

CPU 기반의 볼륨 변형을 위한 다형질 Chainmail 모델

이세인[†], 계획원^{††}

Heterogeneous Chain-mail Model for CPU-based Volume Deformation

Sein Lee[†], Heewon Kye^{††}

ABSTRACT

Since a surgery simulation should be able to represent the internal structure of the human body, it is advantageous to adopt volume based techniques rather than polygon based techniques. However, the volume based techniques induce large computation to deform heterogeneous volume datasets such as bones and muscles. In this study, we propose a new method to deform volume data using multi-core CPUs. By improving previous studies, the proposed method minimizes unnecessary propagation operations. Moreover, we propose an efficient task-partitioning method for volume deformation using multi-core CPUs. As a result, we can simulate the deformation of heterogeneous volume data at an interactive speed without special hardware.

Key words: Virtual Surgery, Volume Deformation, Chainmail, Parallel Computing, OpenMP

1. 서 론

현대 컴퓨터 하드웨어 성능이 발전해감에 따라 가상 의료수술에 대한 연구와 적용이 활발히 진행되고 있다. 가상 수술의 방식은 표면(surface) 기반의 방식과 볼륨(volume) 기반의 방식으로 구분할 수 있는데, 전통적인 표면 기반의 방식은 인체 장기 모델을 삼각형 메시로 구축하고 시뮬레이션과 가시화를 수행하는 것이다. 표면 기반의 방식에서는 메시의 절단 처리가 기술적인 난제로 알려져 있다. 가상 수술에서는 피부를 절개하여 피하조직을 드러내거나 종양을 잘라내 제거하는 등의 작업이 수행된다. 이때, 실시간으로 수술 절단면을 삼각형 메시로 재구축하고, 거동 모델에 절단된 개체를 반영하는 것이 어려움이 있다.

이와 같은 문제를 극복하기 위해 볼륨 기반의 조작(manipulation)과 렌더링 방식[1]이 제안되었다. 볼륨 데이터는 CT, MR 데이터와 같이 삼차원 배열 형태로 구성되며 다수의 이차원 단면영상이 모인 데이터이다. 볼륨 데이터의 변형은 볼륨 데이터를 구성하는 입자인 복셀(voxel)을 조작하여 표현하므로, 가상적으로 인체 조직을 절개하더라도 내부가 자연스럽게 표현된다. 다만 볼륨 데이터는 수 백 만개 이상의 많은 복셀로 구성되어 있기 때문에 가상수술 시뮬레이션을 대화적 속도로 구현하는 것은 어려움이 있다.

볼륨 기반으로 빠르게 변형체 모델을 생성하는 Chainmail[2] 알고리즘은, 물체를 구성하는 입자들 서로 연결된 쇠사슬의 알갱이로 모델링하여 변형체

* Corresponding Author : Heewon Kye, Address: (02876) 116, Samseongyo-ro 16-gil, Seongbuk-gu, Seoul, Korea, TEL : +82-2-760-8014, E-mail : kyei@hansung.ac.kr

Receipt date : Apr. 22, 2019, Revision date : June 3, 2019
Approval date : June 19, 2019

[†] Division of Computer Engineering, Hansung University
(E-mail : foryx1992@naver.com)

^{††} Division of Computer Engineering, Hansung University

* This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(Ministry of Science and ICT) (No. 2017R1E1A1A03070494) (Sein Lee). This Research was financially supported by Hansung University (Heewon Kye)

의 움직임을 모사하였다. 이 방법은 사용자가 근원점을 이동하면, 인접 쇠사슬에 연속적으로 이동을 전파해나간다. 데이터가 동일한 물성(homogeneous)의 볼록한(convex) 형태이면, 이동의 전파를 한 번만 수행하여 안정한 상태에 도달하게 된다.

다만, Chainmail 알고리즘은 절개가 불가능하고, 근육과 골격을 같이 조작하는 다형질(heterogeneous) 데이터에 적용 불가능한 문제가 지적되었다. 다형질 데이터에 절개가 가능한 새로운 병렬 Chainmail 알고리즘은 HP-Chainmail[3]이다. 최초 조작점에서 시작하는 전파를 여러 번 반복 수행하여 기존 문제를 해결하였으며, 그로 인한 계산 속도의 증가 문제는 GPU를 이용한 병렬 알고리즘으로 해소하였다. 다만, GPU를 이용하여 고속 연산을 수행하자면 별도의 장치가 필요하며 상대적으로 적은 메모리 및 기존 시스템과 접목과 같은 시스템 구축의 복잡성을 고려하여야 한다.

본 연구는 특별한 GPU 장치 없이 일반적인 CPU에서 고속 절개 및 변형이 가능한 방법을 제안한다. 본 연구는 기존의 HP-Chainmail과 유사하게 여러 번 전파를 수행하여 절개 및 다형질 데이터에 적용이 가능하다. 그리고 CPU의 연산속도가 GPU에 비해 부족하므로 전파 횟수를 크게 개선하는 새로운 방법을 제안한다. 그리고 CPU의 다중 코어를 활용하여 대화적 속도로 볼록 절개 및 변형을 수행한다.

본 논문의 구성은 다음과 같다. 2장에서 기존의 볼록 기반 변형체 연구에 대해 분석하며, 3장에서는 제안 방법인 전파 횟수를 개선하는 Chainmail 방법을 설명한다. 4장에서 제안 방법에 CPU의 다중 코어를 효율적으로 이용하는 방법을 설명하며, 5장에서 실험 결과를 보이고 6장에서 결론을 맺는다.

2. 관련 연구

물체의 조작을 모델링 하는 데에 기본적인 방법(naïve method)은 유한요소 모델링(finite element modeling) 기법이다. 이와 관련하여 연조직[4], 두개골[5], 안구[6] 등을 모델링 한 연구가 선행되었다. 유한요소 모델링은 직관적인 방법이지만, 시뮬레이션 시간이 매우 오래 걸리므로 대화식 프로그램에서는 사용하기 어렵다. 몇몇 연구에서는 멀티그리드 방법(multigrid method)이나 모달 해석(modal analysis)과 같은 동역학 수치해석 기법을 적용하여 연산

량을 줄이기 위해 노력하였으나[7, 8, 9], 계산 복잡도는 여전히 병목으로 남아있다.

또한 질량-스프링 시스템(mass-spring system)도 빈번하게 사용된다. 물체를 질량이 있는 입자들이 서로 스프링으로 연결된 상태로 여겨, 외력이 가해지면 실시간으로 변형 정도를 계산한다. Provot의 논문[10]에서 최초로 소개되었으며, 인체 시뮬레이션에 관련하여서는 안면 조직[11], 장기[12], 부피가 유지되는 연조직[13] 등을 모델링 한 연구가 선행되었다. 그러나 질량-스프링 시스템은 스프링 특성상 한번 외력이 가해진 시스템은 다시 안정 상태로 돌아가는데 매우 오랜 시간이 소요되므로 사용할 수 있는 입자의 수가 매우 작다는 단점이 있다.

Chainmail[2] 알고리즘은 물체를 Fig. 1과 같이 고리(ring)들이 서로 연결된 그물망으로 모사한 모델이다. 사용자는 특정 고리 하나를 선택하여 움직이며 이 최초 조작점을 근원점(epicenter)이라고 한다. 근원점은 조작이 가해진 만큼 위치가 이동되며, 이후 연결되어있는 모든 입자들이 근원점을 따라 사슬과 같이 움직이게 된다.

각 고리들은 인접 고리와 너무 가까워지거나 멀어지지 않도록 거리 범위가 부여된다. 조작 연산 중 제한 범위를 벗어나는 고리가 있다면, 이웃을 당기거나 밀어내어 다음 연산에서 위치를 조정한다. 이를 이동의 전파라고 하며, Chainmail 알고리즘은 입자의 방문 순서를 일관하게 유지하여 각 고리에 대해 1회의 이동 연산(이동 전파 1회)으로 전체 물체가 안정상태가 된다는 점을 증명하였다. 복셀이 이동할 위치를 항상 결정적으로 계산하기 때문에 스프링과 같이 반작용과 관성에 의한 에너지 발산 과정이 없고 신속하게 안정 상태를 결정할 수 있다.

3D Chainmail은 이를 삼차원으로 자연스럽게 확장한 모델이며, 이를 의료 시뮬레이션에 적용하기 위

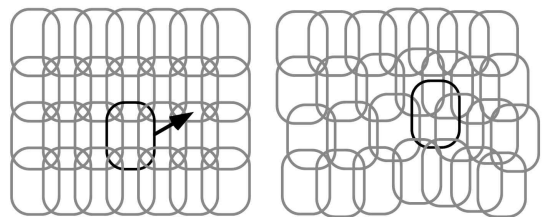


Fig. 1. Manipulating object with the 2D Chainmail model [2].

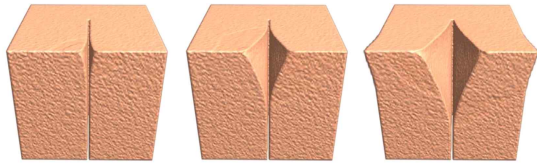


Fig. 2. The skin was incised and pulled on both sides using HP-Chainmail algorithm[3].

해 다양한 방법이 제안되었다. 단일한 물성만을 지원하는 단점은 Enhanced Chainmail[14]이 해결하였는데, 단단한 구성 성분이 먼저 이동하여 자리를 잡고, 부드러운 물체는 그 이후에 배치하는 방법을 사용하였다. 이를 구현하기 위해 우선순위 큐(queue)를 두어 거리 제한 값을 기준으로 전파 순서를 정렬하였다. Chainmail의 다른 문제는 구멍이 있거나 잘리는 형태를 지원하지 않았는데, Generalised Chainmail[15]은 이러한 문제점을 개선하여 임의의 위상의 물체에도 적용 가능하다.

Enhanced Chainmail과 Generalised Chainmail은 각각 다중 물성과 복잡한 위상의 처리라는 기능을 가지고 있지만, 알고리즘 특성상 동시 적용은 불가능하다. 이렇게 상호배타적인 두 기능은 이후 HP-chainmail[3]에서 통합되었다. 게다가 HP-Chainmail 알고리즘은 GPU로 병렬화되어 연산을 빠르게 수행한다. 알고리즘 병렬화에 대한 아이디어는 SP-Chainmail[16]에서 처음으로 제안되었으며, 이후 HP-Chainmail에서 활용되었다.

한편, HP-Chainmail 알고리즘은 복셀의 연산 순서가 정해지지 않으므로, 근원점으로부터 조작성 전파받지 못한 복셀이 무의미한 이동연산을 반복 수행하는 구조이다. 최악의 경우 한 번의 이동 전파로 전체 복셀의 연산을 수행해도, 실제 의미 있는 정보는 한 복셀밖에 전진하지 못하는 경우도 있다. 이 때문에 근원점으로부터 볼륨의 가장자리로 전파가 도달할 때까지 이동 전파를 다수 반복해야 한다. 본 연구는 이동 전파의 횟수를 크게 줄여 성능을 향상시키는 방법을 제안한다.

3. 이동전파 개선을 위한 8방향 분리 진행

본 연구에서는 새로운 이동 전파 알고리즘인 8방향 분리 방법을 제안한다. 8방향 분리는 CPU 환경의 이점을 활용해 조작성의 전파 순서를 효율적으로 제어

하는 방법이다. 각 복셀의 이동 전파는 상호 간 독립적으로 수행되지 않으며, 적절한 중속성을 갖는다.

본 연구는 먼저 Fig. 3과 같이 볼륨을 근원점(epicenter)을 기준으로 논리적으로 8 분할한다. 새로 생긴 자식 볼륨들은 Fig. 4와 같이 근원점으로부터 볼륨의 외곽을 향하여 퍼져나가는 방향 순으로 전파 방향을 정렬하였다. 이와 같은 정렬은 Fig. 5와 같이 x, y, z축의 각 방향에 대한 증감을 다르게 하여 수행하였다. 우선, 근원점의 좌표 epicenter(x, y, z)를 시작으로 x 축에 나란한 방향으로 순회를 시작한다. 이후에도 근원점 혹은 이미 이동 전파가 완료된 복셀과 인접한 복셀에서 순회를 계속한다. 이러한 방식으로 조작성의 전파를 수행하면, 항상 이웃에 방금 갱신된 정보가 있다는 것이 보장되므로 불필요한 연산을 줄일 수 있다.

Fig. 6은 제안 방법의 효과를 설명하기 위해 8방향 분리 전략(좌측)과 HP-Chainmail의 이동 전파 순서(우측)를 비교하여 표현한 것이다. 초록색 복셀은 사

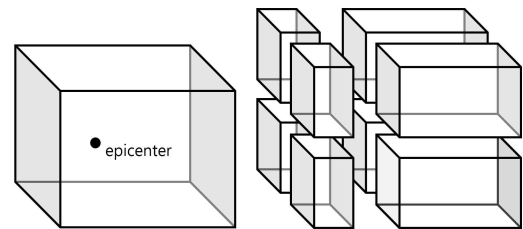


Fig. 3. 8-partition process of volume, the volume is divided into eight zones based on the epicenter (x, y, z).

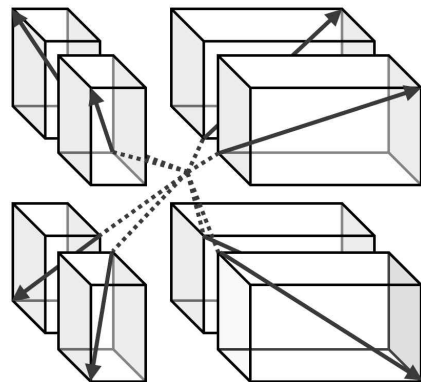


Fig. 4. Manipulation propagation of each voxel in eight child volumes is performed in the outward direction of the volume from the origin.

Child Volume 1	Child Volume 2
for (z=epicenter.z; z>=0; z--) for (y=epicenter.y; y>=0; y--) for (x=epicenter.x-1; x>=0; x--) deform(volume[z][y][x]);	for (z=epicenter.z; z>=0; z--) for (y=epicenter.y; y>=0; y--) for (x=epicenter.x; x<volume.width; x++) deform(volume[z][y][x]);
Child Volume 3	Child Volume 4
for (z=epicenter.z; z>=0; z--) for (y=epicenter.y+1; y<volume.height; y++) for (x=epicenter.x-1; x>=0; x--) deform(volume[z][y][x]);	for (z=epicenter.z; z>=0; z--) for (y=epicenter.y+1; y<volume.height; y++) for (x=epicenter.x; x<volume.width; x++) deform(volume[z][y][x]);
Child Volume 5	Child Volume 6
for (z=epicenter.z+1; z<volume.depth; z++) for (y=epicenter.y; y>=0; y--) for (x=epicenter.x-1; x>=0; x--) deform(volume[z][y][x]);	for (z=epicenter.z+1; z<volume.depth; z++) for (y=epicenter.y; y>=0; y--) for (x=epicenter.x; x<volume.width; x++) deform(volume[z][y][x]);
Child Volume 7	Child Volume 8
for (z=epicenter.z+1; z<volume.depth; z++) for (y=epicenter.y+1; y<volume.height; y++) for (x=epicenter.x-1; x>=0; x--) deform(volume[z][y][x]);	for (z=epicenter.z+1; z<volume.depth; z++) for (y=epicenter.y+1; y<volume.height; y++) for (x=epicenter.x; x<volume.width; x++) deform(volume[z][y][x]);

Fig. 5. Manipulation propagations expressed in pseudocode.

용자가 선택한 근원점이다. 주황색 복셀은 현재 반복(iteration)에서 위치 갱신이 완료된 복셀을 의미하고, 노란색 복셀은 현재 갱신 중인 복셀을 의미한다.

제안 방법인 Fig. 6(a)에서 근원점의 큰 이동이 발생하였다면, 그 이동 정보가 순차적으로 인접 복셀을 통해 전파된다. 이렇게 근원점에서 출발한 이동 정보가 x축 방향으로 전파되고, 각 복셀의 이동 값이 갱신되며 연쇄적으로 전달된다. 다음 행에도 근원점과 인접한 복셀부터 갱신이 수행되므로, 매번 복셀의 움직임은 새로운 정보를 이용한 것이다.

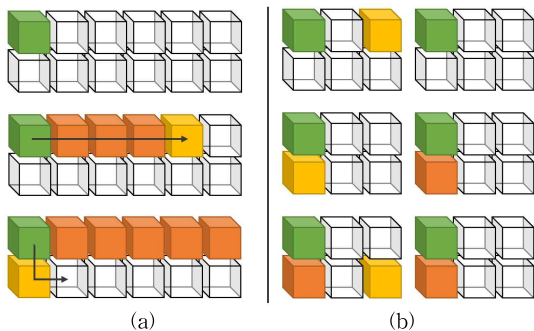


Fig. 6. The process of manipulation propagation out of the epicenter, (a) is the 8-way separation method, and (b) is the HP-Chainmail propagation method.

이에 반해 HP-Chainmail은 각 복셀에서의 병렬 처리를 최우선으로 고려하였기 때문에, 위치 갱신 연산이 각 복셀 상에서 순서 없이 수행된다. 이러한 경우 Fig. 6(b)와 같이 대부분의 복셀이 무의미한 연산을 수행하게 된다. 모든 복셀에 대한 연산을 수행한 후, 확실하게 새로운 정보로 갱신되었다고 보장할 수 있는 복셀은 근원점에 직접 x, y, z축 방향으로 이웃한 6개 복셀뿐이다. HP-Chainmail 방식으로 모든 복셀에 유효한 위치 정보를 전파하기 위해서는, 최악의 경우 근원점을 기준으로 볼륨 꼭지점까지 맨해튼 거리(manhattan distance) 만큼 이동 전파 알고리즘을 반복해야 한다.

한편, 제안 방법과 초기 Chainmail[2]방법을 비교하면, 초기 Chainmail방법은 이동을 일정한 방향으로 우선적으로 전파하여 본 제안방법과 유사하다. 그러나 기존 방법은, 전체 볼륨 데이터에 대해 단일한 큐(queue)를 두어 전파 순서가 결정되므로 병렬화가 어렵다는 단점이 있으며 2장에서 설명한 바와 같이 절개 및 다형질 데이터에 적용이 어렵다는 문제가 있다.

참고로 제안 방법이 1회의 전파로 데이터의 수렴이 가능한 것은 아니다. 예를 들어, Fig. 6(a)의 첫 번째 행이 신축성이 좋은 형질, 두 번째 행이 강체와

같이 단단한 형질인 다형질 데이터를 가정하고 설명한다. 첫 번째 행은 근원점의 큰 이동이 중간 복셀에서 점차적으로 흡수되어 행의 마지막 부분에서는 거의 이동이 없게 된다. 두 번째 행은 근원점의 이동에 따라 마지막 부분까지 크게 이동해야 하므로 점차적으로 첫 번째 행과 두 번째 행의 차이가 크게 벌어진다. 이런 경우 근원점부터 이동 전파를 새로 수행하여, 이미 많이 이동한 두 번째 행의 정보를 첫 번째 행이 참고할 수 있게 된다.

이때, 첫 번째 전파와는 달리 두 번째 전파에서는 첫 번째 행이 두 번째 행의 이동 정보를 참고해야 하는데, 이는 HP-Chainmail에서 제안된 타임 스탬프(timestamp)기법을 이용한다. 이 방법은 각 복셀을 서로 연결된 그래프의 노드로 생각하고, 각 노드에 타임 스탬프를 부여하는 방법이다. 근원점의 타임 스탬프를 0으로 가정하고, 이웃 복셀로 전파될수록 타임 스탬프 값이 증가한다. 빠와 같이 단단한 물체로 전파되는 경우 작은 값이 증가하고, 부드러운 조직으로 연결된 경우 큰 값이 증가한다. 따라서, 임의의 노드에서 움직임은, 인접 노드 중에서 빠른 타임 스탬프 값의 노드만 참고한다. 그 결과는 먼저 물체가 먼저 위치를 결정하고 인접한 부드러운 물체가 이에 종속되어 위치하게 되므로 각 노드간의 이동 종속성 문제가 해결된다.

제안 방법 Fig. 5의 위치 갱신 함수인 deform()은 Fig. 7과 같은 절차로 수행한다. 이웃한 6개의 복셀에 대해 (라인 2), 이웃의 현재 타임 스탬프와 이웃에서 자신에게 연결되는 시간을 합한다(라인 3). 이 문장은 근원점에서 시작되어 이웃 복셀을 경유한 뒤, 현재 복셀로 도착하기까지 걸린 시간(newTimeStamp)을 계산하는 식이다. 기존 경로에 비해 새로운 경로로 얻은 시간이 더 짧다면, 타임 스탬프를 갱신하고(라인

```

1: deform(voxel) {
2:   for (neighbor : voxel.neighbors) {
3:     newTimeStamp = neighbor.timeStamp +
       getConstaint(voxel, neighbor);
4:     if (newTimeStamp < voxel.timeStamp) {
5:       voxel.timeStamp = newTimeStamp;
6:       if (isPositionViolated(voxel, neighbor))
7:         moveToNonViolatingPosition(voxel);
8:   }}}

```

Fig. 7. Pseudocode for location update function.

5) 안전한 곳으로 자신을 이동한다(라인 6, 라인 7).

제안 방법은 다형질 조직의 분포에 따라 여러 번의 이동 전파가 이루어질 수 있으나, 항상 의미있는 정보가 전파되도록 순서가 조절된다. HP-Chainmail이 순서 없이 갱신을 반복하여 수백 번의 이동 전파를 반복하는데 비해 제안 방법은 크게 성능이 향상된다. 그리고 제안 방법은 HP-Chainmail의 우수한 기능인 볼륨 절개 및 다형질 데이터 지원도 동일하게 수행할 수 있다.

4. CPU 기반 다중 스레드 환경에서 효율적 작업 분할

이번 장에서는 3장에서 제안한 방법을 다중 코어 CPU에서 효율적으로 수행하는 방법에 대해 설명한다. 기본적인 병렬화 방법은 8개의 자식 볼륨마다 각각 별개의 스레드를 할당하는 것이다. 그러나 이 방법은 몇 가지 비효율적인 점이 있으므로, 이를 설명하고 슬라이스 기반의 새로운 병렬화 방법을 제안한다.

Fig. 8은 자식 볼륨에 대해 별개의 스레드를 할당하는 방법을 모사한 것이다. 이 방법의 첫 번째 문제는 작업 부하(load)의 불균형이다. 근원점이 볼륨의 정중앙이 아닌 경우 일부 스레드에만 많은 작업이 할당되는 문제가 발생한다. 그리고 또 다른 문제는 스레드간 경계면에서 이동 전파가 원활하지 않다는 점이다. Fig. 8에서 근원점은 스레드 8, 즉 자식 볼륨 8에 포함되어 있으며, 자식 볼륨 사이의 이동정보 전달은 비효율적인 스레드간의 통신을 이용해야 한다. 한편, 스레드간의 통신을 제거하기 위해 이동 정보를 스레드들이 함께 사용하는 메모리로 설정하는 방법도 고려할 수 있으나, 자식 볼륨 사이의 인접 면에서 메모리를 읽고 쓰기 위해서는 Fig. 7의 조작을 스레

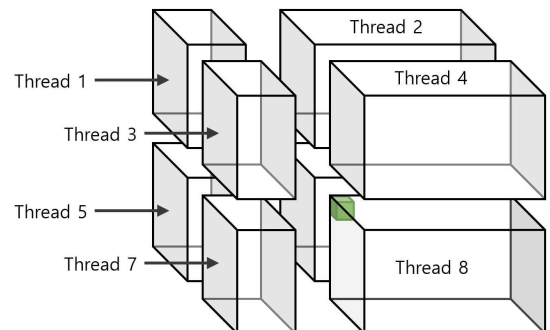


Fig. 8. 8-way separation with multi-thread.

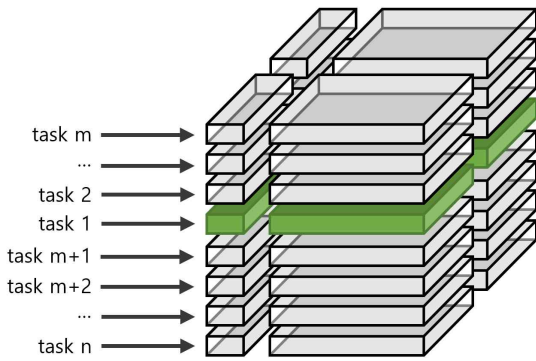


Fig. 9. Explanation of task-splitting Strategy.

드 중첩 쓰기로부터 보호하는 원자화(atomize)를 해야 하는 비효율 문제가 발생한다.

이와 같은 문제들을 해결하고자 본 연구는 각 스레드가 담당하는 영역을 Fig. 9와 같이 평면 단위로 분할하는 방법을 제안한다. 우리는 근원점이 존재하는 x-y 평면에 대한 전파를 1번 작업(task 1), z 방향으로 한 칸 위쪽의 x-y 평면에 대한 전파를 2번 작업(task 2) 등으로 설계하였다. 그리고 1번 작업은 하나의 스레드가 전담하여 처리하되, Fig. 4의 전파 순서가 유지될 수 있도록 4개의 자식 평면으로 논리적으로 분할하여 순차적으로 연산 하였다. 마찬가지로 2번 작업(task 2)도 같은 방법으로 하나의 스레드가 전담하여 처리하며, 이 흐름을 Fig. 10에 의사 코드로써 나타내었다.

제안 방법은 3장에서 설명한 이동 전파 방법을 유

지하면서, 효율적인 병렬화가 가능하다. 이 작업을 Fig. 11에 예를 들어 도시하였다. 각 복셀에 쓰인 번호는 각 스레드별 이동 전파 순서를 표시한 것이다. 각 스레드 상에서 가장 먼저 처리되는 복셀은 근원점 좌측 복셀(스레드 1)이며 높이가 다른 스레드 2, 스레드 3의 경우도 동일한 위치이다.

하나의 스레드 관점에서, 동일 x-y 평면상의 갱신 순서는 Fig. 11에 도시된 바와 같이 Fig. 4와 결과적으로 동일한 순서를 유지한다. 근원점에서 평면의 네 꼭지점 방향으로 각각 순차적으로 전파된다. 한편, 높이 방향으로 x-y 평면들을 계산하는 순서는 OpenMP[17]의 동적 스레드 스케줄링을 이용하여 실행 순서를 가급적 보장하려고 노력하였다. 동적 스레드 스케줄링은 이전 높이 방향의 작업이 끝나야 다음 높이 방향의 새로운 작업을 스레드에 부여하므로, 작업 순서가 되도록 보장된다.

한편, 극단적인 예로 스레드가 시작하여 첫 번째 복셀을 처리하기 전에 정지되는 경우도 가정할 수 있다. CPU의 스레드 스케줄링은 프로그램 외부의 영향을 받는 경우도 있기 때문에, 높이 방향의 실행 순서는 완전하게 보장되지 않을 수 있다. 병렬화 결과, 제안 알고리즘은 추가적인 이동 전파가 발생할 수 있으며, 타임 스템프를 이용하여 결과 영상에 오류는 없지만 이동 전파 횟수가 증가하는 손실이 발생할 수 있다.

제안한 평면 단위 작업 분할은 자식 불륨마다 스레드를 할당하는 방법과 비교해 보았을 때, 임계 영

Task 1	
<pre>z=epicenter.z; for (y=epicenter.y; y>=0; y--) { for (x=epicenter.x-1; x>=0; x--) deform(voxel[z][y][x]); for (x=epicenter.x; x<volume.width; x++) deform(voxel[z][y][x]); }</pre>	<pre>for (y=epicenter.y+1; y<volume.height; y++) { for (x=epicenter.x-1; x>=0; x--) deform(voxel[z][y][x]); for (x=epicenter.x; x<volume.width; x++) deform(voxel[z][y][x]); }</pre>
Task 2	
<pre>z=epicenter.z-1; for (y=epicenter.y; y>=0; y--) { for (x=epicenter.x-1; x>=0; x--) deform(voxel[z][y][x]); for (x=epicenter.x; x<volume.width; x++) deform(voxel[z][y][x]); }</pre>	<pre>for (y=epicenter.y+1; y<volume.height; y++) { for (x=epicenter.x-1; x>=0; x--) deform(voxel[z][y][x]); for (x=epicenter.x; x<volume.width; x++) deform(voxel[z][y][x]); }</pre>

Fig. 10. Pseudocode for a single task.

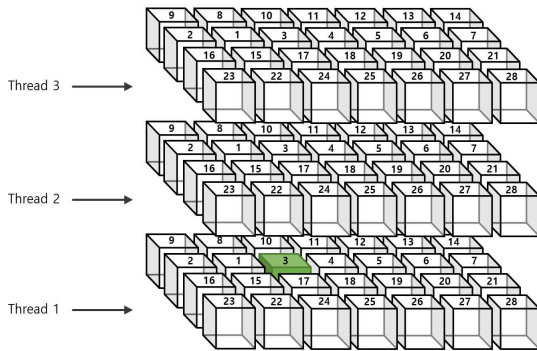


Fig. 11. Planar unit task distribution and manipulation propagation of each thread.

역을 설정하지 않아도 된다는 장점을 가진다. 이는 Fig. 10과 Fig. 11에서 나타내는 바와 같이 스레드 상호 간 중복되어 메모리에 저장하는 영역(복셀)이 없기 때문이다. 따라서 각 스레드는 위치 갱신 연산 시 상호 배제(mutual exclusion)를 고려하지 않고 최대의 효율로 작업을 처리할 수 있다. 게다가 제안 방법은 8개의 고정 작업 분할이 아니므로, 현대의 CPU와 같이 논리적으로 6스레드 혹은 12스레드를 사용하는 경우에도 유연하게 적용할 수 있다.

5. 실험 결과

본 실험은 Intel i7-8700K CPU, 32GB RAM 환경에서 수행되었다. 실험에 사용한 볼륨 데이터는 Head, Abdomen, Cardiac, LowerExtremity로, 총 4개의 CT 영상이다. 이들의 데이터를 Fig. 12로 가시화하였다.

볼륨 데이터에 대한 세부 정보는 Table 1과 같다. 각 볼륨 데이터는 사용자에게 실시간 환경을 제공하기 위해, 볼륨의 크기를 축소한 별도의 거동 모델을 가지도록 구현하였다. 축소한 거동 모델의 크기는 대략 128³의 크기를 가지도록 설정하였다.

사용자가 데이터의 임의의 부분을 미리 절개하여 둔

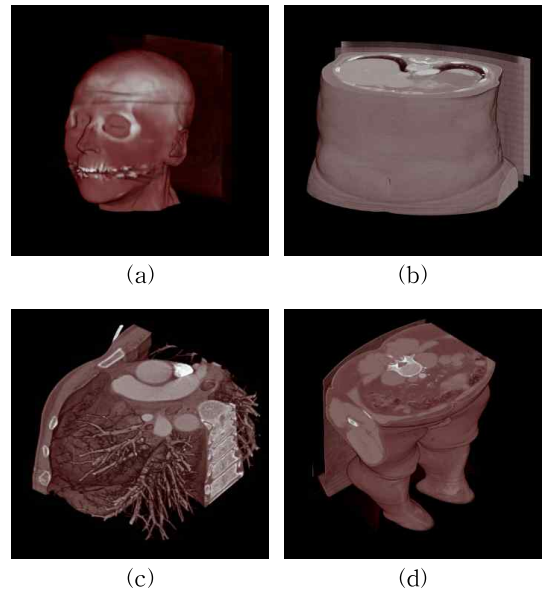


Fig. 12. The CT images in the experiment. (a) Head, (b) Abdomen, (c) Cardiac, and (d) LowerExtremity.

후, 근원점을 움직여 거동한 결과 영상을 Fig. 13에 보이고 있다. 다양한 방향의 절개 및 이동이 자연스럽게 적용되고 있는 것을 파악할 수 있다. 그리고 이러한 조작 과정에서의 성능을 측정하여 기존의 HP-Chainmail과 비교하였다.

첫 번째 실험은 3장에서 설명한 전파의 효율을 측정하기 위해, 단일 스레드로 제안 방법과 HP-Chainmail을 비교하였다. 본래 HP-Chainmail은 GPU 상에서 병렬 처리를 위해 만들어진 알고리즘이므로, 이를 CPU의 단일 스레드로 표현하기 위해 HP-Chainmail의 이동 전파는 순차적으로 처리되도록 조정하였다.

Table 2는 단일 스레드 상에서 8방향 분리 알고리즘과 HP-Chainmail의 성능을 비교한 것이다. HP-Chainmail은 50회 이상의 많은 이동 전파(The Number of Iterations)가 수행되며 본 연구는 3장에서 제

Table 1. Details of the volume data used in the experiment

	Image Name			
	Head	Abdomen	Cardiac	LowerExtremity
Image Size	256×256×225	512×512×326	512×512×318	512×512×580
Behavioral Model Size	128×128×112	128×128×81	128×128×79	128×128×145
Model Reduction Ratio	0.5	0.25	0.25	0.25

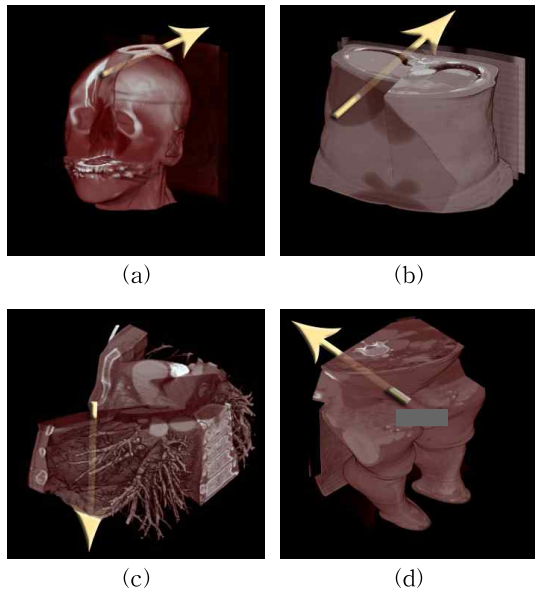


Fig. 13. Manipulating Volume Data.

안한 방법을 이용하여 이동 전파의 횟수를 기존 방법에 비해 8배 이상 향상시켰다. 그 결과 전체 알고리즘 수행 시간도 6배 이상 향상되었다.

두 번째 실험은 4장에서 설명한 다중 스레드의 효율을 실험하기 위해 제안 방법과 HP-Chainmail을 비교한 실험이다. 스레드의 수를 제외한 모든 환경은 첫 번째 실험과 동일하며 스레드의 작업 분할은 4장

의 제안 방법과 같이 평면 단위로 분할을 사용하였다. 스레드의 수는 실험에 사용한 CPU의 논리 코어 개수에 따라 12개로 설정하였다.

Table 3은 다중 스레드 상에서 제안 방법과 HP-Chainmail의 성능을 비교한 것이다. 다중 스레드를 사용한 결과 이동 전파(The Number of Iterations)가 Table 2에 비해 다소 증가한 것을 알 수 있다. 이미 4장에서 설명한 바와 같이, 작업 스레드는 나중에 생성되어도 먼저 생성된 스레드보다 진행이 더 빨라지는 경우가 있다. 이렇게 진행이 역전되는 경우 나중에 생성된 스레드가 유효한 조작 정보를 전파 받지 못해 추후 추가적인 반복이 필요하게 되는 경우가 생기기 때문에 파악된다. 그러나 그 정도는 크지 않아서 10% 이내(1.1배)의 비효율(overhead)이 발생하는 것으로 나타난다.

다중 스레드를 이용한 성능 향상은 단일 스레드 대비 3-4배 정도이다. 제안 방법의 성능 향상은 병렬화에 적합한 HP-Chainmail의 성능 향상과 비슷한 결과로, 제안 방법이 병렬화에 적합한 것을 파악할 수 있다. 그 결과 볼륨의 조작에 소요되는 시간이 100ms 내외이므로 사용자에게도 대화적 환경을 제공할 수 있다.

참고로, 1344개의 연산코어를 내장한 GeForce GTX 670 GPU를 이용한 HP-Chainmail의 속도는 완충(relaxation)작업을 포함하여 53ms가 소모되는데,

Table 2. Performance comparison between proposed algorithm and HP-Chainmail using a single thread

	Algorithm							
	Proposed Algorithm				Single thread HP-Chainmail[3]			
	Head	Abdomen	Cardiac	Lower Extremity	Head	Abdomen	Cardiac	Lower Extremity
Behavioral Model Size	128×128×112	128×128×81	128×128×79	128×128×145	128×128×112	128×128×81	128×128×79	128×128×145
The Number of Iterations	7.0	6.9	8.1	8.0	58.2	66.5	64.7	74.3
Iteration Performance Improvement against Existing algorithm	8.31	9.64	7.99	9.29	N/A			
Elapsed Time (Unit: ms)	384.52	268.20	293.32	575.63	2385.97	1913.78	1833.26	3963.67
Elapsed Time Performance Improvement against Existing Algorithm	6.20	7.13	6.25	6.89	N/A			

Table 3. Performance comparison between proposed algorithm and HP-Chainmail using multiple threads on CPU.

	Algorithm							
	Proposed Algorithm				HP-Chainmail[3] on CPU			
	Head	Abdomen	Cardiac	Lower Extremity	Head	Abdomen	Cardiac	Lower Extremity
Elapsed Time (Unit: ms)	98.42	63.74	77.76	152.27	658.24	513.84	486.89	1012.73
The Number of Iterations	7.2	7.0	8.9	8.7	66.1	73.7	71.2	80.6
Performance of Iterations against a Single Thread	1.03	1.01	1.10	1.09	1.14	1.11	1.10	1.08
Performance Improvement against a Single Thread	3.91	4.21	3.77	3.78	3.62	3.72	3.76	3.91
Performance Improvement against Existing Algorithm	6.69	8.06	6.26	6.65	N/A			

1.4M개의 노드에 10회의 고정된 이동전과만을 수행한 결과이다[3]. 본 연구의 Abdomen 데이터가 1.26 M개의 노드에 12개의 논리코어가 내장된 CPU만을 이용하여 63ms가 소모된 것과 비교하면, 본 연구가 대단히 효율적인 방법임을 파악할 수 있다.

6. 결 론

본 연구는 의료 시뮬레이션에서 사용되는 볼륨 데이터의 절개 및 이동 연산을 CPU 기반으로 구현하였다. 기존의 GPU 기반의 HP-Chainmail에 착안하여, GPU를 사용하기 어려운 환경에서 다중 코어 CPU 기반의 Chainmail 모델을 개발하였다. 본 논문에서 제안한 모델의 핵심 알고리즘인 8방향 분리는 기존 알고리즘과 대비하여 근원점에서 비록된 조작의 전파를 볼륨의 말단까지 효율적으로 전달한다. 그 결과 적은 수의 이동 전파로 전체 데이터의 위치가 빠르게 수렴된다. 추가적으로 본 연구는 이 방법을 CPU 기반의 다중 스레드 환경으로 확장하면서 평면 단위 작업 분할을 제안하였다. 제안 방법을 사용하여 효율적인 병렬화가 가능하였고, 그 결과 사용자에게 대화적 조작 환경을 제공할 수 있게 되었다.

본 연구의 한계로, 물리적인 6코어 12스레드의 CPU를 사용한 성능 향상이 3~4배에 그친 것은 Chainmail 알고리즘의 이동 전파 연산이 상대적으로

간단하여, 메모리 참조 시간이 총 연산에 영향을 주는 것으로 판단된다. 따라서 CPU 기반의 멀티코어 시스템[18]에서는 메모리 참조 최적화 또는 캐시 메모리의 사용법 향상을 향후 연구로 수행할 필요가 있다. 한편, 본 연구에서 제안한 효율적인 이동 전파 방법을 GPU에서도 사용할 수 있도록, 병렬화 구조가 다른 GPU 프로그래밍으로 적용하거나 CPU와 GPU가 협업[19]하는 연구도 수행 예정이다.

REFERENCE

- [1] L. Marc, "Display of Surfaces from Volume Data," *IEEE Computer Graphics and Applications*, Vol. 8, No. 3, pp. 29-37, 1988.
- [2] S.F. Gibson, "3D Chainmail: A Fast Algorithm for Deforming Volumetric Objects," *Proceeding of Symposium on Interactive 3D Graphics*, pp. 149, 1997.
- [3] A. Rodríguez, Alejandro, A. León, and G. Arroyo, "Parallel Deformation of Heterogeneous Chainmail Models: Application to Interactive Deformation of Large Medical Volumes," *Computers in Biology and Medicine*, Vol. 79, No. 1, pp. 222-232, 2016.
- [4] R. Alterovitz, K. Goldberg, J. Pouliot, R.

- Taschereau, and I. Hsu, "Needle Insertion and Radioactive Seed Implantation in Human Tissues: Simulation and Sensitivity Analysis," *Proceeding of IEEE International Conference on Robotics and Automation*, pp. 1793-1790, 2003.
- [5] A. Boryor, M. Geiger, A. Hohmann, A. Wunderlich, C. Sander, and S.F. Martin, et al., "Stress Distribution and Displacement Analysis During an Intermaxillary Disjunction—a Three-Dimensional FEM Study of a Human Skull," *Journal of Biomechanics*, Vol. 41, No. 2, pp. 376-382, 2008.
- [6] E.Y.K. Ng and E.H. Ooi, "FEM Simulation of the Eye Structure with Bioheat Analysis," *Computer Methods and Programs in Biomedicine*, Vol. 82, No. 3, pp. 268-276, 2006.
- [7] C.H. Wolters, M. Kuhn, A. Anwander, and S. Reitzinger, "A Parallel Algebraic Multigrid Solver for Finite Element Method Based Source Localization in the Human Brain," *Computing and Visualization in Science*, Vol. 5, No. 3, pp. 165-177, 2002.
- [8] I. Essa, S. Sclaroff, and A. Pentland, "Physically-Based Modeling for Graphics and Vision," *Directions in Geometric Computing*, Information Geometers, UK., 1993.
- [9] D. Metaxas and D. Terzopoulos, "Dynamic Deformation of Solid Primitives with Constraints," *Proceeding of ACM SIGGRAPH Computer Graphics*, Vol. 26, No. 2, pp. 309-312, 1992.
- [10] P. Xavier, "Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behaviour," *Proceedings of Graphics Interface*, pp. 147-154, 1995.
- [11] E. Keeve, S. Girod, R. Kikinis, and B. Girod, "Deformable Modeling of Facial Tissue for Craniofacial Surgery Simulation," *Computer Aided Surgery*, Vol. 3, No. 5, pp. 228-238, 1998.
- [12] C. Paloc, F. Bello, R.I. Kitney, and A. Darzi, "Online Multiresolution Volumetric Mass Spring Model for Real Time Soft Tissue Deformation," *Proceeding of International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 219-226, 2002.
- [13] Y. Duan, W. Huang, H. Chang, W. Chen, J. Zhou, S.K. Teo, et al., "Volume Preserved Mass-Spring Model with Novel Constraints for Soft Tissue Deformation," *IEEE Journal of Biomedical and Health Informatics*, Vol. 20, No. 1, pp. 268-280, 2016.
- [14] M.A. Schill, S.F. Gibson, H. Bender, and R. Manner, "Biomechanical Simulation of the Vitreous Humor in the Eye Using an Enhanced Chainmail Algorithm," *Proceeding of International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 679-687, 1998.
- [15] Y. Li and K. Brodlie, "Soft Object Modelling with Generalised ChainMail—Extending the Boundaries of Web based Graphics," *Computer Graphics Forum*, Vol. 22, No. 4, pp. 717-728, 2003.
- [16] A. Rodríguez, A. León, G. Arroyo, and J.M. Mantas, "SP-ChainMail: A GPU-based Sparse Parallel ChainMail Algorithm for Deforming Medical Volumes," *The Journal of Supercomputing*, Vol. 71, No. 9, pp. 3482-3499, 2015.
- [17] OpenMP, <https://www.openmp.org/> (accessed Apr., 18, 2019).
- [18] W. Lee and H. Kye, "Load Balanced Volume Rendering System for Concurrent Users in Multi-CPU Server Environment," *Journal of Korea Multimedia Society*, Vol. 18, No. 5, pp. 620-630, 2015.
- [19] K. Son, M. Son, and Y. Kim, "Collaboration Method Between CPU and GPU for Generating All Possible Cases in Combination," *Transactions on Computer and Communication Systems*, Vol. 7, No. 9, pp. 219-226, 2018.



이 세 인

2012년 3월~현재 한성대학교 컴
퓨터공학과 학사과정
관심분야: 볼륨 가시화, 영상처
리, 기계 학습



계 희 원

2007년~현재 한성대학교 컴퓨터
공학부 교수
2005년 서울대학교 컴퓨터공학
박사
1999년 서울대학교 전산학사
관심분야: 실시간 가시화, 대용량
의료영상처리, 의료 시물
레이션