CT reconstruction using GPU acceleration

2023 Spring - 창의적통합설계 II : 최종발표

OSSTEM 조 : 우동균, 윤덕형, 이민우, 정지후

Goal & Requirement

- **Goal**

① FDK 알고리즘을 활용한 CT reconstruction 구현
② GPU 병렬 연산을 통해 CT reconstruction 속도 개선

- **Requirement**

① FDK 알고리즘을 통해 얻은 output image의 품질 개선
② reconstruction 과정을 치과에서의 실용성을 위해 3분 이내로 단축

- **Input Data**

-projection data 706장 (3.21GB)

Approach

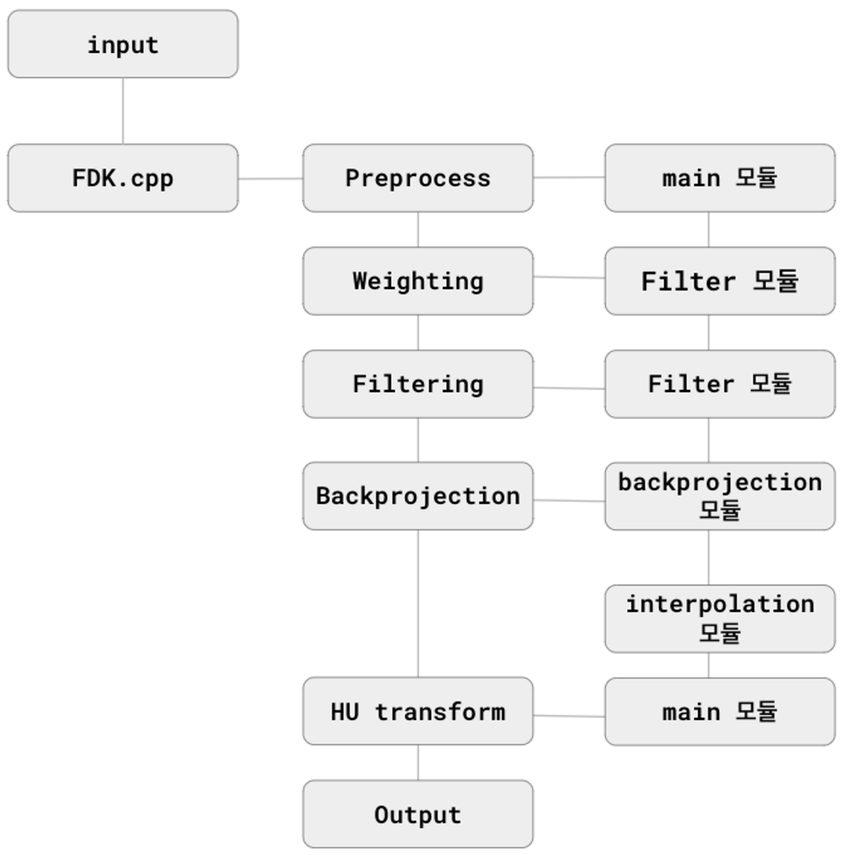
- **개발환경**

① CPU 모델 : matlab, c
② GPU 모델 : matlab, CUDA

- **Approach**

① CPU를 이용한 FDK 알고리즘 구현 및 활용 가능성 확인
② half-fan detector의 특성을 반영하여 weighting 진행
③ 다양한 filter function 적용을 통한 image 품질 개선
④ GPU multithreading을 통해 연산 속도 향상

Architecture- C++



- **Preprocessing - main 모듈**

① input data read & preprocessing

- **Weighting - filtering 모듈**

① cosine weighting
② half-fan detector의 특성을 반영하여 half fan weighting 진행

- **Filtering - filtering 모듈**

① high-pass filter 중 하나인 ramp filter 적용

- **Backprojection - backprection 모듈, interpolation 모듈**

① angle별 backproject 반복 수행.
② CUDA 프로그래밍을 위해 make_rx_ry, make_pu_ratio, make_pv, interp2, image_final 커널을 정의
③ 데이터를 device로 복사하고 device에서 interpolation

Implementation

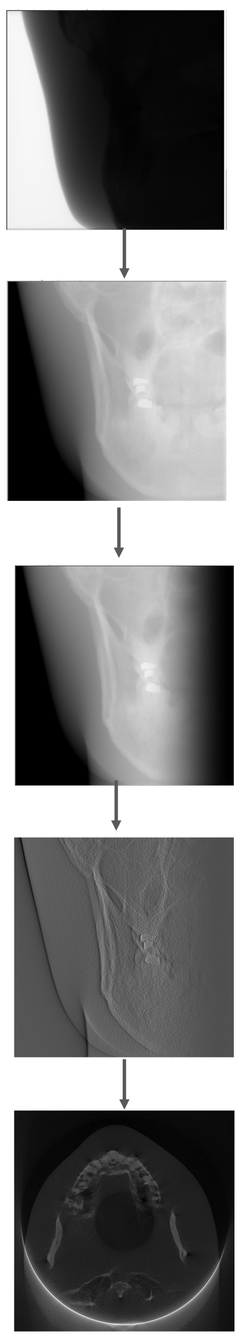
- **FDK 알고리즘**

$$f(x,y,z)=\int_0^{2\pi}\frac{R^2}{U(x,y,\beta)}P(u(x,y,\beta),v(x,y,z,\beta),\beta)d\beta$$

$$u(x,y,\beta)=R\frac{-xsin\beta+ycos\beta}{R+xcos\beta+ysin\beta},$$

$$v(x,y,z,\beta)=z\frac{R}{R+xcos\beta+ysin\beta}$$

$$U(x,y,\beta)=R+xcos\beta+ycos\beta$$



- **Preprocessing**

raw projection data I(r)을 아래 식을 통해 sinogram p(r)로 만듦

$$p_\theta(r)=-\ln\frac{I_\theta(r)}{I_0}=$$

- **Weighting**

기본적인 cosine weighting, half-fan geometry에 적용 위한 half-fan weighting 적용

$$w(u)=\begin{cases}0, & u_{min}\le u<u_{-o}\\ \sin^2\left(\frac{\pi}{2}\frac{u-u_{-o}}{\Delta u}\right), & u_{-o}\le u\le u_{+o}\\ 1, & u_{+o}<u\le u_{max}.\end{cases}$$

- **Filtering**

① 각 값에 Fourier Transform을 적용함
② 각 값에 high pass Filter 인 ramp filter를 적용함
③ 각 값에 Inverse Fourier Transform을 적용함
-> 노이즈가 상당 부분 제거된 값을 구할 수 있음

- **Backprojection & HU scaling**

Point Driven 방법을 사용함
각 slice에서 연산을 통해 (u, v) 좌표의 값들을 (x, y, z) 좌표로 이동함
주어진 HU information으로 HU scaling 진행해 int로 저장

Result

- **matlab 코드**
  - cpu
    - 완성도 : 완성
    - 소요 시간 : 매우 느림 - 706장 이용한 테스트 불가 수준
  - gpu - gpuarray 이용
    - 완성도 : 완성
    - 소요시간 : 매우 느림 - 약 15분 소요
- **C 코드**
  - cpu
    - 완성도 : 완성
    - 소요시간 : 매우 느림 - 약 50분 소요
- **Cuda 코드**
  - gpu
    - 완성도 : 정확한 filtering과 backprojection 구현 실패
    - 소요시간 : 2분 40초