

의료영상 분석을 위한 CUDA 기반의 고속 DRR 생성 기법

양상욱*, 최영**, 구승범***

CUDA-based Fast DRR Generation for Analysis of Medical Images

Sangwook Yang*, Young Choi** and Seungbum Koo***

ABSTRACT

A pose estimation process from medical images is calculating locations and orientations of objects obtained from Computed Tomography (CT) volume data utilizing X-ray images from two directions. In this process, digitally reconstructed radiograph (DRR) images of spatially transformed objects are generated and compared to X-ray images repeatedly until reasonable transformation matrices of the objects are found. The DRR generation and image comparison take majority of the total time for this pose estimation. In this paper, a fast DRR generation technique based on GPU parallel computing is introduced. A volume ray-casting algorithm is explained with brief vector operations and a parallelization technique of the algorithm using Compute Unified Device Architecture (CUDA) is discussed. This paper also presents the implementation results and time measurements comparing to those from pure-CPU implementation and open source toolkit.

Key words : CUDA, DRR generation, Pose estimation, Ray-casting

1. 서 론

컴퓨터 비전과 이미지를 이용한 인체 동작의 분석은 지난 2-30동안 폭넓게 이루어져 오고 있다^[1]. 정형외과적 위치 추적에 있어서 초기에는 관찰 대상인 뼈에 작은 구슬 마커를 부착하고 두 방향의 엑스선 이미지를 이용하여 3차원 위치를 추적하는 RSA(Roentgen Stereo-photogrammetry Analysis) 방법이 연구되었으나^[2,3], 최근에는 이러한 마커를 사용하지 않고 엑스레이 이미지와 CT 볼륨 데이터를 이용하여 영상기반의 2D-3D 정합(registration)을 계산하여 뼈의 위치를 추적하는 연구들이 수행되고 있다^[4]. 이미지 기반의 정합은 관찰 부위 외 다른 부위와의 간섭 등에 의해 이미지 비교에 어려움이 있을 수 있고 이로 인해 정합 결과가 수렴하지 않는 경우도 있기 때문에 수렴 속도와 정확성을 향상시키기 위한 연구들 또한 많이 이루어지고 있다.

90년대에는 특징(feature)기반의 3D-2D 정합 연구가 다수 수행되었으며^[5-9], 이후 병암(intensity) 기반의 연구^[10-12] 및 하이브리드 방법과^[13,14] 함께 그래디언트 기반의 연구가^[15,16] 수행되었다. 최근에는 계산속도의 향상을 위하여 병렬 컴퓨팅이나 GPU(Graphic Processing Unit)를 이용한 연구도 수행되고 있다^[17-21]. 특히 최근에는 GPU의 병렬 계산 능력을 그래픽뿐 아니라 일반적인 계산에 사용할 수 있도록 하는 GPGPU(General-Purpose GPU)의 출현으로 이를 활용한 병렬 컴퓨팅에 관한 연구도 활발해지고 있으며^[22], 대표적인 GPGPU 환경인 NVIDIA사의 CUDA^[23]를 이용한 연구도 증가추세에 있다^[17,18,20,21,28].

CT 볼륨 데이터와 엑스레이 이미지의 2D-3D 정합을 통한 자세 추적 과정에서는 볼륨 데이터로부터 DRR을 생성하고 이를 엑스레이 이미지와 비교하는 과정을 반복하게 되는데, DRR 생성과 이미지 비교에 소요되는 시간이 정합 소요 시간의 대부분을 차지한다. 본 논문에서는 일반적으로 널리 사용되는 복셀 기반 볼륨데이터의 DRR을 생성하는 레이캐스팅(ray-casting)을 구현하는데 있어서 GPU지위를 통한 빠른 텍스처(texture) 테이블 검색이 가능하도록 하는 간결

*정회원, (주)코체인솔루션스 선행연구원

**교신저자, 종신회원, 중앙대학교 기계공학부

***비회원, 중앙대학교 기계공학부

- 논문부고일: 2011. 03. 11

- 논문수정일: 2011. 05. 27

- 심사완료일: 2011. 05. 28

한 벡터 연산 방법과 CUDA를 이용한 병렬화 방법을 소개한다. 또한, 구현 결과의 수행속도를 널리 사용되는 오픈소스 레이캐스팅 도구의 수행속도와 비교한다.

2. 관련연구 및 기술

2.1 Stereoradiometric Analysis

의학 영상의 2D-3D 정합에 의한 자세 분석은 대상자의 CT데이터를 먼저 획득하고 대상자가 움직일 때 획득되는 관심 대상 부위의 엑스레이 동영상에 정합 시킴으로써 3D 볼륨 데이터의 6자유도 위치정보($T_x, T_y, T_z, R_x, R_y, R_z$)를 각 프레임에 대해서 계산하여 대상 뼈의 위치를 추적한다^[5,21]. Stereoradiometric 자세 분석에서는 Fig. 1에 나타난 바와 같이 양방향(bi-plane) 엑스레이 동영상에 프레임별로 위치정보를 일치시킴으로써 정합의 수렴 속도와 정확도에서 이점을 얻고자 하며, 궁극적으로는 동영상 프레임에 실시간으로 정합을 계산하는 것이 목표이다. 본 논문에서는 정합 과정 중 가장 많은 시간이 소요되는 DRR의 고속 생성에 중점을 두어 논의한다.

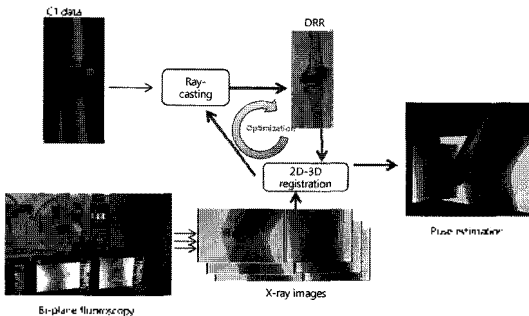


Fig. 1. Stereoradiometric analysis.

2.2 CUDA

CUDA는 수십에서 수백 개의 병렬 코어를 가지는 NVIDIA 사의 GPU를 이용하여 수천 개의 스레드에 의한 병렬 계산을 가능하도록 하는 GPGPU 아키텍처이다^[29]. Fig. 2(a)와 같이 순차적으로 증가 혹은 감소하는 인덱스와 관련된 연산을 수행하는 프로그램 코드는 Fig. 2(b)와 같이 동시에 수행되도록 병렬화 될 수 있으며, 이 때 병렬화된 코드는 CPU가 아닌 GPU 상에서 수행된다. Fig. 2에서는 병렬 스레드가 N개의 일차원으로 예시되어 있지만, 실제CUDA에서는 다차원 그리드(grid) 단위로 병렬 수행이 이루어진다. 그리드는 1~2차원 블록(thread block)으로 구성되고 각 블록은 최대 3차원까지의 스레드들로 구성될 수 있다.

며, 동일 블록의 스레드들은 동시 수행이 보장된다. 몇 개의 블록이 실제로 동시 수행될 지는 장치 하드웨어에 따라 달라질 수 있다.

CUDA에서는 CPU 부분을 호스트(host), GPU 부분을 장치(device)라고 부른다. 특히, C 언어를 확장한 CUDA C에서는 장치에서 병렬화되어 동시에 호출될 수 있는 함수를 커널(kernel)이라고 하는데, 커널에서 사용될 코드와 메모리는 장치상에서 실행 및 접근 가능하여야 한다.

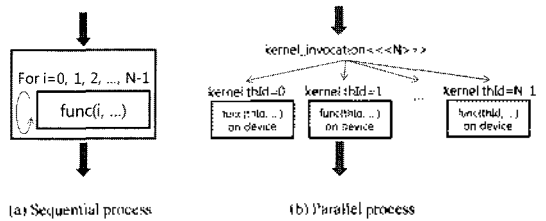


Fig. 2. 순차적 프로세스와 병렬 프로세스.

호스트의 입력데이터 I_h 를 처리하여 출력데이터 O_h 를 생성하는 CUDA 프로그램은 다음과 같은 절차를 따라 구현된다^[29,30].

- a. 장치 메모리 I_d 와 O_d 를 할당.
- b. I_h 의 데이터 블록을 I_d 로 복사.
- c. 커널을 호출하여 I_d 로부터 O_d 데이터를 구성.
- d. O_d 를 O_h 에 복사하여 완료.

병렬 프로세스가 효율적으로 수행되기 위해서는 몇 가지 고려사항이 있다. 위의 절차의 b와 d 과정에서 대량의 메모리 복사가 일어날 수 있는데, 이들의 빈도를 낮출 수 있도록 구현되어야 하고 수행되어야 하는 GPU 장치의 모든 물리적 코어가 동시에 작동되도록 병렬 스레드의 구성을 적절히 이루어야 한다. 또한 입력 데이터의 빠른 검색과 보간을 위한 텍스처(texture) 메모리의 사용이나 다수 스레드의 중복 접근을 줄이기 위한 커널간 공유 메모리의 사용도 고려할 수 있다.

병렬 계산 과정에서 다수의 스레드에서 특정 메모리 영역에 동시에 접근하여 변경을 시도한다면, 한 스레드가 데이터를 변경하는 동안 다른 스레드들은 대기해야 하는 경우가 생겨 결과적으로 순차적 수행보다 느린 결과를 가져올 수도 있다. 따라서 동일 메모리 영역을 여러 스레드에서 동시에 사용하지 않도록 스레드 구성과 알고리즘을 설계하는 것이 중요하며, 퍼치 못할 경우 아톰릭 연산(atomic operation)^[30]을 사용하여 계산의 완결성을 보장하여야 한다. 변수나

메모리 공간에 새로운 값을 더하는 누적 연산은 조화, 덧셈 및 지장의 세부연산으로 이루어지는데 병렬 쓰레드에서 동일 변수에 대한 누적 연산이 경쟁적으로 발생한다면 세부연산이 섞임으로써 연산의 완결성 보장이 어렵지만 아토믹 연산은 특정 연산을 분리되지 않는 연산으로 수행한다. 응용 분야별 프로그래밍 모델과 고려사항은 참고문헌^[11]에 잘 정리되어 있다.

3. 고속 DRR 생성

3.1 Calibration과 x-ray image plane

DRR 생성은 3D 데이터인 CT 데이터로부터 엑스레이 영상과 유사한 2D 이미지를 생성하는 기술이다. 따라서 DRR 생성 과정에서는 3차원 좌표계에서 엑스레이 소스의 위치와 감광판의 위치 및 상대적인 위치를 알 수 있는 정보가 필요한데 이는 보정(calibration)을 통해 얻어진다. 대상 인체를 조영하기 전에 일정한 격자로 이루어진 보정틀(calibration cage)을 조영하여 얻어진 이미지로부터 3D 공간상의 기준 좌표계를 정하고 그 좌표계에서의 카메라(엑스레이 소스) 정보를 계산한다. 그 결과로 카메라 위치 C , 카메라의 초점 중심 위치 F , 카메라의 윗방향 벡터 U , 카메라의 조영각 θ 를 얻을 수 있다. 보정 과정은 참고문헌^[12]에 잘 나타나 있으며 본 논문에서는 카메라의 정보가 얻어진 이후의 계산에 대해 논의한다.

본 논문에서 제안하는 DRR 생성 방법들은 3D 기준 좌표계에서 레이캐스팅을 과정을 기술한다. 따라서 레이캐스팅의 결과로 생성된 이미지 또한 3D 좌표계에서 정의되는데, 엑스레이 이미지를 3D 상의 조영 스크린으로 보내는 변환 M 은 다음과 같이 계산된다.

$$M = TRT_cT_f$$

여기서, T_f 는 엑스레이 이미지 좌표계에서 중심을 원점으로 보내는 이동변환이고 R 은 $F-C$ 를 z 축, U 를 y 축이 되게 하는 회전변환이며, T_c 는 이미지 중심을 카

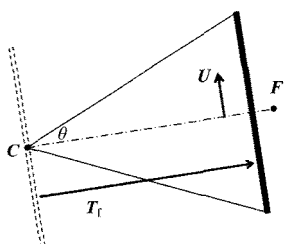


Fig. 3. Image to screen transform.

메라 위치 C 로 보내는 이동변환, T_c 는 회전된 이미지 중심이 FC 의 연장선 상에 있도록 하면서 조영각 θ 에 이미지의 크기가 일치하도록 하는 이동변환이다. Fig. 3에서 점선은 변환 TRT_c 에 의한 이미지 위치를 나타내며 검은 박스는 변환 M 에 의해 기준 좌표계 공간에 사상된 조영 스크린의 위치를 나타낸다.

보정과정에서 기준이 되는 엑스레이 이미지의 해상도가 (r_x, r_y) 이고, 픽셀 간격(spacing)이 (s_x, s_y) 라면, 조영 스크린은 다음과 같이 정의될 수 있다.

$$S = sR_0 + tR_1 + O, (0 \leq s \leq 1, 0 \leq t \leq 1) \quad (1)$$

where,

$$O = [0, 0, 0] M,$$

$$R_0 = [(r_x - 1) s_x, 0, 0] M - O$$

$$R_1 = [0, (r_y - 1) s_y, 0] M - O$$

3.2 CT 기반 볼륨 레이캐스팅

3.2.1 볼륨 레이캐스팅 알고리즘

볼륨 레이캐스팅에 대해서는 관련된 많은 연구들이 있으며^[13,14], 여러 연구들에서 엑스레이의 물리적 성질을 적용한 명암의 재현 방법도 제시하고 있다. 본 논문에서는 간결한 계산방법에 집중하기 위하여 레이를 추적하는 동안 벡터 연산과 픽셀 계산을 위한 인덱스 연산에 중점을 두어 기술한다.

CT 데이터로부터 얻어진 볼륨데이터는 복셀의 삼차원 배열로 표현되는데, 이 배열의 인덱스를 빠르게 결정하기 위하여 3D 공간상의 기준 좌표계에서 결정된 카메라 위치와 조영 스크린 변환을 볼륨 데이터의 좌표계로 변환하여 계산한다. Fig. 4와 같이 카메라와 조영 스크린이 결정되어 있고, 6자유도 위치 정보($T_x, T_y, T_z, R_x, R_y, R_z$)에 의해 결정되는 볼륨의 변환 T_v 가 주어질 때, 카메라 및 조영 정보에 T_v 의 역변환을 취하여 볼륨의 좌표계를 기준으로 레이를 추적하면, 레이가 지나는 복셀의 인덱스를 쉽게 구할 수 있다.

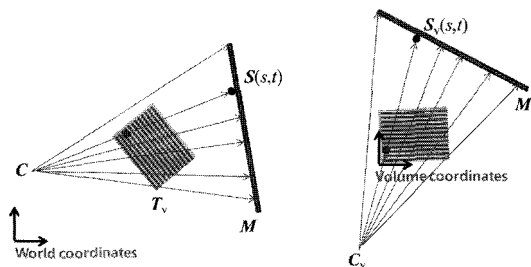


Fig. 4. 볼륨 좌표계 기준의 조영 정보.

볼륨 변환 행렬 및 복셀 배열로부터 보정 정보에 의하여 DRR을 생성하는 알고리즘은 다음과 같다. 보정 정보는 볼륨 좌표계 기반으로 변환되어 사용된다.

Algorithm VolumeRaycasting

Input:

Voxel array $v[w:h:d]$ with spacing (s_0, s_1, s_2)

Volume transform T_v

Calibration information;

$$C_v = CT_v^{-1},$$

$$M_v = MT_v^{-1},$$

$$S_v = sR_0T_v^{-1} + tR_1T_v^{-1} + OT_v^{-1}$$

Output:

DRR image $d[l:m]$

For $0 \leq i \leq l$ and $0 \leq j \leq m$,

$$s = iw, t = jh$$

$$r_{in}, r_{out} = \text{intersection_points}(C_v S_v(s, t), v)$$

$$n = \text{calculate_step}(r_{in}, r_{out}, (s_0, s_1, s_2))$$

$$sum = 0$$

For $0 \leq k \leq n$,

$$u = kn$$

$$p = (1-u)r_{in} + u r_{out}$$

$$idx = p / [s_0, s_1, s_2]$$

$$sum = sum + v[idx]$$

End for k

$$d[i, j] = sum$$

End for i, j

알고리즘에서 *intersection_points*는 선분과 마운딩 박스의 두 교차점을 구하는 것이고 *calculate_step*는 볼륨 경계의 두 점을 잇는 선분이 몇 개의 복셀을 지나는지 계산하는 것이다.

3.2.2 볼륨 레이캐스팅의 병렬화

알고리즘 *VolumeRaycasting*에서 출력 이미지의 해상도를 따라 순차적으로 수행되는 $l \times m$ 루프를 병렬화할 수 있으며 이는 각각의 레이가 병렬화되도록 하는 의미가 된다. CUDA 지원 장치는 대체로 블록당 255개의 쓰레드를 지원하므로 하나의 블록은 16×16 개의 쓰레드를 가지도록 하면 그리드의 크기는 다음과 2차원으로 정의할 수 있다.

$$\text{grid.x} = (l + 15) \bmod 16$$

$$\text{grid.y} = (m + 15) \bmod 16$$

이렇게 정의된 그리드와 블록에 대해서 커널을 수

행하게 되면 CUDA C의 빌트인 변수를 이용하여 각각의 커널이 처리해야 할 화소의 i, j 인덱스를 계산할 수 있으며 각각의 레이가 생성해야 할 DRR 화소는 다른 레이에 대해 독립적이므로 쓰레드간 충돌없이 쉽게 병렬화 구현이 가능하다. 커널에서 수행되는 계산들은 장치 상에서 수행되기 때문에 프로그램의 구현에 있어서 C 표준함수가 아닌 CUDA C가 지원하는 수학 연산 함수와 벡터 데이터 타입 기반으로 직접 구현되어야 하는데, 소개된 알고리즘의 벡터 연산들은 모두 쉽게 구현될 수 있다.

4. 구현 및 수행 속도 분석

4.1 구현 및 실행 환경

본 연구의 구현에서는 DICOM 형식의 CT 데이터와 엑스레이 이미지를 읽어들이어서 CPU를 이용한 연산과 CUDA를 이용한 연산 결과를 비교하였다. CT 데이터를 읽고 볼륨으로부터 등치면을 추출하기 위해서 오픈소스인 VTK(Visualization toolkit) 5.6.0을^[32] 사용하고 CUDA 구현을 위해서 CUDA Toolkit 3.1을 사용하였다. 프로그램 소스는 C++로 구현되어서 Visual Studio 9.0 SP1 환경에서 x64 코드로 컴파일 되었다.

프로그램의 실행과 속도 측정은 6 GB 메모리와 Intel Core i7 CPU(3.07 GHz)를 가진 데스크톱 PC 상의 Microsoft Windows 7 Enterprise K 64bit 운영체제에서 이루어졌으며, CUDA 장치로 NVIDIA Quadro FX4800(1.5 GB)을 사용하였다.

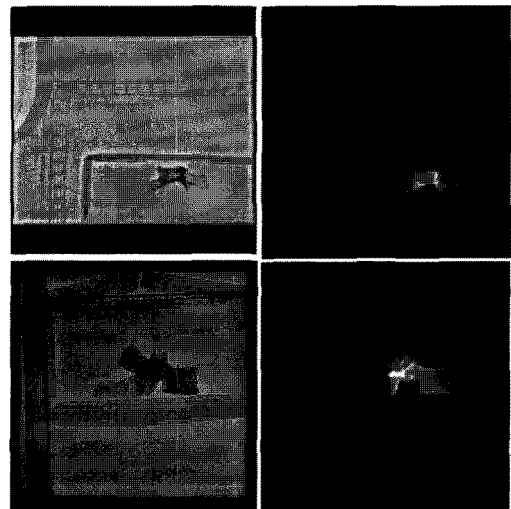


Fig. 5. 엑스레이와 DRR 이미지 비교.

4.2 수행 결과

Fig. 5는 본 연구를 통해 구현된 DRR 생성의 결과의 예를 보여주고 있다. 위, 아래 행은 각각 AP (Anterior-Posterior)와 ML(Mediolateral) 방향을 보여 주며, 각 행의 왼쪽 열은 기준 엑스레이 이미지, 오른쪽 열은 구현된 레이캐스팅 결과이다. 그림에서는 DRR 이미지가 엑스레이와 정합된 위치에 있는 것처럼 보이는데, 이는 볼륨의 변환이 잘 적용되는 지를 확인하기 위해 수동으로 정합한 결과의 변환 행렬을 사용한 것이다.

Fig. 5의 오른쪽에 나타난 이미지들은 512×512×81의 크기를 갖는 CT 데이터로부터 볼륨 레이캐스팅을 수행하여 512×512 크기의 DRR 이미지를 생성한 결과이다.

이들 이미지를 얻기 위하여 제안된 볼륨 레이캐스팅 방법을 각각 CPU 기반과 CUDA 기반으로 구현하고 AP 방향과 ML 방향의 카메라에 맞춘 DRR 생성 수행 시간을 측정하였다. 또한, VTK가 제공하는 볼륨 렌더링 기능들을 같은 방법으로 수행하여 시간을 측정하였으며 이들 결과는 Table 1에 나타나 있다.

Table 1. DRR 생성 시간(millisecond)

Method	AP 방향	ML 방향
CPU	315.6	936.3
CUDA	8.96 (14.3)	8.15 (22.1)
vtkVolumeRayCastMapper	69.6	110.1
vtkVolumeTextureMapper3D	22.8	38.3

Table 1에서 CPU로 표시된 항목은 본 논문의 순차적 구현방법으로 수행한 것이며, CUDA로 표시된 항목은 병렬 구현하여 CUDA 상에서 수행한 것이다. 아래의 두 항목은 VTK에서 제공하는 레이캐스팅 기능을 사용한 것인데, vtkVolumeRayCastMapper는 CPU에서 복셀의 배열을 이용하여 레이캐스팅을 구현한 클래스이며 vtkVolumeTextureMapper3D는 GPU의 3D 텍스처 가속 기능을 사용한 레이캐스팅을 구현한 클래스이다. 이 두 클래스는 공히 CPU의 나중 쓰레드를 사용하여 구현되므로 순차적 CPU 구현보다는 빠른 속도를 보인다. CUDA를 이용한 결과의 필호안 숫자는 입력데이터의 복사 및 준비 시간을 포함하여 레이캐스팅을 수행하는데 걸리는 시간을 나타내며 괄호밖에 표시된 시간은 변환행렬 적용과 DRR 생성을 반복하는 경우 레이캐스팅 수행시간이다. 볼륨 변환 행렬을 변경하면서 DRR 생성을 반복할 경

우 괄호 밖의 시간이 의미있는 소요 시간이 된다. Table 1에는 나타나 있지 않지만 VTK 클래스 중 vtkVolumeTextureMapper3D의 경우에도 최소 한번의 렌더링 시간은 반복 렌더링의 경우보다 오래 걸리는 데, 내부적으로 볼륨 데이터를 GPU의 텍스처 메모리로 복사하는 처리를 수행하는 것으로 추측할 수 있다.

본문에서 소개된 알고리즘 VolumeRaycasting에도 나타나 있지만, 수행시간은 출력 DRR 이미지의 해상도와 볼륨의 크기에 비례해서 늘어난다. 출력 이미지가 커질수록 추적하는 레이의 수가 늘어나고, 볼륨 데이터가 커질수록 레이 추적 과정의 복셀 조회 회수가 늘어난다. 이로 인해 같은 크기의 데이터라도 카메라의 방향에 따라 수행 시간이 달라진다.

5. 결 론

본 논문에서는 엑스레이 조영 환경의 보정 정보와 볼륨의 변환 행렬을 적용한 레이캐스팅 방법을 간결하게 기술하는 벡터 연산을 정리하였다. 또한 DRR 생성 시간을 단축할 수 있도록 GPU 병렬 컴퓨팅을 사용한 고속 DRR 생성 알고리즘을 소개하고, 병렬화 과정에서 고려되어야 하는 점들을 정리하였다. 소개된 병렬화 레이캐스팅 방법의 구현 결과는 CPU만으로 수행된 결과와 함께 VTK에서 제공하는 레이캐스팅 기능의 결과들과 비교하였다.

구현된 고속 DRR 생성 방법은 CT와 엑스레이를 이용한 인체 자세추적을 계산하는 과정에서 활용되어 자세추적 계산의 속도 향상에 기여할 수 있을 것이다. 향후 자세추적 과정의 비교 매트릭 계산 또한 병렬 알고리즘으로 고속화 하고, DRR 생성 또한 비교 매트릭에 최적화하는 연구의 진행을 통하여 실시간 정합을 수행할 수 있을 것으로 기대한다.

감사의 글

이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업 임(No. 2010-0001641, 2010-0015897, 2010-0006582).

참고문헌

1. Moeslund, T. B., Hilton, A. and Krüger, V. A., "Survey of Advances in Vision-based Human Motion Capture and Analysis." *Computer Vision and Image Understanding*, Vol. 104, pp. 90-126, 2006.
2. Selvik, G., "Roentgen Stereophotogrammetry: A

- Method for the Study of the Kinematics of Skeletal System," *Acta Orthop Scand Suppl*, Vol. 24, pp. 343-352, 1983.
3. Valstar, E. R., Nelissen, R., Reiber, J. and Rozing, P. M., "The Use of Roentgen Stereophotogrammetry to Study Micromotion of Orthopaedic Implants," *ISPRS Journal of Photogrammetry & Remote Sensing*, Vol. 56, pp. 376-389, 2002.
 4. van de Kraats, E. B., Penney, G. P., Tomažević, D., van Walsum, D. and Niessen, W. J., "Standardized Evaluation Methodology for 2-D-3-D Registration," *IEEE Trans. Medical Imaging*, Vol. 24, pp. 1177-1188, 2005.
 5. Murphy, M. J., "An Automatic Six-degree-of-freedom Image Registration Algorithm for Image-guided Frameless Stereotaxic Radiosurgery," *Med Phys*, Vol. 24, No. 6, pp. 857-866, 1997.
 6. Hamadeh, A., Lavalée, S. and Cinquin, P., "Automated 3-Dimensional Computed Tomographic and Fluoroscopic Image Registration," *Computer-Aided Surgery*, Vol. 3, No. 1, pp. 11-19, 1998.
 7. Guźciec, A., Kazanzides, P., Williamson, B. and Taylor, R. H., "Anatomy-based Registration of CT-scan and Intraoperative X-ray Images for Guiding a Surgical Robot," *IEEE Trans Med Imag*, Vol. 17, pp. 715-728, 1998.
 8. Bansal, R., Staib, L., Chen, Z., Rangarajan, A., Knisely, J., Nath, R. and Duncan, J. S., "A Minimax Entropy Registration Framework for Patient Setup Verification in Radiotherapy," *Computer-Aided Surgery*, Vol. 4, pp. 287-304, 1999.
 9. Czopf, Á., Brack, C., Roth, M. and Schweikard, A., "2D-3D Registration of Curved Objects," *Periodica Polytechnica Ser Elect Eng*, Vol. 43, No. 1, pp. 19-41, 1999.
 10. Penney, G. P., Batchelor, P. G., Hill, D. L. G., Hawkes, D. J. and Weese, J., "Validation of a Two- to Three-dimensional Registration Algorithm for Aligning Pre-operative CT Images and Intraoperative Fluoroscopy Images," *Med Phys*, Vol. 28, No. 6, pp. 1024-1032, 2001.
 11. Rohlfing, T., Russakoff, D. B., Murphy, M. J. and Maurer Jr, C. R., "An Intensity-based Registration Algorithm for Probabilistic Images and Its Application for 2-D to 3-D Image Registration," *Proc. SPIE 2002*, pp. 581-591, 2002.
 12. Hipwell, J. H., Penney, G. P., McLaughlin, R. A., Rhode, K., Summers, P., Cox, T. C., Byrne, J. V., Noble, J. A. and Hawkes, D. J., "Intensity-based 2-D-3-D Registration of Cerebral Angiograms," *IEEE Trans Med Imag*, Vol. 22, No. 11, pp. 1417-1426, 2003.
 13. Russakoff, D. B., Rohlfing, T., Ho, A., Kim, D. H., Shahidi, R., Adler Jr, J. R. and Maurer Jr, C. R., "Evaluation of Intensity-based 2D-3D Spine Image Registration using Clinical Gold-standard Data," *Lec Note Comput Sci WBIR*, Vol. 2717, pp. 151-160, 2003.
 14. Vermandel, M., Betrouni, N., Palos, G., Gauvrit, J. Y., Vasseur, C. and Rousseau, J., "Registration, Matching, and Data Fusion in 2D/3D Medical Imaging: Application to DSA and MRA," *Lec Note Comput Sci MICCAI*, Vol. 2878, pp. 778-785, 2003.
 15. Livyatan, H., Yaniv, Z. and Joskowicz, L., "Gradient-based 2-D/3-D Rigid Registration of Fluoroscopic X-ray to CT," *IEEE Trans Med Imag*, Vol. 22, No. 11, pp. 1395-1406, 2003.
 16. Tomažević, D., Likar, B., Slivnik T. and Pernuš, F., "3-D/2-D Registration of CT and MR to X-ray Images," *IEEE Trans Med Imag*, Vol. 22, No. 11, pp. 1407-1416, 2003.
 17. Shams, R., Sadeghi, P., Kennedy, R. and Hartley, R., "A Survey of Medical Image Registration on Multicore and the GPU," *Signal Processing Magazine*, Vol. 27, No. 2, pp. 50-60, 2010.
 18. Mori, S., Kobayashi, M., Kumagai, M. and Minojima, S., "Development of a GPU-based Multithreaded Software Application to Calculate Digitally Reconstructed Radiographs for Radiotherapy," *Radiological Physics and Technology*, Vol. 2, pp. 40-45, 2009.
 19. Xu, L. and Wan, J. W. L., "Real-Time 2D-3D Medical Image Registration using RapidMind Multi-core Development Platform," *Proc EMBC08*, pp. 20-24, 2008.
 20. Grabner, M., Pock, T., Gross, T. and Kainz, B., "Automatic Differentiation for GPU-accelerated 2D/3D Registration," *Lec Note Comput Sci AAD*, Vol. 64, pp. 259-269, 2008.
 21. Shams, R., Sadeghi, P., Kennedy, R. and Hartley, R., "Parallel Computation of Mutual Information on the CPU with Application to Real-time Registration of 3D Medical Images," *Computer Methods and Programs in Biomedicine*, Vol. 99, No. 2, pp. 133-146, 2010.
 22. Owens, J.D., Luebke, D., Govindaraju, N., Harris, M., Krüger, J., Lefohn, A. E. and Purcell, T., "A Survey of General-Purpose Computation on Graphics Hardware," *Computer Graphics Forum*, Vol. 26, No. 1, pp. 80-113, 2007.
 23. NVIDIA, CUDA Zone, http://www.nvidia.com/object/cuda_home_new.html, 2010.
 24. Frühauf, T., "Raycasting Vector Fields," *In IEEE Visualization '96*, pp. 115-120, 1996.
 25. Levoy, M., "Display of Surfaces from Volume Data," *IEEE Computer Graphics and Applications*, Vol. 8, No. 3, pp. 29-37, 1988.
 26. Danskin, J. and Hanrahan, P., "Fast Algorithms for Volume Ray Tracing," *VVS '92 Proceedings of the workshop on Volume Visualization*, pp. 91-106, 1992.
 27. Parker, S., Parker, M., Livnat, Y., Sloan, P., Hansen,

C. and Shirley, P., "Interactive Ray Tracing for Volume Visualization", *Visualization and Computer Graphics*, Vol. 5, No. 3, pp. 238-250, 1999.

28. Weiskopf, D., *GPU-Based Interactive Visualization Techniques*, Springer, 2006.

29. NVIDIA, *CUDA C Programming Guide*, http://developer.download.nvidia.com/compute/cuda/3_2_prod/toolkit/docs/CUDA_C_Programming_Guide.pdf, 2010.

30. Sanders, J. and Kandrot, E., *CUDA by Example*, Addison-Wesley, 2010.

31. NVIDIA, *CUDA C Best Practices Guide*, http://developer.download.nvidia.com/compute/cuda/3_2_prod/toolkit/docs/CUDA_C_Best_Practices_Guide.pdf, 2010.

32. Kitware, *VTK - The Visualization Toolkit*, <http://www.vtk.org>, 2010.



양 상 옥

1998년 중앙대학교 기계설계학과 학사
 2000년 중앙대학교 기계설계학과 석사
 2000년~2003년 삼성SDS 솔루션개발추진실 선임
 2003년~2004년 이다이엑솔루션즈 차장
 2009년 중앙대학교 기계공학부 박사
 2009년~2010년 중앙대학교 미래신기술연구소 전임연구원
 2011년~현재 코체인솔루션즈 수행연구팀 수석연구원
 관심분야: 형상모델링, 3D 가시화, CAD 데이터교환, 협업 설계, 병렬 컴퓨팅



최 영

1979년 서울대학교 기계설계학과 학사
 1981년 KAIST 생산공학과 석사
 1989년 Carnegie Mellon University 박사
 1990년~1991년 KIST CAD/CAM 연구실 선임연구원
 2001년~2002년 미국 국립표준연구소(NIST) 객원연구원
 1992년~현재 중앙대학교 기계공학부 교수
 관심분야: 형상모델링, Physically based modeling, Haptic 기술 응용, 의료이미지 기반 형상모델링



구 승 범

2000년 서울대학교 농기계과 학사
 2002년 서울대학교 기계항공공학부 석사
 2006년 Stanford University 기계공학과 박사
 2006년~2007년 Stanford University 기계공학과 박사후연구원
 2007년~2009년 Stanford University 영상의학과 선임연구원
 2009년~현재 중앙대학교 기계공학부 조교수
 관심분야: 인체운동역학, 의료영상처리