

햅틱 볼륨 렌더링을 위한 효과적인 역행렬 계산법

김남오*, 민완기*, 정원태**, 김영동**
 조선이공대학*, 조선대학교**

Effective Inverse Matrix Transformation Method for Haptic Volume Rendering

Nam-Oh Kim*, Wan-Ki Min*, Won-Tae Jung**, Young-Dong Kim**
 Chosun College Scie.&Tech.*, Chosun Univ**.

Abstract - Realistic deformation of computer simulated anatomical structures is computationally intensive. As a result, simple methodologies not based in continuum mechanics have been employed for achieving real time deformation of virtual reality. Since the graphical interpolations and simple spring models commonly used in these simulations are not based on the biomechanical properties of tissue structures, these "quick and dirty" methods typically do not accurately represent the complex deformations and force-feedback interactions that can take place during surgery. Finite Element (FE) analysis is widely regarded as the most appropriate alternative to these methods. However, because of the highly computational nature of the FE method, its direct application to real time force feedback and visualization of tissue deformation has not been practical for most simulations. If the mathematics are optimized through pre-processing to yield only the information essential to the simulation task, run-time computation requirements can be drastically reduced.

To apply the FEM, We examined a various inverse matrix method and a deformed material-model is produced and then the graphic deformation with this model is able to force. As our simulation program is reduced by the real-time calculation and simplification because the purpose of this system is to transact in the real time.

1. 서론

이전 연구에서는 가상현실 환경을 구축하기 위하여 게임과 같은 속이 빈 강체를 가정하고, 그래픽을 중시하는 경향으로 발전되었다[1]. 그러나 수술 시뮬레이션 및 건축분야에서는 유한요소법(FEM)같은 방법을 사용하여 보다 정확하고 정밀한 파라미터를 추출을 목표로 한다. 기존의 게임에서처럼 폴리곤(polygon)방식의 그래픽 대신 두께를 지닌 그래픽 물체(volume)의 형식을 취하여야 한다. 따라서 기존의 구조물을 유한요소법을 적용하기 위한 ANSYS 같은 툴이 개발되었는데, 이 툴들은 실시간 그래픽처리에 적용하기는 곤란하였다. 가상현실에서 유한요소법 기반에 실시간 그래픽 및 햅틱처리를 위해서 연구된 방법중 워싱턴대 제프리 논문에서 그 해법이 제

시되었다. 그러나 이 방법도 몇가지 제한을 주었는데 가장 큰 문제는 역행렬 계산시간으로, 사전에 계산이 필요한 1,000노드(node) 크기의 역행렬을 구하는데 5분 이상의 시간이 걸렸다. 본 연구에서는 이 역행렬 계산의 고속화를 구현하기 위해 연구하였고, 연구결과 계산시간을 1/3로 단축하는 성과를 이루었다.

이 연구가 필요한 이유는 일반적으로 공학용 행렬 연산시 MatLAB 등을 사용하는데 이 툴은 80*80 행렬의 크기를 넘으면 계산이 불가하였고, 워싱턴대 제프리 알고리즘의 경우도 1,000 노드를 넘으면 오류가 발생하는 문제가 있었다. 따라서 10,000 노드의 크기를 지니 섬세하게 표현이 가능한 가상물체를 구현하기 위해 현재 개발된 라이브러리들은 사용할 수 없거나 너무 느렸다. 다양한 실험에서 역행렬 계산시간 이외의 특성은 동일하기 때문에 역행렬 계산에 주안점을 두고 연구하였으며, 연구결과 만족스러운 성과를 얻었다.

본 연구에서 개발된 알고리즘을 이용할 경우 수술 시뮬레이션에서는 봉합수술이 구현가능하고, 건축 및 구조물 분석 등에 적용하여 실시간으로 그 결과를 확인할 수 있다. 특히 역행렬 계산의 고속화를 위해 spacer 등 특수한 행렬의 적용과 배치를 통해 기존에 유한요소법 기반에서 실시간 처리가 어려웠던 상황을 모델링할 수 있으므로 우수한 연구결과임을 확인할 수 있을 것이다.

2. FE 모델링 기법

라그랑지(Lagrangian) 방정식을 이용한 FE 분석에서 지배 행렬식은 다음과 같다.

$$M\ddot{u} + C\dot{u} + Ku = f \quad (1)$$

단, 여기에서 M 은 질량행렬(mass), C 는 감쇠행렬(damping), K 는 강성행렬(stiffness), u 는 노드변위이고 f 는 작용력이다. 각 행렬의 크기는 $n \times n$ 이고, 여기에서 n 은 모델의 노드수에 자유도(이 경우처럼 3차원이면 3)를 곱한 값이다. 정적분석에서, 질량과 감쇠요소는 0으로 지정한다. 각각의 행렬은 FE 메쉬 요소로 구성된다. 지금까지 요소들은 단순한 형상(예:

tetra- hedrons)이었으며, 이는 연속체 역학에서 정의된 성질로 사용되어 왔다. 이 요소들을 모두 도입하여 약간 복잡한 기하구조와 물질의 성질을 행렬식으로 근사화 하였다. FE모델의 정확도는 물체를 모델링한 기하구조를 구성하는 요소 수의 증가에 따른다[2]. 화재나 건축 구조물 해석등에서는 실시간 계산을 필요로 하지 않고 FE분석을 사용하여 예측 값을 도출하기도 한다. 최근 가상현실 연구에서 제프리 등 몇몇 연구자들은 최적의 시뮬레이터에 FE분석을 사용하였는데 포스 피드백과 변형을 제공하기 위해 확장된 분석법을 활용하고 있다[3].

실시간 변형 FE분석법의 흥미로운 통합은 최근 논문 제출이 늘고 있다. 각 노드의 변형은 일련의 텐서들에 의해 컨트롤되고 FE 변형 결과를 기반으로 최적화 되었다. 한 개의 접촉 노드는 x, y, z방향으로 초기에 변형되고, 모든 다른 노드의 이동 변위 결과를 기록된다. 텐서는 최적화 기법을 통하여 개발되었고, 접촉 노드의 위치가 원거리 노드의 위치를 변형시킬 것이다. 각각의 노드는 접촉점의 모든 가능한 수의 많은 텐서로 연관된다. 지금까지 이 방법론은 정확한 FE해를 주지는 않으며 다만 고차원 모델의 근사화된 FE 결과를 제공한다는 것을 의미한다. 노드의 수가 작은 FE 모델은 이미 수술 시뮬레이터에 적용되었고, 대수식의 연립방정식은 행렬연산을 통해 계산된다. 식 (1)의 선형분석에서 $n \times n$ 강성행렬(단, $n =$ 전체 노드수 * 자유도, 3차원인 경우 자유도는 3)이 조합되고, 적용된 힘에 의해 역행렬 및 곱하기 연산을 통해 변위를 계산한다. 이는 정적인 FE방정식을 풀기 위한 고전적인 방법이다. 고전적인 분석에서는 응력과 변형력 요소를 결정하기 위해 추가적인 행렬연산을 포함한다. 강성행렬 K 조합에서 역행렬 및 곱하기 연산은 매우 많은 계산시간이 소요되며, 어떠한 실시간 해석을 하려면 모델의 크기에 영향을 많이 받는다. 더욱이 동적 분석, 비선형분석, 그리고 자르는 작업등 메쉬가 활발하게 변형되는 상황이라면 계산량이 기하급수적으로 증가한다. 따라서 고전적 FE분석은 시간과 정보량 면에서 불리하다. 수술시뮬레이션에서 우리는 면-노드(볼 수 있는 노드)의 변위와 접촉점의 반작용력만이 관심을 둔다. 만약에 수학적으로 전처리절차에서 단지 필요한 정보만을 얻는 최적화가 이루어지면 계산시간은 매우 감소할 것이다. Bro-Nielsen and Cotin (1996)은 실시간 변형을 성취하기 위한 그와 같은 전략을 적용했다[4]. 이는 처리시 알려진 $n \times n$ 크기의 행렬을 $V \times V$ ($v =$ visible 노드 * 자유도)크기로 절감된 압축 기법을 적용했다. 이 행렬 크기의 절감은 행렬 처리에 필요한 시간을 줄일 수 있었다. 감소된 강성행렬은 가시 노드 변위의 계산시 역행렬도 줄이며, 실시간계산은 감소된 강성행렬의 역행렬에 작용력을 곱하는 연산이 얼마나 빨리 처리되느냐에 달려있다. 이 감소된 포맷은 고전적 FE해법에서 얻어진 결과와 똑같은 결과를 나타낸다.

3. FE 적용절차

가상현실 그래픽에서 모핑 및 변형을 위해서는 기존 그래픽에서 사용하던 솔리드(solid)형태의 3각형(polygon)이나 4각형(quad) 모델 등을 사용했는데, 모핑에서는 두께를 지닌 형태의 재료 모델인 테트라 헤드론(tetrahedron)모델과 헥스(Hex)모델 등을 사용하나 본 실험에서는 테트라 헤드론 모델 기반의 Mooney-Rivlin Material model을 사용하였다[5,6]. 유한요소법이란, 복잡한 부품(model)의 형상을 요소라고 부르는 단순한 형상으로 분할하여 부품전체의 응력과 변위를 계산하는 것으로 부품모델을 요소의 집합으로 표현하는 것을 메싱(meshing) 이라고 한다.

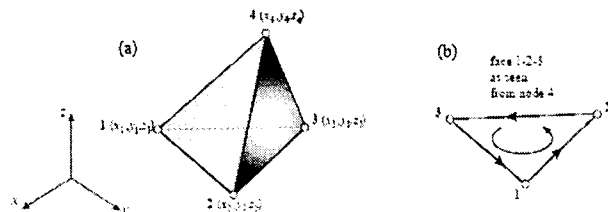


그림 1. 솔리드 모델과 테트라 헤드론 모델

물체의 변형을 실시간으로 효과적으로 디스플레이 하기 위해 유한요소법(FEM)을 활용한 연구가 많이 발표되었다. 3D Object 모델은 Tri(3-node) Face로 구성되고, 그래픽 디바이스에서 렌더링 된다. FEM을 적용하기 위해서는 1. 모델을 이산화 하고(이 과정에서 3D 모델 데이터를 요소형태로 쪼개는 전처리 과정 필요) 2. 각 요소의 변위를 구하고, 3. 각 요소의 Potential Energy를 구하고, 4. 각 요소의 강성행렬을 구하고, 5. 전체 강성행렬 Assembly를 구하고, 6. Inverse 행렬을 구하는 과정을 수행한다.[7]

3.1 Fem 적용 절차

1. Discretize model into tetrahedral finite elements
2. Find the displacement for each element
3. Find the potential energy for each element
- 4./5./6. Build a system describing the displacement of all the elements and solve the system



- (a) The linear tetrahedron element also called the 4-node tetrahedron
 (b) Node numbering convention

그림 2. 선형 테트라 헤드론 엘리먼트

3.2 역행렬 계산

- **LU Decomposition:** K 는 $n \times n$ 강성(stiffness)행렬이고, u and f 는 $n \times 1$ 벡터(Vector)이다. 모든 값이 0이 아닌 100% dense matrix라면, 계산시간은 $\frac{1}{3}n^3$ 이다.
- **Cholesky Decomposition (SPD):** K 행렬의 의 특성이 Symmetric Positive Determination이므로, Cholesky 법을 적용하면 계산시간은 $\frac{1}{6}n^3$ 이다.
- **Sparse Linear Solver:** K 행렬은 요소의 형태(element type), 메쉬 구조, 노드의 수에 의존적이지만 대부분 많은 메모리 공간을 필요로 한다. 1,000개 노드의 모델은 전체 강성 행렬의 약 1%정도만 0이 아닌 값을 가지고 있다(sparcity = 0.01). sparse LU 해석 알고리즘은 100개의 노드인 경우 20%의 전체 계산 시간 감소와 1,000개의 노드인 경우 효율이 80%로 증가한다.
- **Sparse and Cholesky:**본 논문에서 제안하는 방법으로 가장 처리속도를 높일수 있으며 Sparse 방법보다 계산시간이 1/3정도 줄어든다. 그 외 Intel Math Library가 있는데 본 논문에서 비교는 제외하였다.

4. 실험 및 고찰

표 1. 시스템 테스트 환경

system	contents
CPU	intel XEON 2.8Ghz, 2processor
RAM	2G
HDD	160G
Haptic Device	Omni phantom
Langauge	VisualC++ v6.0
LIB	Open Haptics
OPenGL	GLUT 3.7.3

프로그램은 표 1의 환경에서 테스트하였다.

4.1 역행렬 계산방법에 따른 수행시간

실험에서는 1~7번까지의 영역으로 프로그램을 수행한다.

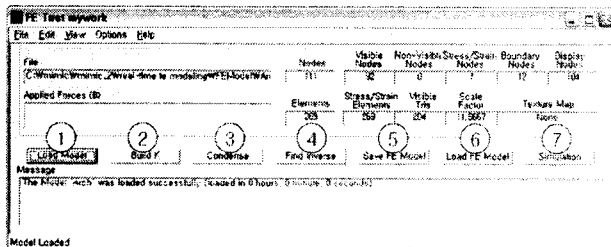


그림 3. 시뮬레이션 프로그램

■ Simulation Process

- 1 Tetrahedron model loading
- 2 Global K matrix make
- 3 Condensation

4 Calculation of inverse matrix

5 save the FE model

6 load the FE model

7 Haptic interface simulation using omni Phantom

본 프로그램 수행결과 4번 역행렬 수행시 3장에서 제안한 방법에 따른 연산 시간을 다음 그림에 표시하였다.

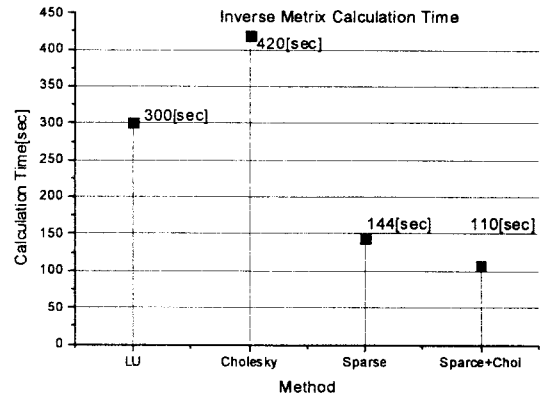


그림 4. 다양한 역행렬 계산시간 비교

시뮬레이션 결과 다양한 역행렬 계산법을 시도하였는데 그 중 빠른 계산결과를 보인 4가지는 LU Decomposition, Cholesky Decomposition (SPD), Sparse Linear Solver 그리고 Sparse와 Cholesky 병행법으로 이들중 병행 계산방법이 가장 빠른 결과를 나타내었다. 이 계산시간은 그림 5의 Arch.mdl 모델을 시뮬레이션하여 시간을 측정하였다.

4.2 모델별 노드수에따른 결과

1) Arch.mdl(111 node)

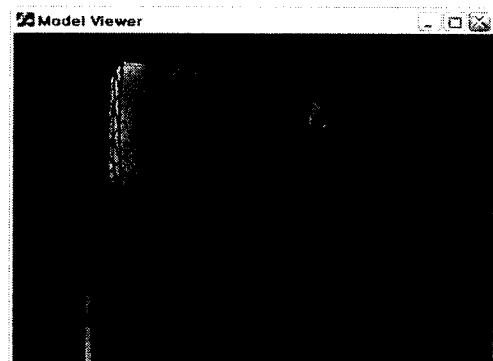


그림 5 arch.m이 시뮬레이션

- Sparcity : 92.155218%
- $K(333 \times 333 \text{ node})$ Matrix Assembled in 1 sec
- Condensation : $333 \times 333 \rightarrow 297 \times 297$
- Sparcity : 90.179007%
- inverse time: 1sec

2) bracket.mdl(976 node)

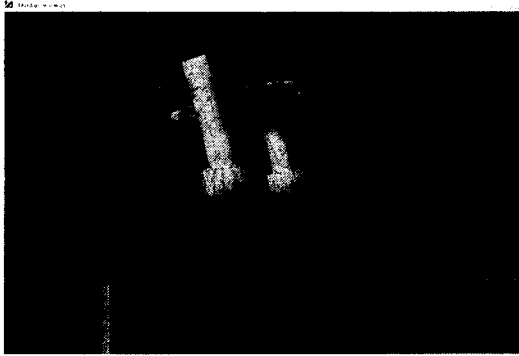


그림 6. bracket.mdl 시뮬레이션

- Sparsity : 98.90684%
- K(2928×2928 node) Matrix Assembled in 1sec
- Condensation : 2928×2928 → 2649×2649
- Sparsity : 98.242105%
- inverse time: 1min 45sec

3) cajon3d.mdl(1020 node)

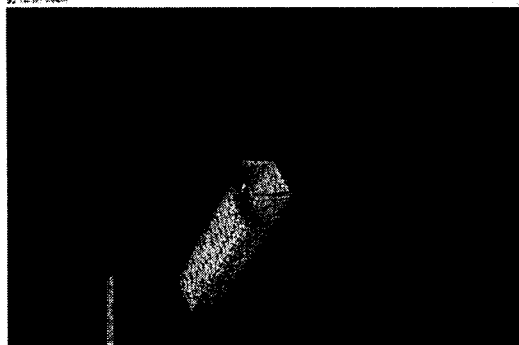


그림 7.cajon3d.mdl 시뮬레이션

- Sparsity : 98.90684%
- K(2928×2928 node) Matrix Assembled in 1sec
- Condensation : 2928×2928 → 2649×2649
- Sparsity : 98.242105%
- inverse time: 2min 17sec

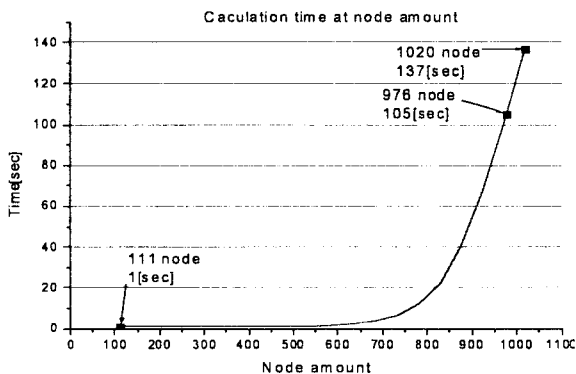


그림 8. 노드수에 따른 계산결과

그림 8의 결과에서 보듯이 노드수가 800개를 넘어서면서부터 계산시간이 지수적으로 상승하며 노드수가

100개와 100개의 차이는 100배 이상의 연산시간이 더 소요됨을 알 수 있다. 따라서 FE모델링에서는 시스템의 능력을 고려하여 적절한 노드수를 정하여 시뮬레이션 해야 실시간 처리를 보장할 수 있을 것이나, 한번 역행렬이 계산되면 힘 방향 실험에서는 간단한 곱셈 연산만으로 변형이 계산됨으로 시뮬레이션에서는 제약을 느낄 수 없었다. 다만 한 개의 행렬 크기가 커지므로 이 행렬을 보관하는 RAM의 크기를 고려해야 한다.

5. 결 론

수술 시뮬레이션 및 건축분야에서는 유한요소법(FEM)을 이용하여 정확한 결과를 예측하곤 하였으나 가상현실에서는 그 계산시간 때문에 이를 적용하기 힘들었다. 다만 노드의 수를 줄이거나 컴퓨터의 성능을 개선하여 수술시뮬레이션 하는 사례들이 있었다.

가상현실에서 유한요소법 기반에 실시간 그래픽 및 햅틱처리를 위해서 연구된 방법 중 워싱턴대 제프리 논문에서 그 해법이 제시되었다. 그러나 이 방법도 몇 가지 제한을 주었는데 가장 큰 문제는 역행렬 계산시간으로, 사전에 계산이 필요한 1,000노드(node) 크기의 역행렬을 구하는데 5분 이상의 시간이 걸렸다.

본 논문에서 제안하는 방식은 사전에 프리프로세싱 단계에서 역행렬을 미리 계산하여 놓고 이를 간단한 곱셈만으로 힘의 크기에 따른 변형을 실시간으로 계산하는 방법을 제안하고, 기존에 특히 문제시 되었던 역행렬 계산시간을 줄이는 방법을 연구하였다.

본 연구에서는 이 역행렬 계산의 고속화를 구현하기 위해 연구하였고, 연구결과 계산시간을 1/3로 단축하는 성과를 이루었다. 따라서 이 방법을 유한요소법 계산에 응용한다면 실시간 가상환경 구축에 도움이 될 것이라 확신한다.

[참 고 문 헌]

- [1] 강원찬, 김남오, 최창주, "실시간 그래픽 디포메이션 알고리즘을 이용한 가상환경 제어" 대한전기학회 제53D권, 제5호, pp. 309-314, 2004. 5.
- [2] Berkley, J., "Determining Soft Tissue Material Properties for the Purpose of Finite Element Modeling of the Below Knee Amputee Residual Limb." Masters Thesis, Northwestern University, Chicago, 1997.
- [3] 강원찬, 신석두, 김남오, 민완기, 김영동, "유비쿼터스 환경을 위한 소방시스템", 대한전기학회, 제54P권 제 3호, pp. 141-147, 2005. 9. 1,
- [4] Bro-Nielsen, M., and Cotin, S., "Real-time volumetric deformable models for surgery simulation using finite elements and condensation." Computer Graphics Forum Vol. 15, No. 3, pp. 57-66. 1996.
- [5] P.G. Ciarlet, "Three-dimensional elasticity", North-Holland, Amsterdam, ISBN 0-444-70259-8, Mathematical elasticity Vol. 1, 1987
- [6] Mooney, M., "A theory of large elastic deformation.", Journal of Applied Physics. vol. 11, pp. 582-592, 1940.
- [7] Gid.exe(3d 툴)(<http://www.gidhome.com/>), gidmsh2tet.exe