

axial, CUDA dimension x, related variables(zldx, 편의상 z축으로 정의하기도 함)



lateral,
CUDA dim y,
related vars
(xldx)



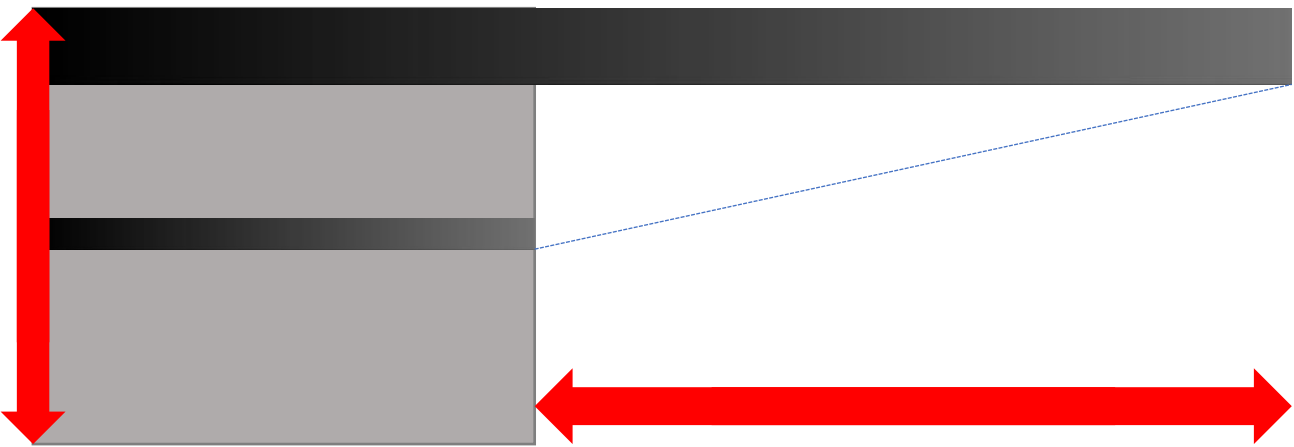
2000x128 data를 892x768 data에 그대로 넣으니
x 방향 데이터는 잘리고, y 방향 데이터는 여백임.

실제 신호 데이터의 경우 공간 비율(spatial ratio)를
바꾸지 않고 이미지를 변형할 수 있어야 함.

spatial ratio는 probe의 spec에 따라 매우 다름.
기본적으로 linear probe는 직각 사각형의 이미지
이지만 convex probe, phased array probe,
transvaginal probe 등 specific application에 사용되는
프로브는 그 형태에 맞게 scan conversion 해줘야 함

지금 구현하는 것은 linear probe scan conversion
이므로 직사각형 형태이며, 일부 검은색 여백이
양쪽에 출력될 수 있음.

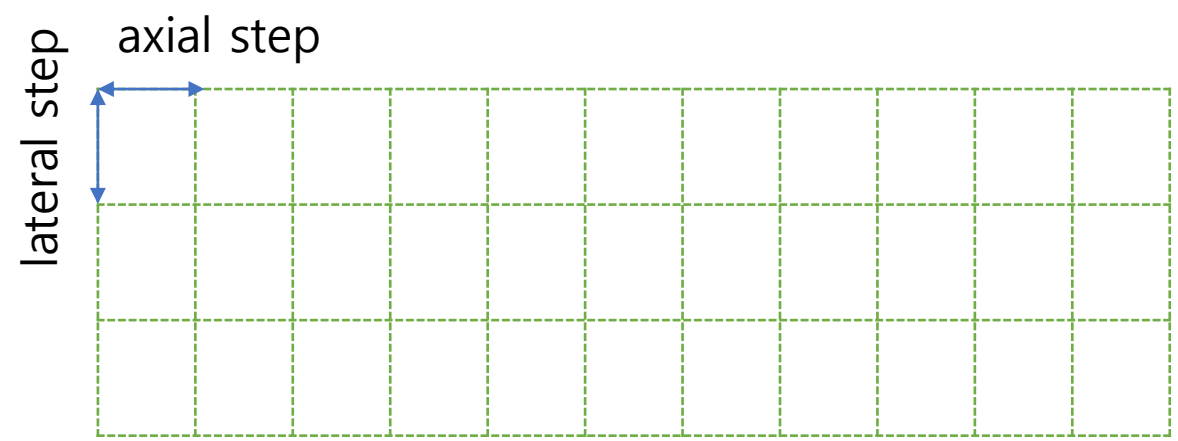
advice: input data size가 2000x128이면 가로로 길쭉한 화면밖에 볼 수 없음.
decimation을 통해 axial size를 줄일 필요가 있음



1. 데이터의 영역이 회색 영역에 맞춰져야 함
2. 데이터의 spatial ratio를 지켜야 함
(input과 output grid의 픽셀당 거리 계산 필요)

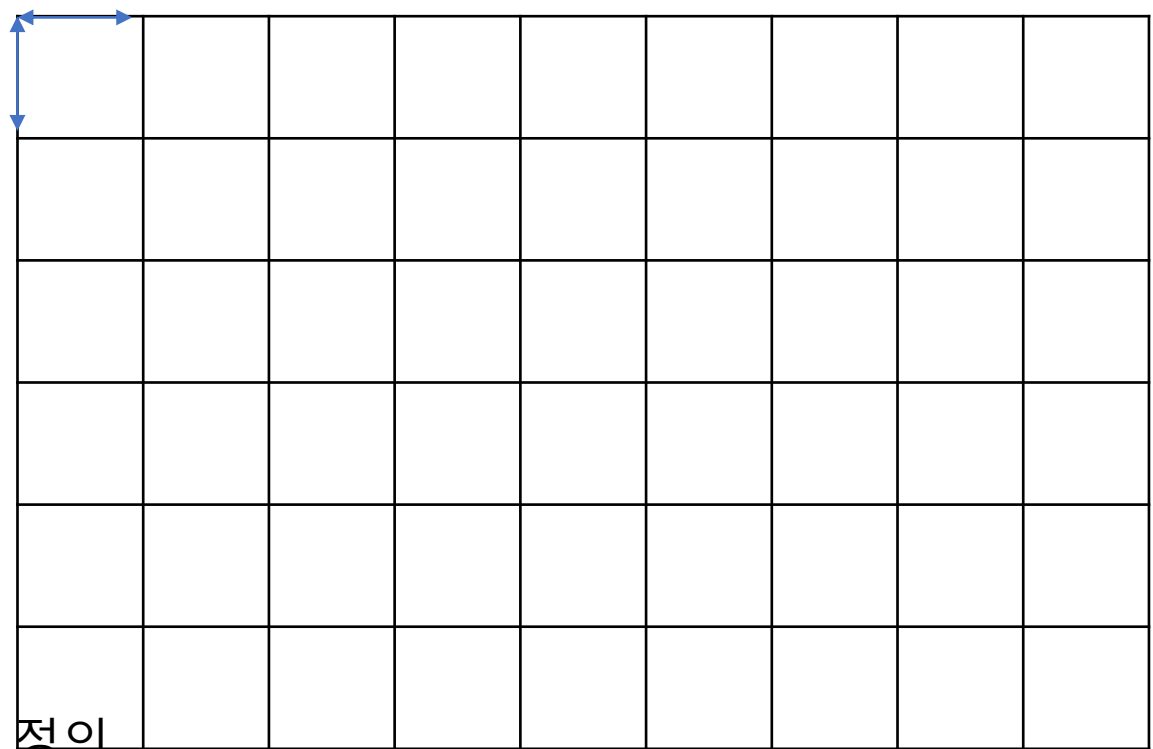
output grid step은 input의 axial, lateral step중 큰 쪽을 선택

Input data의 grid matrix

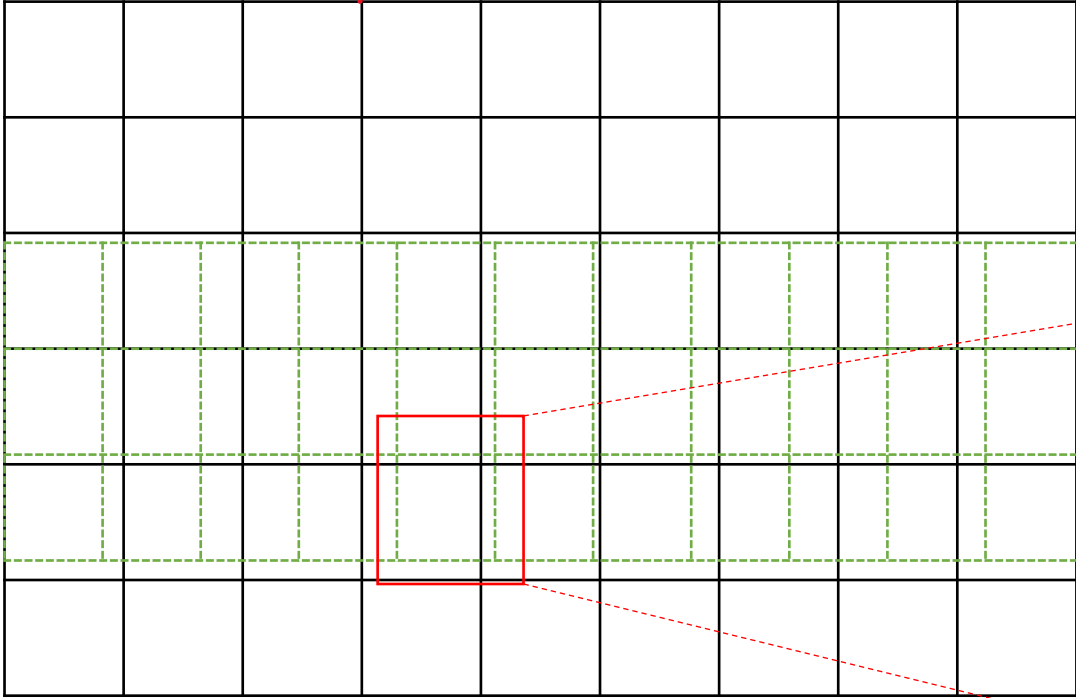


axial step * sample num / output width
 < lateral step * element num / output height 이기 때문에
 lateral step * element num / output height 을 grid matrix로 정의

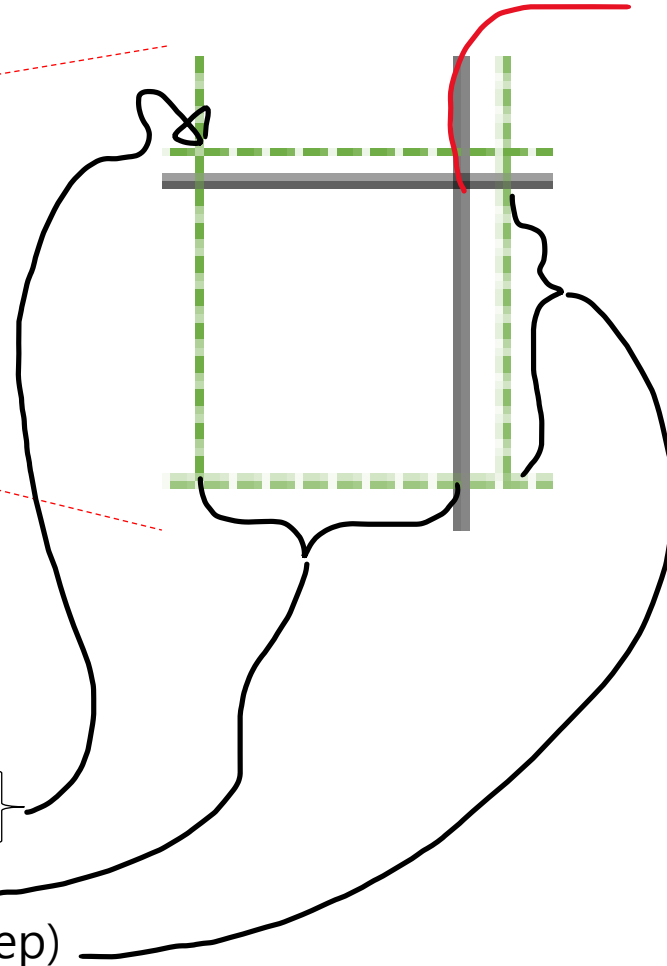
output data의 grid matrix



각 tile의 모서리에 data가 있다고 가정함



(x_i, y_j) 좌표
(i, j)번째 좌표의 데이터는
이웃 초록 tile에 있는 데이터의
비율 합임



검은 선의 교차 지점 거리 구하기

$\text{width_length}_i(\text{m}) = k * \text{grid step}$

j 번째 $\text{height_length}_i(\text{m}) = k * \text{grid step}$

초록 선의 이웃 지점 인덱스 구하기

$\text{width_index} = \text{floor}(\text{width_length}_i / \text{axial step})$

$\text{height_index} = \text{floor}(\text{height_length}_i / \text{lateral step})$

$\text{width_fraction} = \text{modf}(\text{width_length}_i / \text{axial step})$

$\text{height_fraction} = \text{modf}(\text{height_length}_i / \text{lateral step})$

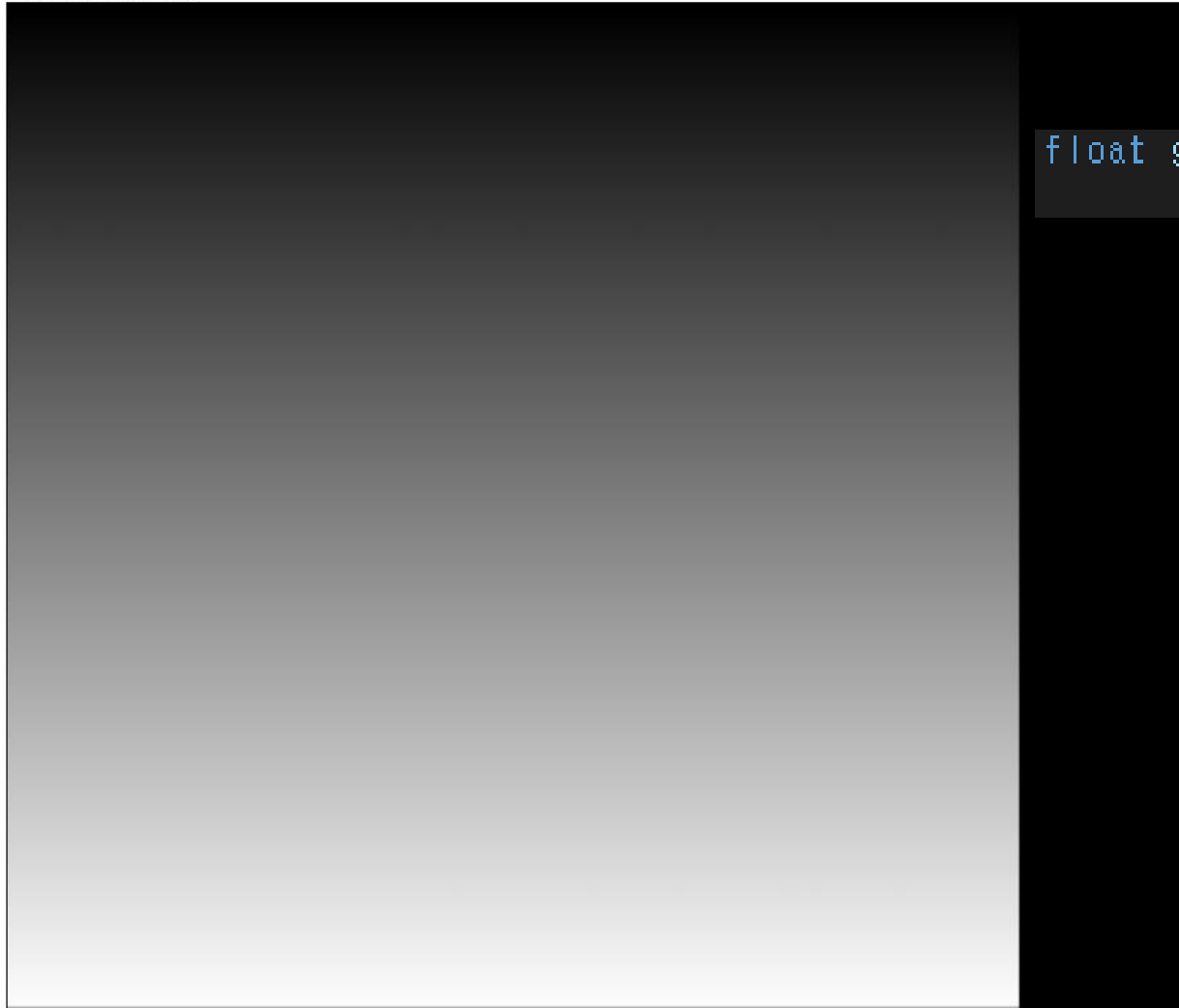
input_500x128_imageType_8-bit.raw (150%)

500x128 pixels; 8-bit; 63K

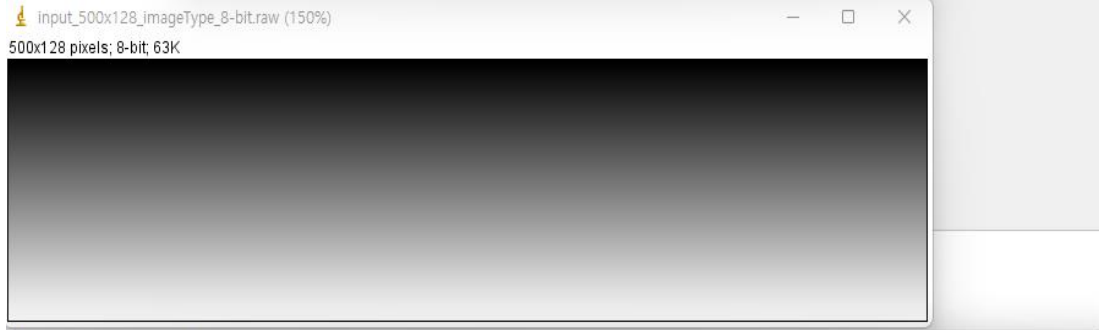


output_892x768_imageType_8-bit.raw

892x768 pixels; 8-bit; 669K



```
float gridStep = fmaxf(axialStep * (float)width / (float)outWidth,  
                        pitchM * (float)height / (float)outHeight);
```



```
float gridStep = fminf(axialStep * (float)width / (float)outWidth,  
                        pitchM * (float)height / (float)outHeight);
```

width에 맞추다보니 height가 짧림
어느 크기에 맞출 것인가에 따라 설계를 잘 해야함.