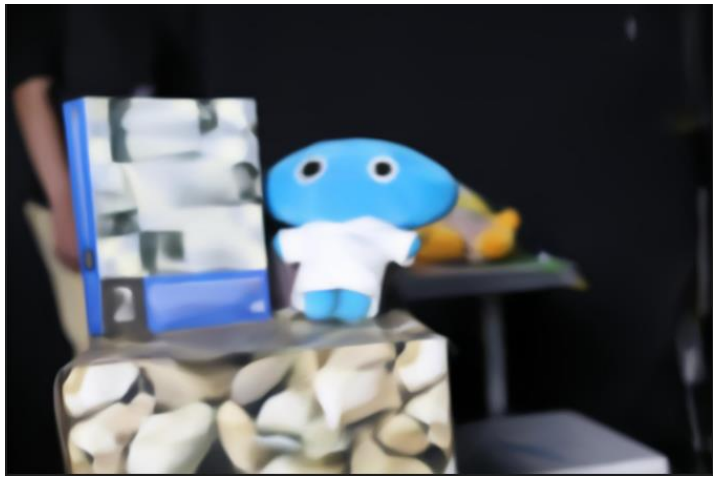
# 개인미팅 자료

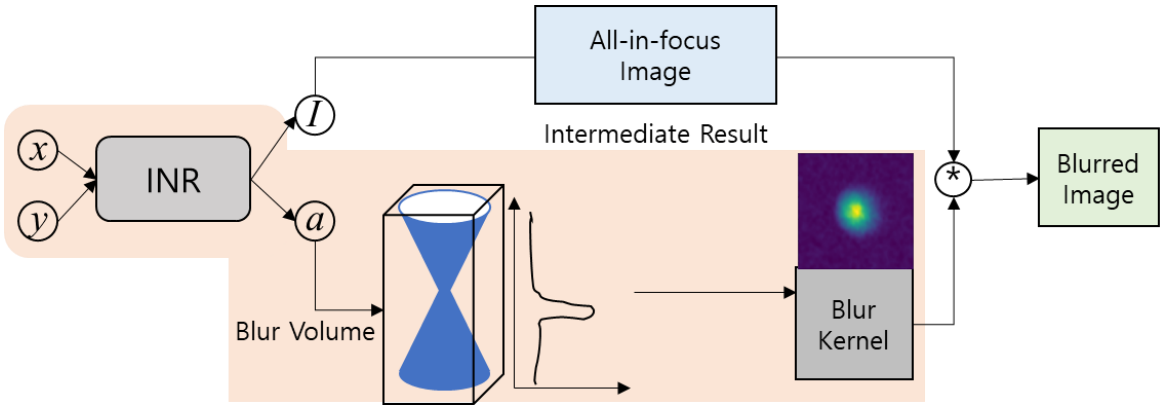
# 현재 진행된 사항

- Dual Pixel관련 진행

  - Calibration 관련해서는 Blur Volume Optimization, INR representation등의 방식으로 테스트 한것으로 전달받음
  - 뒷부분인 Dual-Pixel based INR 관련해서 진행된 건 INR으로 All-in-focus이미지 추정, 이후 All-in-focus image와 기존에 있던 Blur Volume을 사용하여 각 픽셀 위치에서 alpha (depth) 값 추정
  - 하지만 오래 학습해도 아래 framework로는 alpha/depth 추정 불가;

    전체적 프레임워크가 INR/NeRF등의 방식에 있어 비교적 naïve 하다고 판단됨

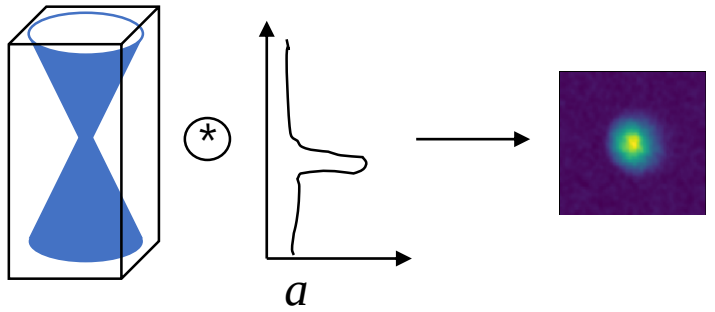


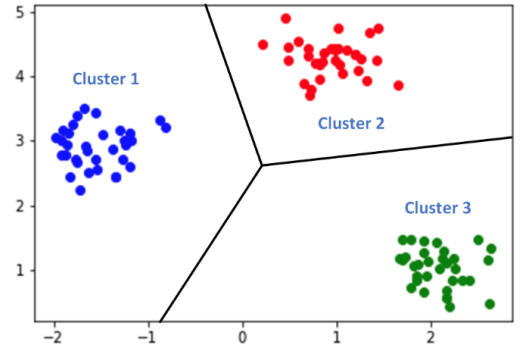


# 현재 진행된 사항

- 현 방식의 추측되는 문제점

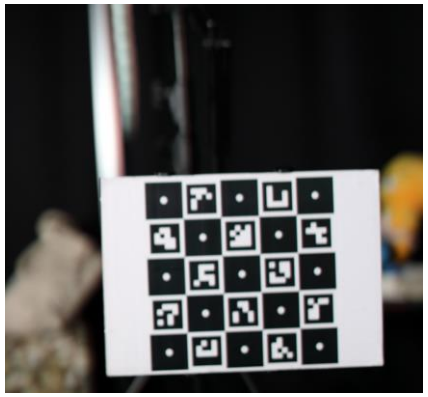
  - 해당 방식에서 학습이 안되는 이유 중 하나는 Blur Volume이 특정한 parameter들로 모델 된 방식이 아닌 calibration data으로부터 sampling하여 취득한 일종의 "lookup" 방식으로 진행하였음 (Volume Optimization, not Model Optimization)
  - INR의 Alpha은 Blur Volume의 depth channel 길이를 가지는 1-D Vector로 되어있음 (일종의 Alpha Confidence 값)
  - Alpha값과 blur volum을 product-sum 하여 sum(3D * 1D) → 2D blur kernel 방식으로 Blur Kernel을 추출함 (Classification 에서의 CrossEntropy와 비슷한 방식)
    - 이미지로부터 Alpha 값으로 back-propagation이 가능하도록 하기 위해 이렇게 설계함
    - 하지만 classification 문제에서의 class distribution에 비해 Blur Volume의 depth 별 blur kernel 들의 차이가 비슷하다 보니 각 픽셀마다 적용해야 되는 Blur Kernel들을 찾아 내지 못함.
  - **Blur Kernel*Alpha Confidence의 Sum을 Blur Kernel로는 각 픽셀의 대응되는 alpha를 찾기 어려움**



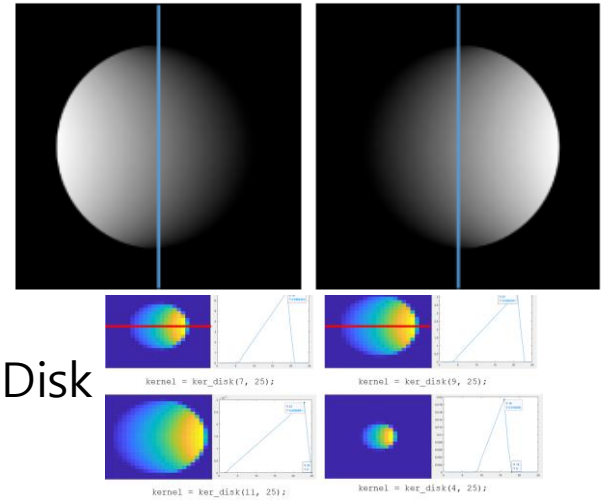



# 현재 진행된 사항

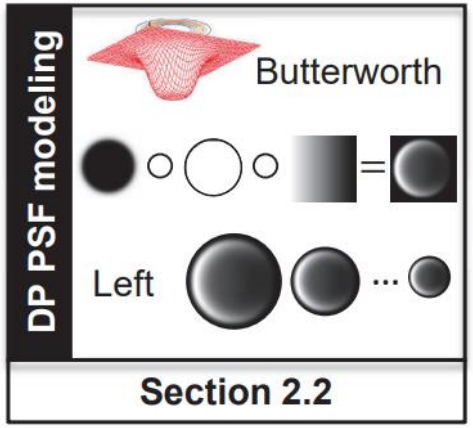
- 새로운 방식 제안
  - 기존에는 Blur Volume Calibration 따로, Application 따로 였음; 이때 Blur Volume으로부터 Alpha를 학습하기 위한 Back-propagation 할 수 있도록 설계한 방법으로는 Alpha 값을 학습시키기 어려움
  - 새로운 방식으로 Calibration과 Application을 동시에 진행 되도록 설계 제안; 하지만 기존의 Blur Volume은 Calibration 이미지들로 얻은 것이다 보니 실제 non-calibration 이미지 들에서는 학습이 어려움
  - Calibration 방식을 Volume Optimization이 아닌 Model Optimization으로 바꾸면 다양한 이미지들에서도 Blur Kernel Model을 Optimization할 수 있게 됨
    - 기존에 있던 방식 중에서 Model Optimization은 Sliding Disk(ICCP '20) 방식, Butterworth Filter (ICCV '21) 등의 방식이 있음



Sliding Disk



Butterworth Filter



# 현재 진행된 사항

- 새로운 방식 제안

  - 하지만 기존의 model based 방식은 dual pixel에 있어서 좌우로 쏠리는 현상들을 단순히 sliding 하면서 생기는 패턴, 좌우 gradient등으로만 표현을 하는데, 이게 실제 모델이랑 흡사하게 표현하는 방법인지?
  - 보통 Blur Kernel을 적용할 때 많이 쓰이는 방식 중 하나 → Gaussian Blur
  - 쏠리는 현상들을 표현하기위해 제안하는 새로운 방식 → Blur와 쏠림을 같이 표현할 수 있는 Skewed Gaussian Distribution
  - Skewed Gaussian Distribution:

Let $\phi(x)$ denote the standard normal probability density function

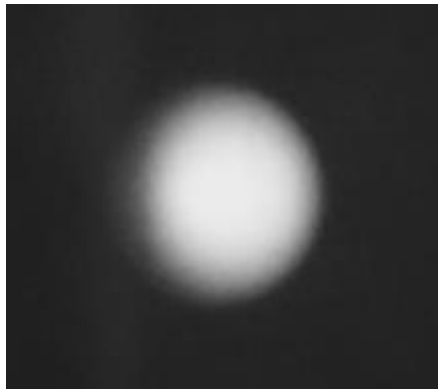
$$\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

with the cumulative distribution function given by

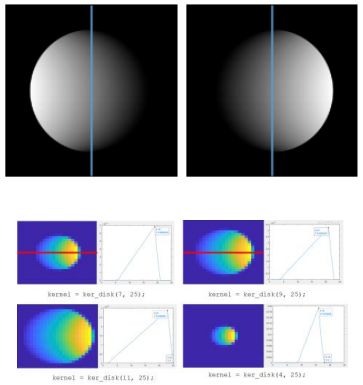
$$\Phi(x) = \int_{-\infty}^{x} \phi(t)\, dt = \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right],$$

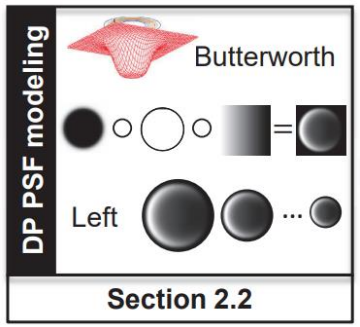
where "erf" is the error function. Then the probability density function (pdf)

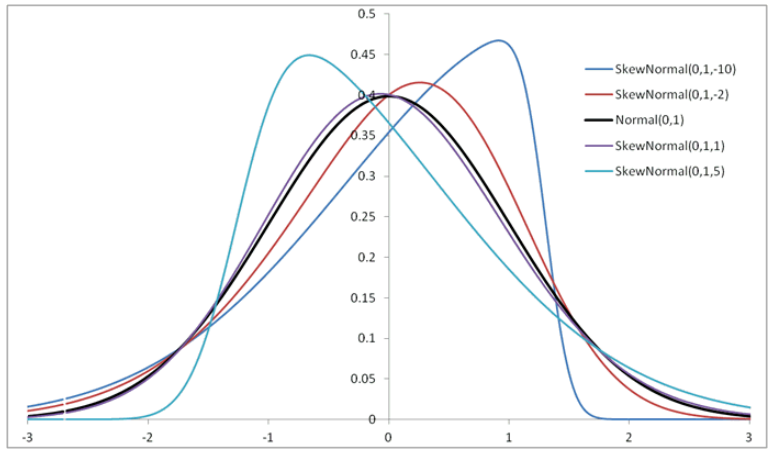
$$f(x) = 2\phi(x)\Phi(\alpha x).$$

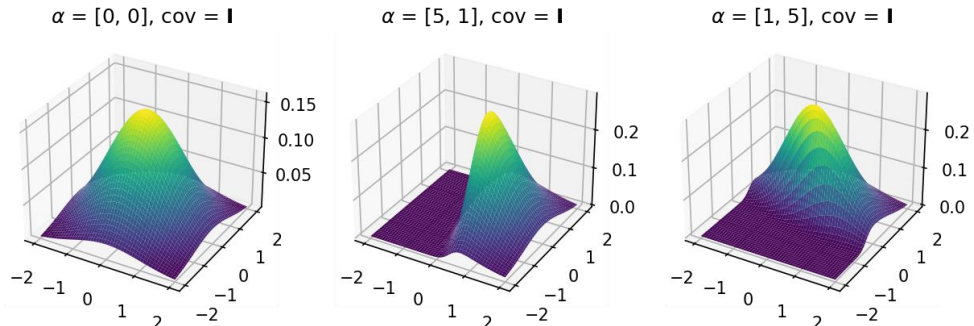


Real Calibration Disk





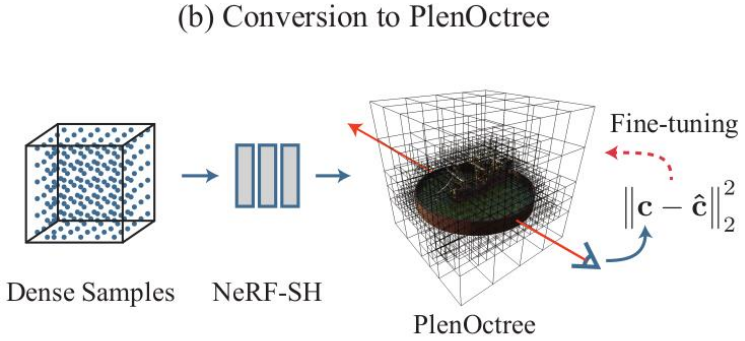



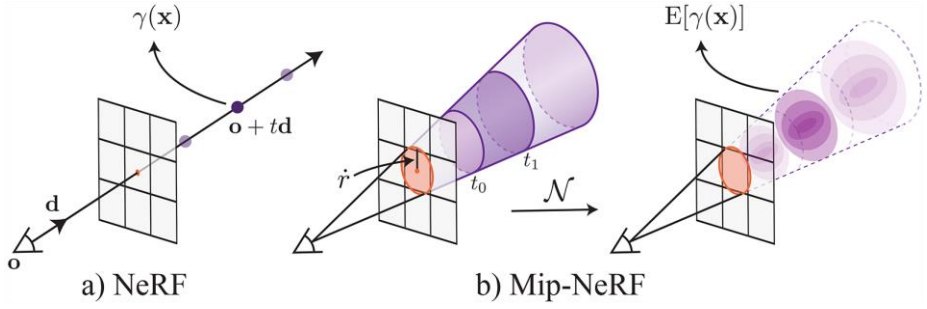



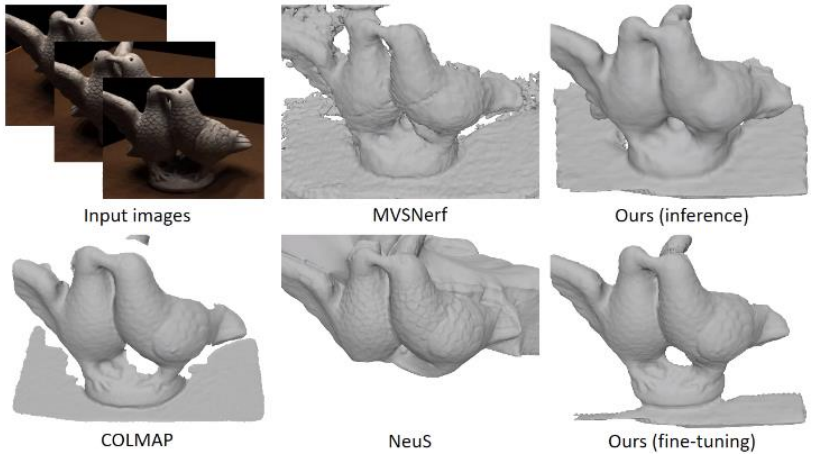
# 현재 진행된 사항

- 새로운 방식 제안
  - 적용 Application도 기존에는 INR 기반으로 All-in-focus image reconstruction과 Pixel-wise Alpha를 구하는거 였음
  - 하지만 INR으로 복원하려는 것은 너무 단순하다 판단되어 최근 NeRF관련 논문들에서 사용하는 technique들을 적용할 예정
    - Voxel-based optimization 방식들이 속도가 빠름; 단순히 균등한 Voxel grid 대신 Spatially Varying Voxel (PlenOctree와 비슷한) 방식으로 하여 "in-focus"인 곳에서는 fine-voxel이, "out-of-focus"인 부분들에서는 coarse-voxel이 optimization 되도록 설계
    - 그리고 Blur Kernel을 Skewed Gaussian Distribution으로 표현하면서, Mip-nerf와 같이 ray-tracing을 할 때 하나의 ray가 아닌 distribution을 tracing하는 방법을 응용해 NeRF 학습; 이때 Skewed Gaussian의 parameter들도 optimization 진행
  - Single-View NeRF ? Neural Surface Representation (NeuS)?

https://vita-group.github.io/SinNeRF/



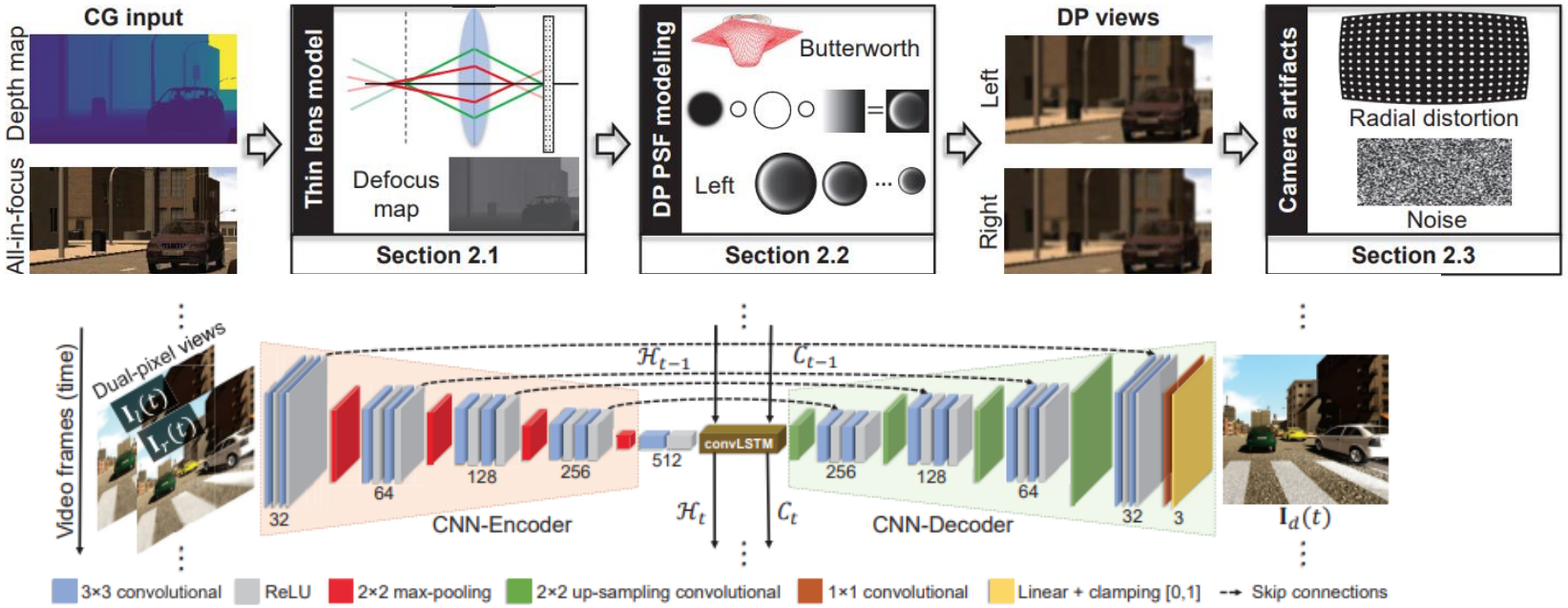



# 추후 진행사항

- 순차적으로 진행되어야할 사항들

  - Skewed Gaussian Kernel 모델링 (현재 기존 pytorch 등 library에 해당 함수 없음, 구현 필요)
  - 기존 Butterworth Filter 방식에 비해 Skewed Normal Distribution 성능이 더 좋은지 확인필요 → Synthetic Data에 적용
  - Skewed Gaussian Kernel으로 CVPR에 해당되는 Novelty 부족?? 새로운 네트워크 설계??
  - 논문의 focus를 어떻게 잡을지 결정 필요
  - Dual-Pixel, Single-View, NeRF (NeuS?) Architecture 정의



# 기타