 대한 확장이 가능하기 때문이다. (2)의 연산은 (1)에 비해 복잡하지만, MtoW 변환은 정점 셰이더에서 육면체를 구성하는 8개 정점에서만 발생하므로, 8번의 행렬 곱이 추가되는 것에 불과하다. 이때, WtoM 변환과 MtoW 변환은 각각 Unity3D에서 저장하고 있는 ‘UNITY_WORLDTOOBJECT’ 행렬과 ‘UNITY_MATRIX_M’ 행렬을 곱하면 된다.

이렇게 얻은 물체 좌표에서의 광선 진행 방향은 텍스처 좌표에서 광선 진행방향과 동일한데, 3.2절에서 텍스처 좌표계와 물체 좌표계 사이의 변환은 평행이동만 반영하도록 설계하였기 때문이다. 좌표계가 일치된 광선의 위치와 진행방향을 이용하여 셰이더 프로그램으로 광선 투사를 구현한다. 변환된 텍스처 좌표계에서 광선의 진행 방향으로 이동하면서 텍스처 샘플링을 수행하고 얻어온 값을 색상으로 변환하고, 누적(alpha blending)하여 영상을 생성하였다.

제 4 절 텍스처 생성

일반적인 볼륨 데이터는 삼차원 배열로 구성되므로, 삼차원 텍스처를 사용하여 가시화를 수행한다. Unity3D는 삼차원 텍스처를 'Texture3D' 자료형으로 지원한다. 다만 파일에서 'Texture3D'형태로 직접 생성하는 방법은 지원되지 않는다. 데이터를 읽을 때 중간 단계의 이차원 메모리에 복사하고, 이를 여러 번 결합하여 Texture3D를 생성하는 과정을 거쳤다.

CT나 MR등의 의료 데이터는 16bit 정수형으로 저장되어 있다. CUDA나 OpenGL은 다양한 텍스처 형식을 지원한다. 예를 들어 OpenGL의 단일 채널 16bit 정수형 텍스처 형태인 LUMINANCE16은 OpenGL 1.1에 포함되어 있으며, 10 여년 이상 오래된 하드웨어에서 지원된다.

그러나 Unity3D에서 제공하는 텍스처 형식인 TextureFormat은 16bit 정수형을 지원하지 않아, 23bit의 정밀도를 갖는 32bit 실수형 'RFloat'을 사용하였다. Unity3D는 16bit 실수형의 텍스처 형식도 지원하지만 이는 가수부가 11bit로 정밀도의 손실이 발생하는 문제가 있어 선택하지 않았다.

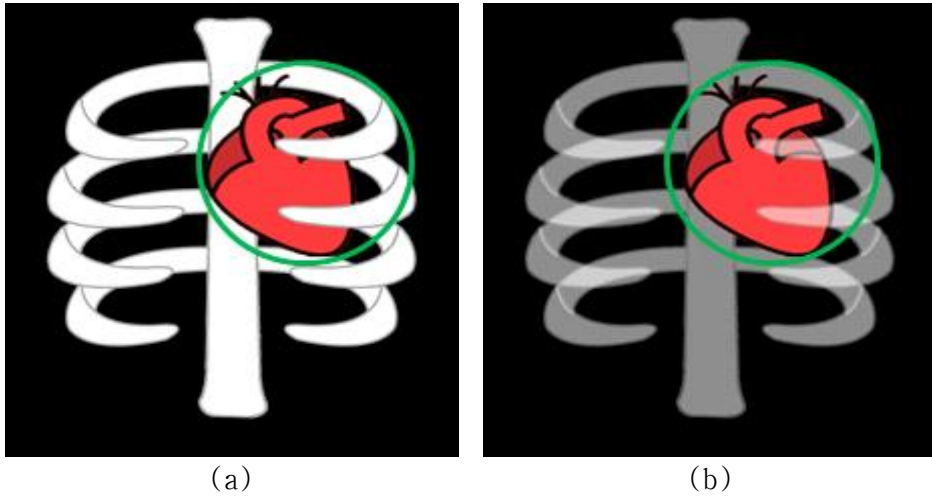
한편, 삼차원 텍스처의 크기는 2의 지수(power)형태의 크기만을 허용한다. 예를 들어 $512 \times 512 \times 400$ 크기의 볼륨 데이터가 있다면, 텍스처의 크기를 512^3 으로 생성해야 한다. 이를 정리하면 일반적인 의료영상의 경우 $512^3 \times 4\text{Bytes} = 512\text{MB}$ 의 텍스처 메모리를 사용하게 된다.

제 4 장 초점 배경 볼륨 가시화

제 1 절 샘플 기반 초점 배경 볼륨 가시화

전통적인 광선 투사법은 생성된 면으로부터 텍스처를 통과할 때까지 점진적으로 이동하면서 샘플링을 하여 색상 및 투명도를 누적한다. 이러한 방법은 초점 배경 가시화(Focus Context Volume Rendering; FCVR)에 그대로 적용하기는 어렵다. 이는 초점 앞에 불투명한 배경 부분이 위치하면 초점은 배경에 가려져 관측하기 어렵기 때문이다. 따라서 배경을 더 투명하게 표현하는 방법이 연구되었으며, 배경을 완전히 투명하게 만든다면 초점과 배경의 상호관계를 알 수 없으므로(Monclus et al., 2009; Ropinski and Hinrichs, 2004), 배경을 적절하게 반투명하게 하여 가시화를 수행하는 것이 일반적이다(Viola et al., 2006; Kruger and Fogal, 2009).

기존 연구에서 사용되는 샘플 기반초점 배경 볼륨 가시화(Sampling based Focus Context Volume Rendering; SFCVR)는, 각 샘플링 위치가 사용자 지정한 영역의 내부라면 초점으로 판단하여 정상적으로 가시화하고, 외부라면 더 투명하게 조정하여 가시화하는 방법이다.



[그림 4-1] FCVR과 SFCVR 도식 비교 :

(a) FCVR, (b) SFCVR

[그림 4-1]에서 사용자는 심장을 초점으로 정의하여 붉은색을 부여하고, 배경은 흰색으로 골격을 표현하였다고 가정하였다. 단순히 두 개의 TF를 사용한다면 (a)는 골격이 심장을 가리게 되지만, (b)는 배경의 투명도를 조정하여 초점인 심장을 드러내고, 사용자가 원하는 영상을 생성한다. 물론 사용자가 배경의 TF를 더 투명하게 제작한다면 (b)와 동일한 효과를 얻을 수 있다.

제 2 절 누적 기반 초점 배경 볼륨 가시화

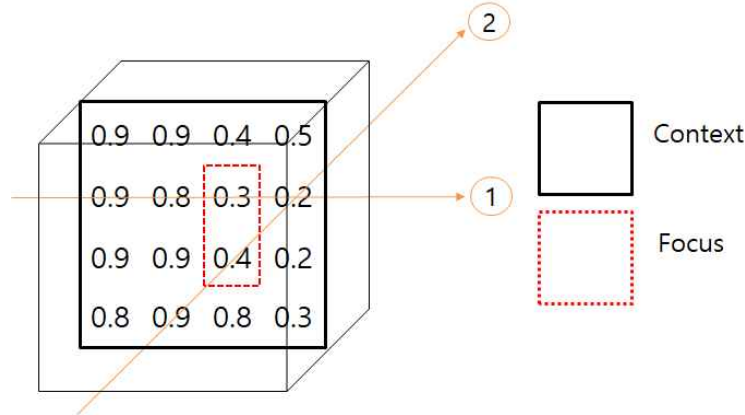
4.1절 SFCVR를 이용하면, 배경이 투명해져서 초점 관찰이 쉽다. 하지만 초점 앞에 배경 물체가 두텁게 존재한다면, 여전히 초점 부분을 가리는 문제가 있다. 이 과정을 [그림 4-2]을 이용하여 설명한다.

[그림 4-2]은 입력 데이터의 각 위치별로 불투명도(alpha)가 표시되어 있으며 초점이 불투명도가 높은 배경에 둘러싸여 있는 상태이다. 일반적인 누적은 alpha blending(3)을 사용하나(Levoy, 1988), SFCVR은 배경 부분을 더욱 투명하게 만드는 사용자 지정 상수 β 를 곱해 (4)와 같이 계

산한다.

$$acc_n = acc_{n-1} + \alpha_n(1 - acc_{n-1}) \quad (3)$$

$$acc_n = acc_{n-1} + \alpha_n\beta(1 - acc_{n-1}) \quad (4)$$



[그림 4-2] 광선 진행시 샘플링 예시

단, acc_n 은 현재까지 누적된(accumulated) 불투명도, acc_{n-1} 은 직전 샘플까지 누적된 불투명도이고, a_n 은 현재 샘플의 불투명도이다. [그림 4-2] 1번 광선의 경우 광선이 진행하면서 누적되는 불투명도를 계산하여 보았다. 예를 들어 $\beta=0.5$ 라고 가정하면, 광선이 진행하면서 0.9, 0.8, 0.3, 0.2와 같은 값을 만나며 이들 (4)식에 반복 적용하면 (5)와 같은 변화를 겪는다.

$$acc_0 = 0 \quad (5)$$

$$acc_1 = 0 + (1 - 0) \times 0.9 \times 0.5 = 0.45$$

$$acc_2 = 0.45 + (1 - 0.45) \times 0.8 \times 0.5 = 0.67$$

즉, 불투명한 배경에 여러 번 누적을 수행하면 초점에 도달하기 전에 점점 불투명해져서 초점이 가려진다. 이를 방지하고자 β 를 매우 작게 설정하면, [그림 4-3](a) 같이 배경이 지나치게 투명해지는 문제가 있다.

이 문제를 해결하기 위해, 초점과 배경 각각을 가시화하고 영상처리 기법으로 두 결과 영상을 혼합하거나 (Viola et al., 2006; Kruger and Fogal, 2009), 배경의 투명도를 자동으로 조절하는 기존 연구도 있으나(Zheng et al., 2013), 여러 번 반복적으로 가시화를 수행해야 하는 단점이 있다.

본 연구는 이 문제를 해결하기 위해 누적 기반초점 배경 볼륨 가시화 (Blending based Focus Context Volume Rendering; BFCVR)을 제안한다. BFCVR은 한 번의 가시화를 수행하면서, [표 4-1]과 같이 배경에서 초점으로 진입하는 지점에서만 누적된 투명도를 조절한다.

[표 4-1] BFCVR 알고리즘

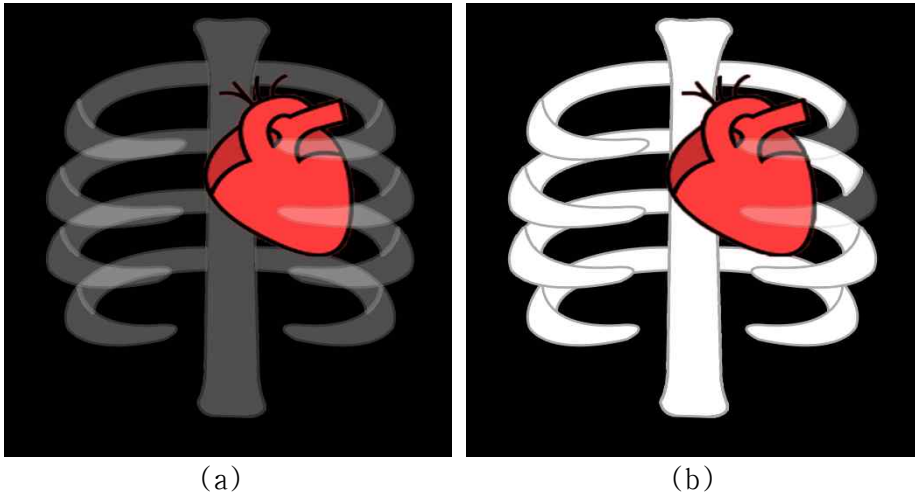
```

if (crossBoundary) then  $acc_{n-1} = acc_{n-1} * \beta$ 
 $acc_n = acc_{n-1} + (1 - acc_{n-1}) * a_n$ 
    
```

배경이 두겹게 위치하더라도 누적된 불투명도는 1을 넘을 수 없으므로, 제안 방법은 경계면을 통과하면서 누적된 배경의 불투명도가 β 이하로 조절된다. 그 결과 배경 뒤의 초점을 용이하게 관찰할 수 있다. 예를 들어 (5)의 식은 (6)와 같이 계산되어 누적 값이 결과가 β 이하가 된다.

$$\begin{aligned}
 acc_0 &= 0 \\
 acc_1 &= 0 + (1 - 0) \times 0.9 = 0.9 \\
 acc_2 &= 0.9 + (1 - 0.9) \times 0.8 = 0.98 \\
 acc_2 &= 0.98 \times 0.5 = 0.49
 \end{aligned} \tag{6}$$

제안한 BFCVR 방법은 배경 두께에 독립적으로 투명도를 제어할 수 있다. 그 결과 기존 SFCVR이 [그림 4-3](a)와 같다면 제안하는 BFCVR은 [그림 4-3](b)와 같은 일관된 영상이 생성된다.

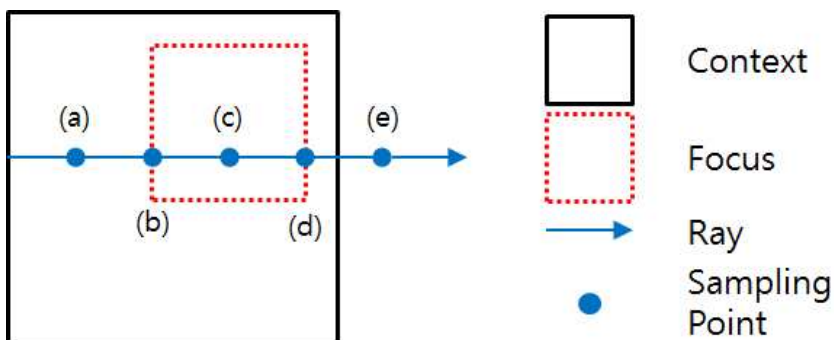


[그림 4-3] SFCVR과 BFCVR 도식 비교 :

(a) SFCVR, (b) BFCVR

한편, BFCVR을 구현하기 위해서는 경계면을 통과하는 시점을 알아야 한다. 초점 부분의 모양은 사용자가 입력하므로, 샘플이 초점 내부인지의 판정은 간단하게 할 수 있는 점을 착안하여 본 연구는 다음과 같이 구현하였다.

광선 진행 과정에서 직전 샘플링 지점이 배경에 속하였으나 현재 샘플링 지점이 초점에 속한다면 경계면을 통과 하였다고 정의할 수 있다. 본 연구는 광선의 직전 샘플링 상태를 preState, 현재 샘플 위치가 초점에 속하는지 여부를 inFocus로 두고 상태 변화를 이용하였다.



[그림 4-4] BFCVR 초점 경계면 판단 방법

[표 4-2] 각 샘플링 포인트의 상태변수 값

Sampling point	state	
	preState	inFocus
(a)	0	0
(b)	0	1
(c)	1	1
(d)	1	0
(e)	0	0

[그림 4-4]는 제안 방법을 이용한 광선 진행의 예시이다. 광선위의 각 점은 샘플링 위치를 나타내며 초점진입 이전(a), 초점진입 지점(b), 초점 내부(c), 초점탈출 지점(d), 초점 외부(e)를 가정하였다. 상태 변수의 값으로는 배경에 위치하면 0, 초점에 위치하면 1로 정하였다. 위와 같은 방식으로 각 지점의 상태변수를 계산하면 [표 4-2]와 같다. [그림 4-4](b)점은 경계면에 진입을 의미하는데 그 값은 preState=0, inFocus=1인 값이다. 광선이 가질 수 있는 모든 경우에서 통과하기 까지 상태 변수가 중복되는 경우가 없으므로, 상태 변수의 값으로 경계면 진입을 판단 할 수 있다. 따라서 본 연구는 [표 4-1]의 crossBoundary를 $(1 - \text{preState}) \times \text{inFocus}$ 로 정의하여 초점 경계면 진입을 판단하였다. 제안 하는 방법은 광선의 진행마다 상태변수를 업데이트 하는 방법으로 간결하며, 이후 다수의 초점이 존재하여도 적용 가능한 확장성을 갖고 있다.

제 3 절 볼륨데이터 절단 기능

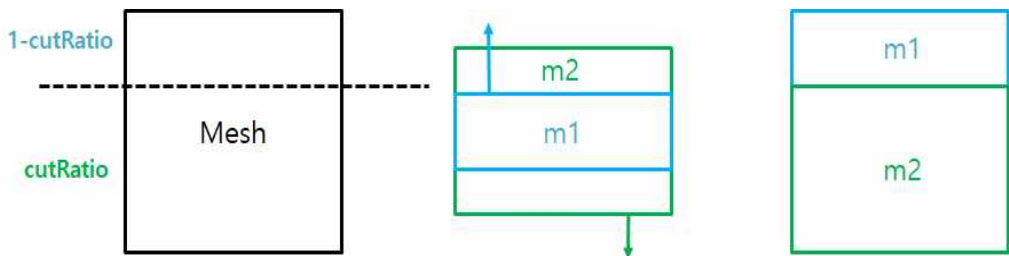
의료 시뮬레이션은 본래 관찰만 하던 일반적인 볼륨가시화와 달리 타 물체와 상호작용이 빈번하게 일어난다. 그 중 절단은 가장 기본적인 기능이다. 사용자가 직관적으로 이용하기 위해서는 절단한 후에도 이전과 같은 기능들이 동일하게 작동해야 한다. 본 연구는 하나의 의료 데이터를 두

개로 나누어 각각의 데이터를 관찰하는 상황을 구현하였다. 이를 위해서는 메쉬를 두 개 생성하고 각 메쉬에 정점 좌표와 및 텍스처 좌표를 변경해서 대입해야 한다. 각 메쉬는 셰이더 프로그램에 의해 Unity3D에서 반복적인 계산을 통하여 영상이 출력된다. 각 메쉬는 깊이 정렬도 보존되지만 메쉬 간 교차가 일어나면 카메라와 더 가까운 메쉬가 먼저 렌더링 되고, 그 뒤에 더 먼 메쉬가 렌더링된다. 즉, 겹친 부분에 대한 깊이 정렬은 보장하지 않는다. 결과적으로 기존 게임엔진의 기능을 이용하여 자연스럽게 이용 가능하도록 절단 기능을 구현하였다. 구현을 위해서 크게 두 가지 과정을 거친다.

첫 번째는 월드 좌표계에 의료 데이터를 2개의 메쉬로 나누고 위치 조정을 해야 한다. 목표하고자 하는 의료 시뮬레이션은 의료 데이터가 전역 좌표 변환이 자유롭기 때문에 절단 이전과 동일한 위치로 2개의 메쉬를 생성하기 위해서는 원 메쉬의 위치 정보를 이용하여야 한다. Unity3D는 객체의 중점을 속성으로 갖고 있으므로 (7),(8)공식을 이용하면 동일한 위치로 생성 가능하다. 아래 식에서 $cutRatio$ 는 아래 생성될 객체의 절단할 비율을 의미하고, (7)번은 $cutRatio$ 비율로 절단될 객체, (8)번은 $(1-cutRatio)$ 비율로 절단될 객체의 생성 위치식이다.

$$objPos = objPos - 0.5 \times objScale \times (1 - cutRatio) \quad (7)$$

$$objPos = objPos + 0.5 \times objScale \times cutRatio \quad (8)$$



[그림 4-5] 전역 좌표계 메쉬 생성과정

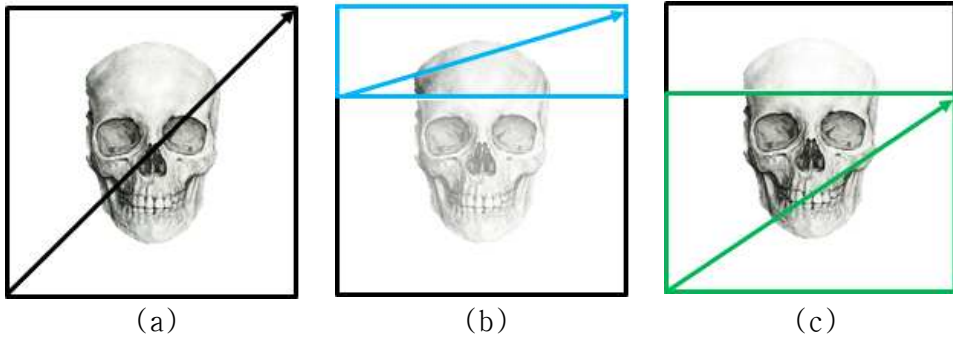
두 번째는 절단된 각 메쉬의 텍스처를 조정해야 한다. 여기에는 두 가지 방법이 있는데, 각 메쉬에 맞는 텍스처를 생성하는 것과 원 텍스처를 사용하되 내부 광선을 조작하는 방법이 있다. 본 연구에서는 후자를 선택하였는데 의료 시뮬레이션에서 절단은 기본 기능이므로 빈번하게 사용하는데, 절단이 시행 될 때 마다 텍스처를 생성하는 것은 한정적인 텍스처 메모리가 손실되기 때문이다. 원 메쉬의 광선 투사법은 $[0, 1]$ 범위 텍스처의 처음부터 끝까지 광선이 지나가도록 설계되어 있는데, 이러한 광선의 이동 범위를 절단한 비율을 이용하여 제한하는 방법을 사용한다. 한편 $(1 - \text{cutRatio})$ 비율로 절단된 메쉬의 경우 광선의 시작 위치를 보정해 줘야 하는데, $[0, \text{누적된 cutRatio}]$ 범위에는 아래 객체의 텍스처로 사용되기 때문이다. 광선의 출발 위치를 아직까지 누적된 cutRatio의 합만큼 이동하여야 한다. [표 4-3]에서 절단이 적용된 코드를 소개한다. accumulateH는 아직까지 누적된 cutRatio의 합을 의미하고, 위에 생성될 객체의 광선 출발위치를 조절하기 위해 사용된다.

[표 4-3] 절단 기능을 적용한 광선 투사법 코드

```
rayStart += accumulateH;
for( step = startTex; step < endTex; step++)
{
    pos = rayStart + dir * step * accumulateRatio;
}
```

[표 4-3]에서 pos는 광선의 현재 위치, rayStart는 광선의 출발지점, dir는 광선의 진행방향, step는 광선의 진행정도, accumulateRatio는 $(1 - \text{cutRatio})$ 의 합으로 광선의 길이를 얼마나 제어하는지를 결정한다. 이를 이용하면 원 텍스처에 필요한 한 벌의 메모리로 절단된 모든 객체를 가시화 할 수 있어 효율적이다. [표 4-3]의 코드는 [그림 4-6]와 같이 작동한다. [그림 4-6](a)와 같이 진행하던 광선은 [그림 4-6](b),(c)와 같이 렌더링 되어야 할 부분으로 공간이 제한되며 (b)는 위쪽 객체이므로

광선의 출발위치가 이동하였음을 알 수 있다.



[그림 4-6] 각 메쉬별 광선의 진행 :

(a) Original, (b) $(1 - \text{cutRatio})$, (c) cutRatio

제 5 장 실험

제 1 절 실험 환경

본 연구의 실험은 Intel i5-4670 @ 3.40GHz CPU와 GeForce GTX 1060 GPU를 장착한 개인용 컴퓨터에서 수행되었다. Unity3D의 버전은 5.5.4p4이고 셰이더 언어는 CG을 사용하여 광선 투사법을 구현하였다.

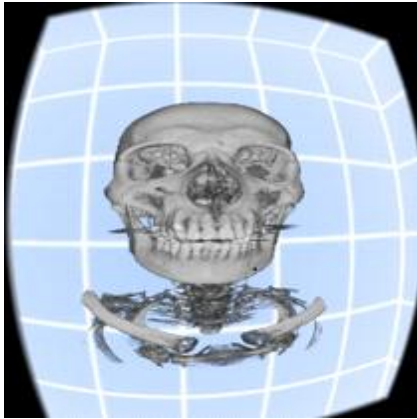
[표 5-1] 볼륨데이터 정보

	Dimension	Size	Texture usage
Head	512×512×512	256MB	512MB
Chest	512×512×512	256MB	512MB
Leg	512×512×512	256MB	512MB
Lower	512×512×512	256MB	512MB

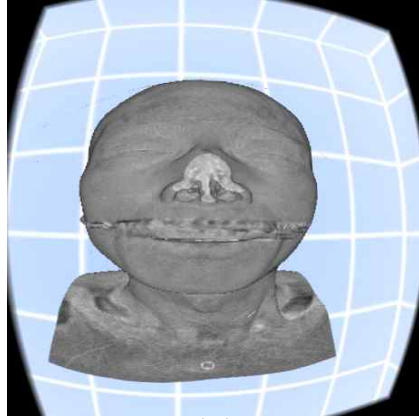
사용한 의료 데이터의 정보는 [표 5-1]과 같다. Unity3D에서 지원하는 Texture3D의 크기는 2의 지수형태만 가능하며, 텍스처 메모리 용량 문제로 512MB 이상 데이터는 생성이 불가능하였다.

제 2 절 표면 물체와 복합 장면 구성

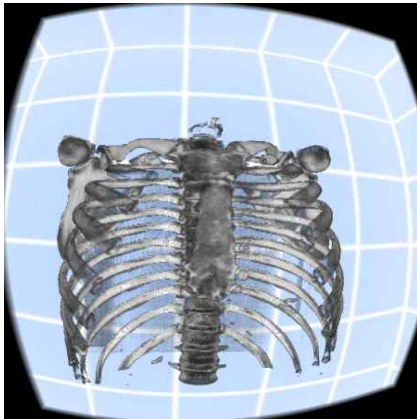
1) 다중 데이터 사용



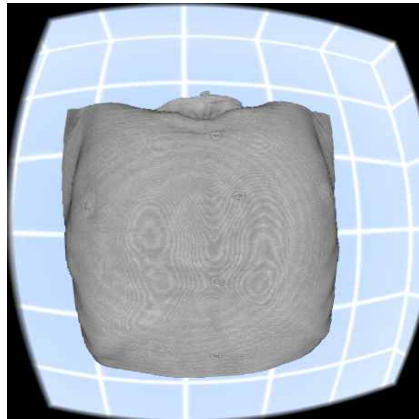
(a)



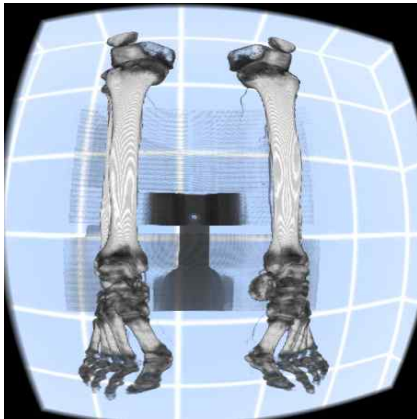
(b)



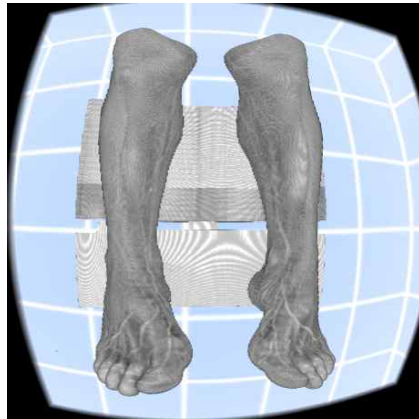
(c)



(d)



(e)

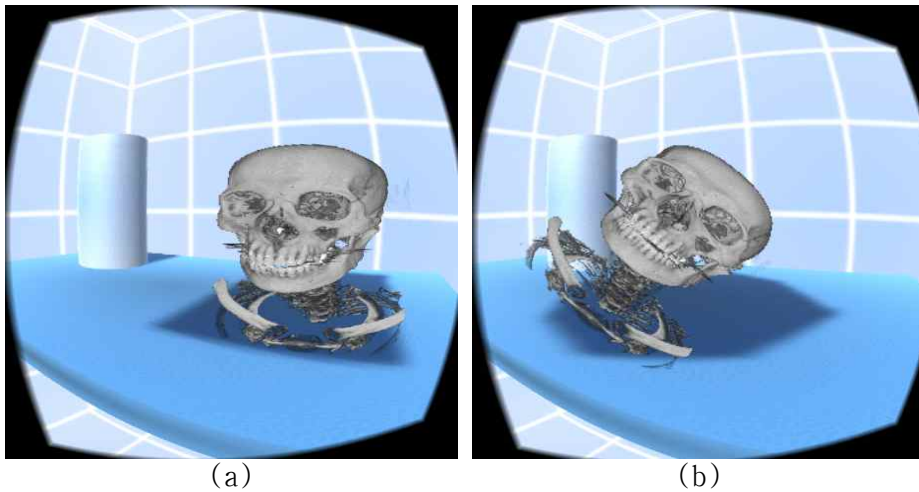


(f)

[그림 5-1] 다양한 볼륨데이터 가시화

[그림 5-1]은 다양한 볼륨데이터를 가시화한 결과이다. (a),(b)는 Head 데이터, (c),(d)는 Chest 데이터, (e),(f)는 Leg 데이터를 나타내고 있다, 불투명도 전이함수(opacity transfer function)를 조정하여 왼쪽은 골격을 가시화 하였으며, 오른쪽은 피부를 가시화 하였다.

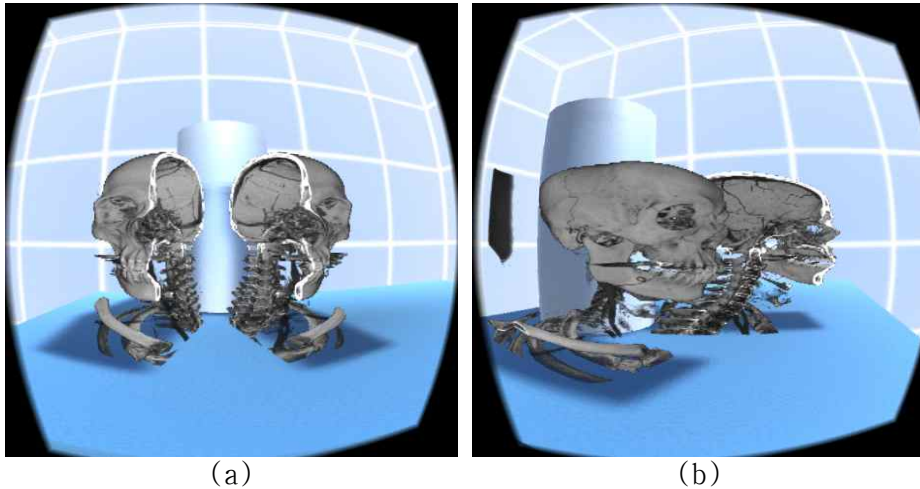
2) 표면 물체와 복합 장면 구성



[그림 5-2] 볼륨 및 표면데이터 복합 장면 구성

볼륨 데이터와 육면체, 원기둥 등의 다른 표면 물체를 복합 가시화 한 결과를 [그림 5-2]에 보이고 있다. Head 데이터의 뒤로 일반 게임 오브젝트인 원기둥이 표시되며, [그림 5-2](b)에서 볼 수 있듯이 볼륨 데이터는 모델 변환이 자유롭게 가능하다. 또한 볼륨데이터가 게임오브젝트로 대응되어 표면 데이터와 볼륨 데이터간의 가림 효과가 정상적으로 표현된다. Unity3D에서 기본으로 제공하는 그림자 효과도 표면 되어있다. 볼륨 데이터를 육면체로 모델링하여 그림자가 육면체로 표시된 것을 확인할 수 있다.

3) 볼륨 절단



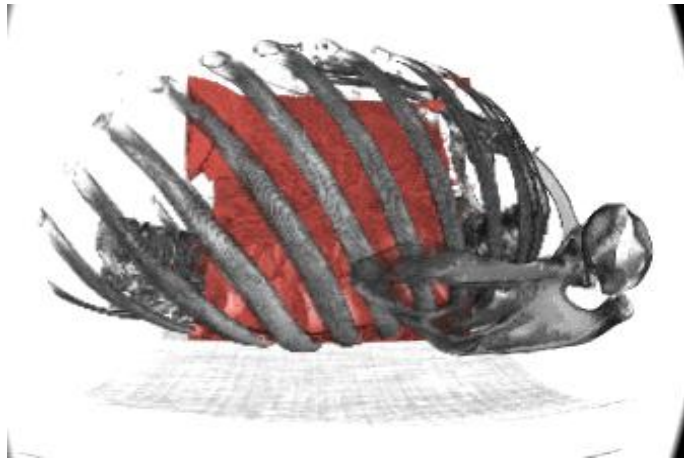
[그림 5-3] 서로 다른 시점에서 관찰한 분할된 볼륨 데이터

Head 데이터를 좌우로 나누고, 둘을 회전, 이동하여 분할된 형태로 가시화한 결과를 [그림 5-3]에 제시하고 있다. 분할된 각각의 볼륨은 자유로운 기하 변환이 가능하며, [그림 5-3](b)에서 관찰할 수 있듯이 볼륨들이 겹쳐서 서로 가려지는 효과도 올바르게 표시되고 있다.

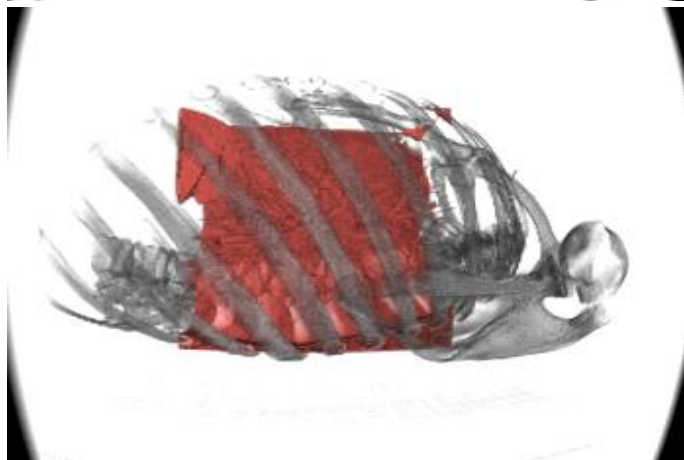
제 3 절 초점 배경 가시화 결과

1) 기법별 가시화 결과

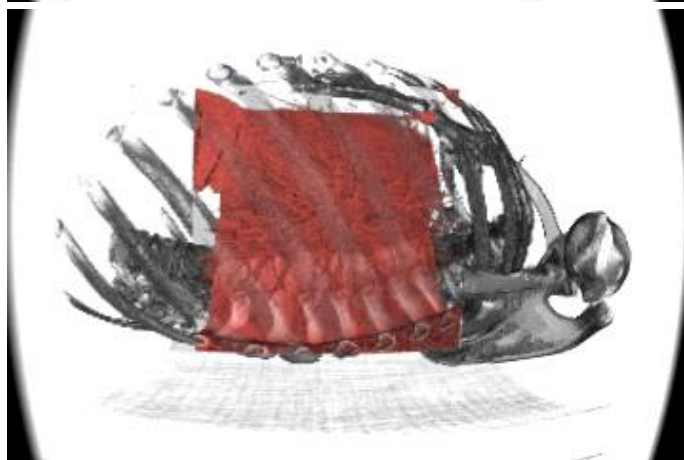
(a)



(b)



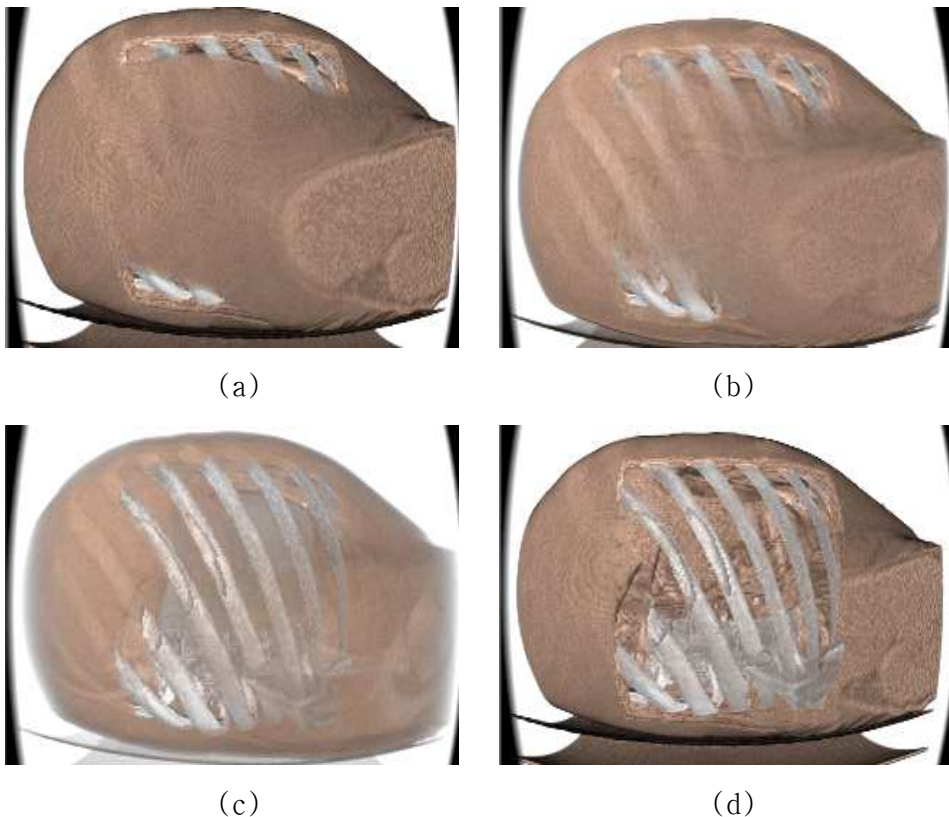
(c)



[그림 5-4] 기법별 가시화 (초점 : 근육, 배경 : 골격) :
(a) FCVR, (b) SFCVR, (c) BFCVR

[그림 5-4]는 4.1, 4.2에서 제안한 SFCVR과, BFCVR을 사용한 가시화 결과이다. 초점은 육면체 영역으로 근육을 붉은색으로 보이도록 설정하였고, 배경은 초점 외의 영역으로 설정되며 골격을 흰색으로 관찰하도록 설정하였다. 기법에서 사용되는 감소율은 각각 0.25로 동일하게 지정하였으며, 볼륨 데이터는 Chest를 사용하였다.

기존 방법(a)는 초점 부분이 외부의 골격에 가려 관측이 방해되고 있다. 이를 해결하는 SFCVR(b)은 배경에 해당하는 골격을 투명하게 만들어 내부 관찰이 용이하나, 주변부 골격도 모두 투명해지는 부작용이 있다. 제안 방법인 BFCVR(c)에서는 초점 부분 앞쪽에 위치한 골격만 투명하게 하여 중심부 관찰이 가능하며, 초점의 외부에 해당하는 다른 골격은 그대로 유지시켜 복합가시화가 원활하게 일어나고 있다.

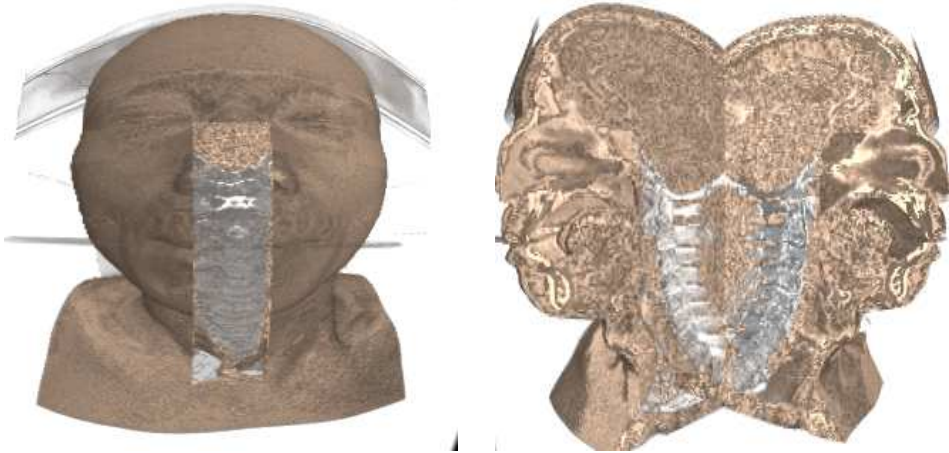


[그림 5-5] 기법별 가시화 (초점 : 골격, 배경 : 피부) :

- (a) FCVR, (b) SFCVR($\beta=0.25$),
(c) SFCVR($\beta=0.05$), (d) BFCVR($\beta=0.25$)

[그림 5-5]는 4절의 [그림 4-3](a)와 같이 밀도가 높은 배경안쪽에 초점이 있을 때의 영상이다. 픽셀을 색상이 결정되기 위해서 많은 샘플링을 거치기 때문에 SFCVR기법으로 초점을 뚜렷하게 보이도록 높은 감소율을 적용하면 (c)와 같이 배경이 매우 흐릿하게 나타는 것을 확인할 수 있다. 반면 제안하는 방법인 (d)는 감소율을 0.25로 하였음에도 불구하고 초점과 겹치는 배경에 대해서는 투명하게 처리하고, 나머지 배경을 또렷하게 관측 가능하다.

2) 초점 배경 가시화 + 볼륨 절단



(a)

(b)

[그림 5-6] 절단된 메쉬의 초점 배경 가시화

[그림 5-6]는 4절에서 소개한 BFCVR과 볼륨 절단을 같이 적용하여 얻은 영상이다. Head 데이터를 사용하였으며, 초점은 경추 영역으로 골격이 흰색으로 보이도록 설정하였고, 배경은 피부로 설정하였다. (a)는 절단

이전의 영상이며, (b)는 중앙을 기준으로 절단한 영상이다. 절개 기능이 원활하게 수행되고 있으며, (b)에서 볼 수 있듯이 두개골 부분을 제외한 경추 부분만이 흰색으로 올바르게 표시되어 있다. 절개를 시행하면 기존에 초점도 동일하게 분해가 되어 동일한 위치에 표시되는 것을 볼 수 있다. 이를 통해 사용자는 미리 설정해 놓은 초점을 잃지 않고 관측 가능하다. 또한 원 의로 데이터가 두 부위로 나뉘고 따로 조작이 가능한 것과 동일하게 초점을 두 부위로 나뉘서 조작 가능하여 사용자의 초점 부분을 자유롭게 설정 가능하도록 설계 하였다.

3) 초점 형태 변경



[그림 5-7] 구 형태 초점 배경 가시화

[그림 5-7]은 초점을 육면체가 아닌 구 형태로 설정한 결과이다. TF는 상기 [그림 5-6]과 동일하며 Lower데이터를 사용하였다. 초점 부분을 구로 설정하여 골반에 위치하였다. 이처럼 초점 부분의 설정은 내부/외부 관정이 가능한 수식을 생성한다면, 임의의 도형으로 적용 가능하다. 하지

만 셰이더 프로그램이 GPU에서 실행되므로 조건 분기 문을 최대한 줄이는 내부/외부 판정식을 완성하는 것이 중요하다.

제 6 장 결론

본 논문은 게임 엔진 Unity3D를 이용하여 의료 시뮬레이션 프로그램을 제작하기 위한 기초 모델을 설계하고, 사용자 요구의 즉각적으로 반응 가능한 응용 기술을 제안하였다. 먼저 Unity3D에서 제공하는 육면체 게임 오브젝트에 볼륨 데이터를 대응하고, 텍스처 좌표와 전역 좌표 사이의 변환을 셰이더 프로그램으로 수행하여, 광선 투사법을 구현하였다. 기존의 가상 평면에서 광선을 발사하는 대신, 육면체 표면에서 광선을 투사하는 방법을 적용하여 삼차원 CT 의료 영상을 게임 오브젝트로 취급 가능하였고, 게임 엔진에서 제공하는 이벤트처리 및 스크립트를 사용한 동작을 구현할 수 있다. 그 활용으로, 절개 기능 등을 손쉽게 추가할 수 있다.

그 후 의료영상에 대해 중요부분을 조작하여 내부/외부를 복합적으로 관찰하는 방법을 제안하였다. 기존 방법은 배경이 지나치게 초점을 가리거나, 배경이 지나치게 투명해지는 단점이 있다. 본 연구는 초점의 경계 진입 시 누적된 불투명도를 수정하는 BFCVR방법을 제안하였고 이를 통해 원활한 복합 가시화를 수행할 수 있다.

한편, Unity3D를 사용할 경우 제한점은 다음과 같다. 의료 데이터에서 주로 사용하는 16bit 정수형 텍스처를 'TextureFormat'으로 지원하지 않아 32bit 실수형 텍스처를 사용해야 하고, 2의 지수 형태로만 생성 가능하여 메모리 낭비가 존재한다. 또한 의료파일을 읽어 삼차원 텍스처를 생성할 때 중간 단계로 이차원 텍스처를 통해 생성해야 하는 불편함이 있다. 그럼에도 최근의 게임 엔진의 발전을 검토해보면 볼륨 가시화와 의료 응용프로그램의 제작에 게임 엔진을 사용하는 것은 전망이 밝다.

참 고 문 헌

1. 국내문헌

하태준, 계획원. (2017). Unity3D 게임 엔진을 이용한 의료 데이터 가시화. 『컴퓨터그래픽스학회 논문지』, 23(3), 87-94.

하태준, 계획원. (2018). 유니티 게임 엔진 기반의 의료 시뮬레이션을 위한 초점 배경 볼륨 가시화. 『한국게임학회 논문지』, 18(4), 65-74.

2. 국외문헌

- Engel, K., Hadwiger, M., Kniss, J. M., Rezk-Salama, C. and D. (2006). Real-Time Volume Graphics. 『CRC Press』.
- Gallo, L. and Placitelli, A. P. (2013). High-Fidelity Visualization of Large Medical Datasets on Commodity Hardware. 『ISRN Biomedical Engineering』, 1-9.
- Khalifa, N.H., Nguyen, Q.V., Simoff, S. and Catchpoole, D. (2015). A visualization system for analyzing biomedical and genomic data sets using Unity3D platform. 『In Proc. 8th Australasian Workshop on Health Informatics and Knowledge Management』, 47-53.
- Krüger, J. and Fogal, T. (2009). Focus and context-visualization without the complexity. 『World Congress on Medical Physics and Biomedical Engineering』, 45-48.
- Kruger, J., Schneider, J. and Westermann, R. (2006). Clearview: An interactive context preserving hotspot visualization technique. 『IEEE Transactions on Visualization and Computer Graphics』, 12(5).
- Kruger, J. and Westermann, R. (2003). Acceleration techniques for GPU-based volume rendering. 『In Proceedings of the 14th IEEE Visualization』, 38.
- Levoy, M. (1988). Volume Rendering Display of Surfaces from Volume Data. 『IEEE Computer Graphics and Application』, 8, 29-37.
- Le Muzic, M., Autin, L., Parulek, J. and Viola., I. (2015). cellVIEW: a Tool for Illustrative and Multi-Scale Rendering of Large Biomolecular Datasets. 『In Eurographics Workshop on Visual

- Computing for Biology and Medicine』, 61–70.
- Lv, Z., Tek, A., Da Silva, F., C. Empereur–mot, M. Chavent, and M. Baaden, (2013). Game On, Science – How Video Game Technology May Help Biologists Tackle Visualization Challenges. 『PLOS ONE』, 8(3), e57990.
- Monclus, E., Diaz, J. Navazo, I., and Vazquez, P. P. (2009). The virtual magic lantern: an interaction metaphor for enhanced medical data inspection. 『In Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology』, 119–122.
- Ropinski, T. and Hinrichs, K. (2004). Real–time rendering of 3D magic lenses having arbitrary convex shapes. 『In Proc. of International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)』, 379–386.
- Sabellia, P. (1998). A Rendering Algorithm for Visualizing 3D Scalar Fields. 『Proceeding of the 15th Annual Conference on Computer Graphics and Interactive Techniques』, 51–58.
- Viola, I., Feixas, M., Sbert, M. and Groller, M. E. (2006). Importance–driven focus of attention. 『IEEE Transactions on Visualization and Computer Graphics』, 12(5), 993–940.
- Wang, L., Zhao, Y., Mueller, K. and Kaufman, A. (2005). The magic volume lens: An interactive focus+ context technique for volume rendering. 『Visualization, IEEE』, 367–374
- Zheng, L., Correa, L, C. and Ma, K. L. (2013). Visibility guided multimodal volume visualization. 『In Bioinformatics and Biomedicine (BIBM), IEEE』, 297–304.

인터넷 사이트

Google. (2017). 『Volumetric Shader』. 검색일자: 2018. 01. 14.

<http://www.shadowmint.com/articles/code-volumetric-shader-1/>

Google. (2018). 『Unity』. 검색일자: 2018. 01. 14.

<https://unity3d.com/kr>

ABSTRACT

Construction of medical simulation using commercial game engine

Ha, Tae-Jun

Major in Information System Engineering

Dept. of Information System Engineering

The Graduate School

Hansung University

As the functions of commercial game engines have improved, attempts to use it for general purpose research have continued. This study attempts to use game engines to develop medical simulations such as clinical operations. When using a game engine, there is an advantage that various functions required for an application are basically provided, such as depth sorting of translucent objects or virtual reality hardware support. On the other hand, there is a restriction that the structure of the application program should be modified to suit the characteristics of the game engine. Therefore, we explain how to visualize medical volume data according to the structure of game engine, and propose a method that can promptly respond to user's demand and help observation. Specifically, when the user selects an important portion of the human body, the

focus-context visualization, which exhibits complex effects by changing the visualization parameters for the selected portion, is applied to the volume data. In the process, this study proposes the accumulation based focus+context volume visualization method. Proposed method is to make the background part more transparent and naturally fuse it with the focus part. Since the proposed method combines well with the existing volume visualization, the virtual surgery function such as incision of volume is performed smoothly. As a result, we created a virtual reality scene composed of surface data and medical volume data pieces together, and developed an application that responds to input events. A game engine with a variety of built-in functions can help develop medical simulation production tools.

【KEYWORD】 Medical Image Rendering, Game Engine, Unity3D, Volume Rendering, Focus Context Rendering, Virtual Reality