

3. Results & Discussion

Exercise 1

Create projection data with $\Delta\theta = 1^\circ$ from 0 to 180 degree. Display the projection data (i.e., sinogram) as an image. In simulation, the projections from 0 to 180 degrees will do, but not in real situations, why? Discuss

Exercise 1에서는 본인이 정한 image를 0° 에서 180° 까지 1° 마다 회전시켜 projection data를 기반으로 한 sinogram을 얻는 활동이다. Image를 회전 시킬 때는

$$\begin{aligned}x_n &= x_o \cos\theta - y_o \sin\theta \\y_n &= x_o \sin\theta + y_o \cos\theta\end{aligned}\tag{2.1}$$

Eq. (2.1)을 기반으로 각각의 각도에서 얻은 projection data는 sinogram array에 저장한다. Figure 2.1은 크기가 512×512 인 원본 image를 보여준 것이며 직사각형 형태를 가지고 있다. 아래 Figure 2.2는 원본 image를 256×256 으로 변환해주고 rotate function에 의해 최종적으로 180° 회전된 image이다.



Figure 2.1 512×512 sized original image



Figure 2.2 원본 image를 256×256 size로 변환하고 흑백으로 설정한 뒤 180° 회전시킨 image

Figure 2.3은 sinogram을 보여주는 그림인데, 256개의 projection data를 0°에서 180°까지 회전시키며 181번 얻어온 것이기 때문에 크기가 256×181이 된다.

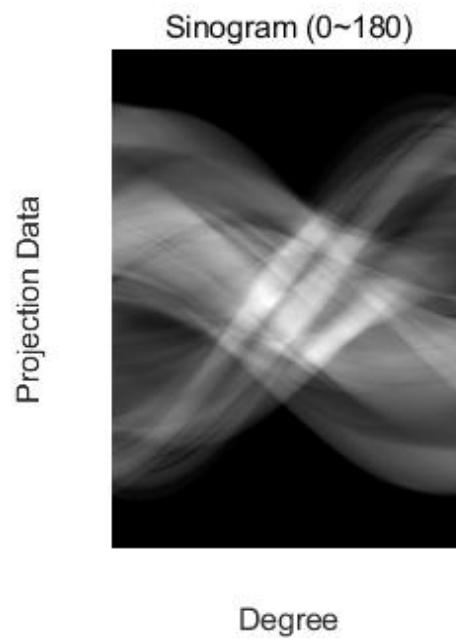


Figure 2.3 Sinogram of original image

Figure 2.3을 보면 왜 회전을 360° 가 아닌 180° 로 하는 것이다. Sinogram 행렬의 행은 각 단위 각도마다 돌았을 때 projection data 181개를 나타내며, 열은 0° 부터 180° 를 나타낸다. 360° 로 회전시키게 되면 중복된 data가 생기기 때문에 360° rotation이 아닌 180° rotation을 사용한다.

위의 Figure 2.3에서 확인할 수 있듯이, 원본 image의 sinogram에서 좌우는 정상적으로 회전이 되는 것을 관찰할 수 있지만, 상하는 image가 잘린 느낌이 난다. 위의 Figure 2.2에서 원본 image를 256×256 으로 변환하였지만 좌우는 회전할 때 공간이 있어서 안정적으로 회전하는 반면, 상하는 그림이 빼곡이 차있으므로 공간이 부족하기에 잘리게 된다.

Exercise 2

Reconstruct your image using the backprojection method only. Display the original image and reconstructed image, Compare the mean square error. Discuss on the reconstructed image and its quality and errors. Hint: use `iradon.m` for reconstruction, but you will need to disable the digital filter.

Exercise 2는 Exercise 1에서 얻은 sinogram으로 image를 reconstruction 하기 위해 back-projection을 진행하는 활동이다. Back-projection을 진행하면 sinogram의 data들이 projection시에 attenuation된 부분만 복원이 되는 것이 아니라 모든 지점에 복원이 되기 때문에 blurring이 일어난 image로 복원이 된다.

아래 Figure 3.1과 Figure 3.2를 비교해보면 그 차이를 확실하게 알 수 있다. Figure 3.1에선 image의 경계가 명확한 사람 모양을 나타내고 있는 반면, Figure 3.2에선 image의 경계가 확실히 보이지 않고 blurring 현상이 일어나서 알아볼 수 없게 왜곡되어 있다.



Figure 3.1 512×512 sized original image

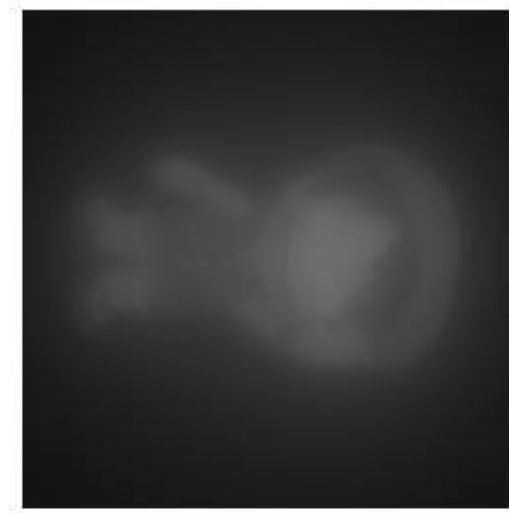


Figure 3.2 Non-filtered-back-projection image

Table 1.1 Original image와 back-projection image 사이의 MSE 값

Name	Value
MSE_Back-Projection	101.4704

위의 Table 1.1에서 확인할 수 있듯이, 이러한 점을 수치적으로 확인하기 위해 original image와 back-projection image 사이의 MSE 값을 구한 결과 약 101.4704이라는 매우 큰 값이 나왔고, 이는 원본 image가 제대로 복원되지 않았음을 뜻한다.

Exercise 3

You are going to reconstruct your original image using `iradon.m` (i.e., filtered back-projection), but before reconstructing the image

- A. `iradon.m` has 5 filter options. Plot their filter responses, and compare their characteristics.
- B. Choose the best filter for your image based on the mean square error. Plot one projection before and after filtering in one figure. What seems to be an effect of filtering? Discuss about the effect of digital filtering.

Exercise 3에서는 Exercise 2와는 달리 non-filtered-back-projection이 아니라 `iradon.m`에서 구현할 수 있는 5가지의 filter를 사용하여 back-projection을 진행하였다. `Iraddon` function에서는 H1부터 H5까지의 filter를 구현할 수 있는데, 각각의 filter는 아래의 Table 2.1과 같다.

Table 2.1 Filter name of H1~H5

Iraddon Function	Filter Name
H1	Ram-Lak filter
H2	Shepp-Logan filter
H3	Hamming filter
H4	Hann filter
H5	Cosine filter

아래 Figure 4.1부터 Figure 4.10은 각각의 filter들을 주파수 대역에 따라 그래프와 적용했을 때의 image를 나타낸 것이다. 여기서 x축의 frequency는 원본의 image를 복원하기 위한 최소 비율은 Nyquist Ratio를 기준으로 하고 있고 y축은 filter의 gain 값을 나타낸다.

모든 filter들은 250Hz를 기준으로 대칭적이고 High-Pass-Filter와 비슷한 역할을 한다. Ram-Lak filter는 sharp한 특성을 띠고 있으며 spatial resolution이 좋다는 장점이 있다, 그러나 단점으로는 noise가 심한데 이를 보완한 filter가 Shepp-Logan filter로 Ram-Lak filter와는 다르게 smooth하다. 하지만 이 filter 또한 Gibb's phenomenon을 유발할 수 있다는 단점이 있다. Hamming filter의 경우 Shepp-Logan filter와 같이 smooth하지만 spatial resolution이 낮다는 단점이 있다. Hann filter와 Cosine filter 경우 Hamming filter와 유사하지만 중간 대역의 주파수를 어떻게 깎아주는가에 따라 나뉘는 것을 알 수 있다.

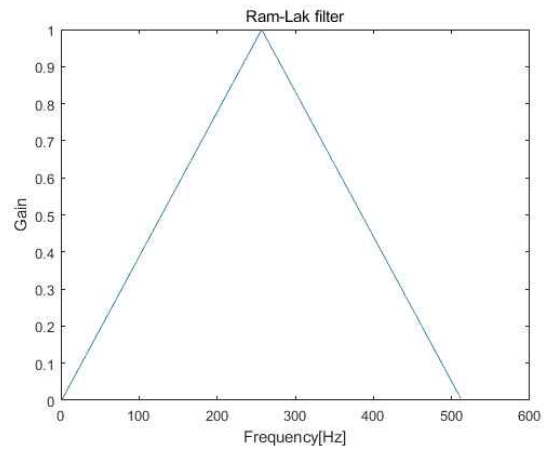


Figure 4.1 Ram-Lak filter

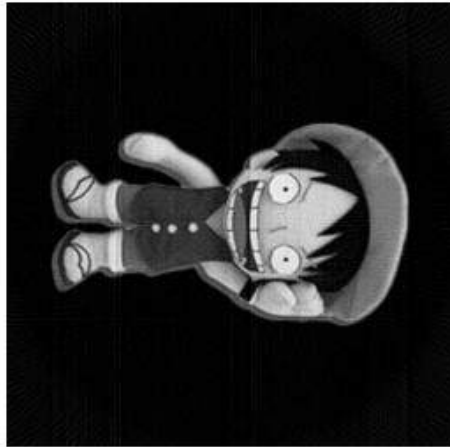


Figure 4.2 Ram-Lak filter를 이용한 filtered-back-projection image

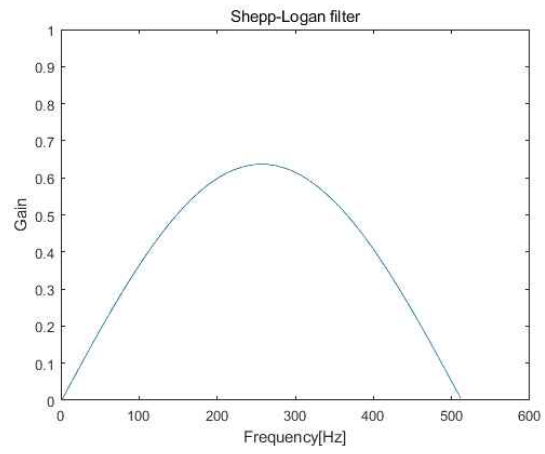


Figure 4.3 Shepp-Logan filter

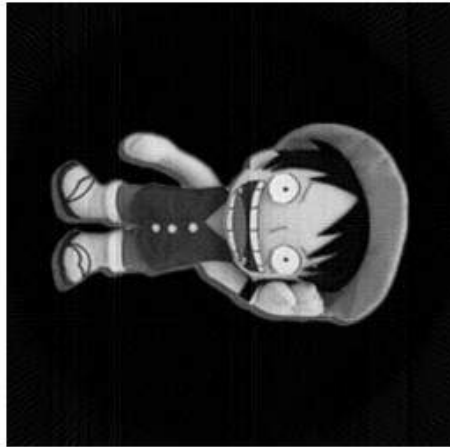


Figure 4.4 Shepp-Logan filter를 이용한 filtered-back-projection image

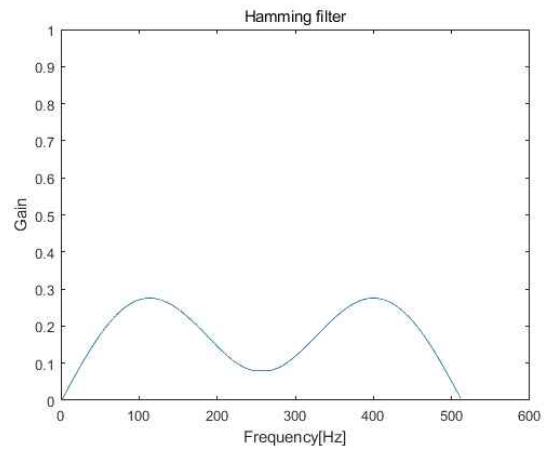


Figure 4.5 Hamming filter

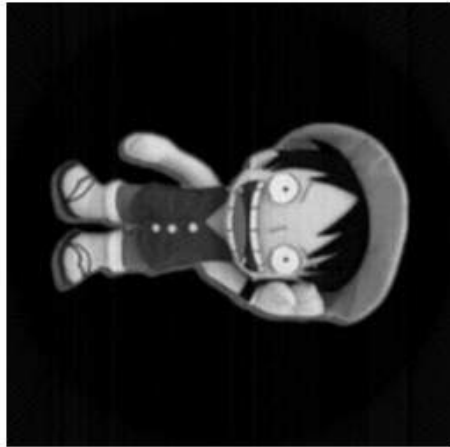


Figure 4.6 Hamming filter를 이용한 filtered-back-projection image

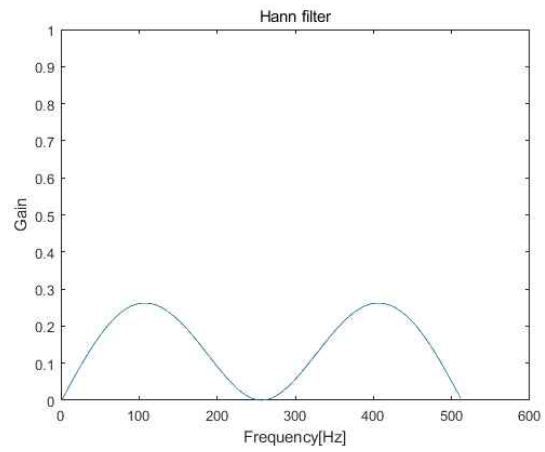


Figure 4.7 Hann filter

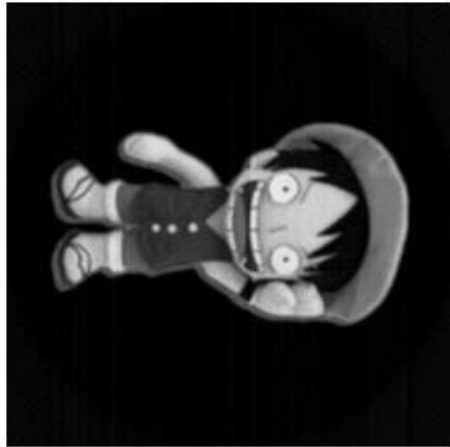


Figure 4.8 Hann filter를 이용한 filtered-back-projection image

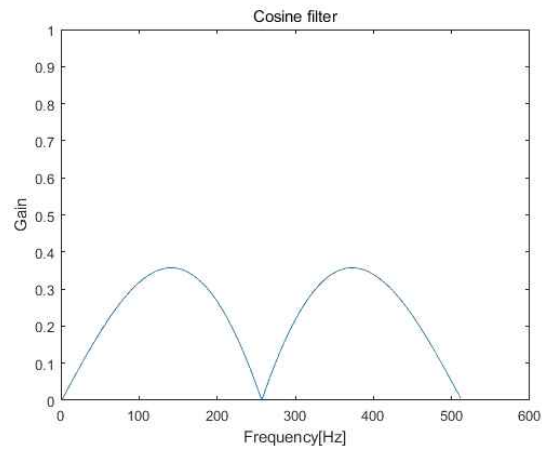


Figure 4.9 Cosine filter

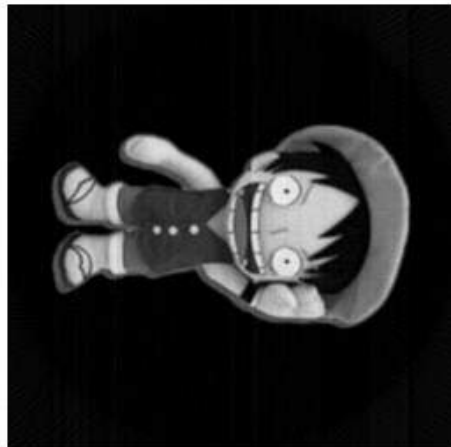


Figure 4.10 Cosine filter를 이용한 filtered-back-projection image

Filtered-back-projection을 진행했을 때 Exercise 2보다 훨씬 선명하게 spatial resolution이 향상된 것을 확인할 수 있다. 하지만 optimal한 filter를 찾기에는 육안으로 큰 차이를 구별할 수 없다. 따라서 아래의 Table 2.2를 보면 가장 optimal한 filter를 찾기 위해 MSE 값을 구한 결과를 확인할 수 있다. Filter를 적용하여 얻은 image와 원본 image를 비교하여 구한 MSE 값이 가장 작은 것은 Hann filter를 적용했을 때로 그 값은 약 100.0304이고 가장 큰 값은 Ram-Lak filter로 약 204.2605이다. 따라서 image에 가장 optimal한 filter는 Hann filter라는 것을 알 수 있다.

Table 2.2 Filter마다 back-projection을 진행했을 때 원본 image와의 MSE 값

Filter Name	MSE Value
Ram-Lak filter	204.2605
Shepp-Logan filter	171.8804
Hamming filter	105.8004
Hann filter	100.0304
Cosine filter	123.7304

Exercise 4

Reconstruct your original image with the sinogram from Exercise 1. Try the best digital filter and all interpolation options of iradon.m in reconstruction and find the best interpolation option again in the mean square error sense. What is the effect of the interpolation options for reconstruction. (you should be careful with the size of the reconstructed image and the original image. Check the image size.)

Exercise 4는 interpolation(보간법) 방식의 차이에 따른 image quality의 변화를 알아보는 활동이다. Exercise 3에서 보았듯이 가장 optimal한 filter는 Hann filter이기 때문에 filter를 고정한 상태에서 각각 'nearest', 'linear', 'spline', 'pchip', 'cubic', 'v5cubic'의 6가지 interpolation 방식을 이용하여 iradon function을 진행하였다. 주의해야할 점은 size가 256×256인 원본 image와는 달리 interpolation을 바꾸면서 iradon을 진행할 경우 reconstruction image의 size는 180×180이 되기 때문에 iradon function을 이용할 때 size를 256으로 설정해주었다. 각각의 interpolation을 하였을 때 image quality를 수치적으로 확인하기 위해 MSE 값을 구한 결과는 아래 Table 3.1과 같다.

Table 3.1 Interpolation 별 MSE 값

Interpolation Option	MSE Value
nearest	101.1004
linear	100.0003
spline	100.8404
pchip	100.4804
cubic	100.3807
v5cubic	100.7704

가장 작은 MSE 값을 보이는 interpolation option은 linear으로써 그 값은 약 100.0003이고, 가장 큰 MSE 값을 보이는 interpolation option은 nearest로써 그 값은 약 101.1004이다. 결과적으로 내 image quality를 가장 좋게 할 수 있는 interpolation option은 linear임을 알 수 있으며 filter의 종류만 바꿨을 때보다 interpolation option까지 같이 조정했을 때 MSE가 모두 감소하였다.

Exercise 5

Create projection data with THREE different values of the rotation angle. Reconstruct the image. What is the effect of the number of projections? Discuss the effect of increasing the number of projections. What would you say on the effect of increasing the projections (i.e., more angular views)? You should relate to Fourier slice theorem.

Exercise 5에서는 image를 회전시킬 때 angle step을 다르게 하여 reconstruction을 진행하는 활동이다. Exercise 4까지는 angle step을 1° 로 고정시키고 181개의 projection data를 기반으로 reconstruction을 진행하였지만 이번 Exercise 5에선 angle step을 1° 말고 5° , 10° 로 바꾸어 Hann filter, linear interpolation을 진행하였다. 아래의 Figure 5.1에서 Figure 5.3까지를 보면 angle step이 커질수록 image quality가 점점 낮아지는 것을 확인할 수 있다. 이는 angle step이 커질수록 projection data가 181개에서 37개, 19개로 점점 줄어들고 이로 인한 reconstruction 또한 적은 data 양을 기반으로 진행되기 때문에 spatial resolution이 낮아져 image가 왜곡되어 결과적으로 quality가 낮아지게 되는 것이다.

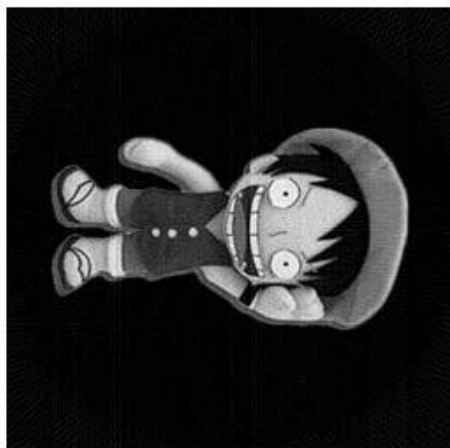


Figure 5.1 Reconstruction image when angle step is 1°

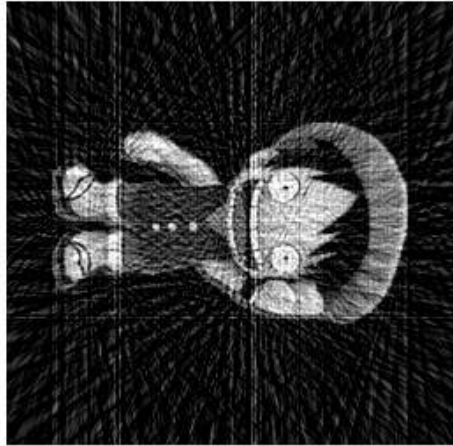


Figure 5.2 Reconstruction image when angle step is 5°

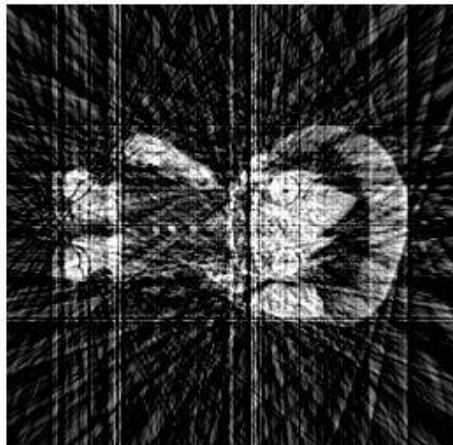


Figure 5.3 Reconstruction image when angle step is 10°

영상을 찍을 때, angle step을 너무 촘촘하게 하면 처리해야하는 data가 많아지기 때문에 image quality가 너무 나빠지지 않는 선에서 적당한 angle step을 찾아 회전해야 가장 효율적으로 reconstruction image를 얻을 수 있을 것이다.

Exercise 6

For the best reconstruction options out of Exercises 3 and 4, try image reconstruction by iradon on GPU. Compare the gains of performing image reconstruction between CPU vs. GPU (i.e., w/ GPU vs. w/o GPU).

```
>> gpuArray
```

다음 사용 중 오류가 발생함: gpuArray
지원되는 GPU 장치를 찾을 수 없습니다.

Figure 6.1 Error