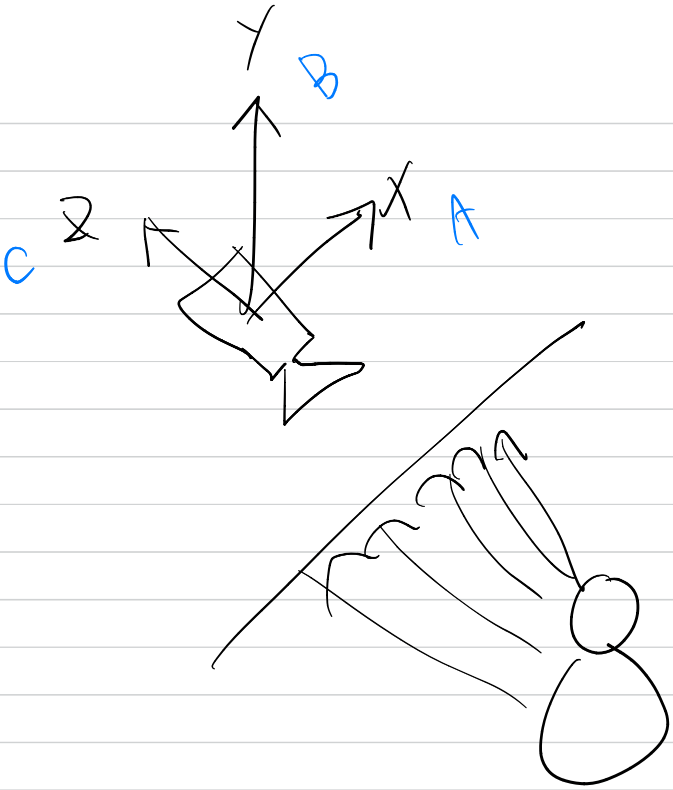
NeRF 코드 분석 필기

<카메라 방향 추출>



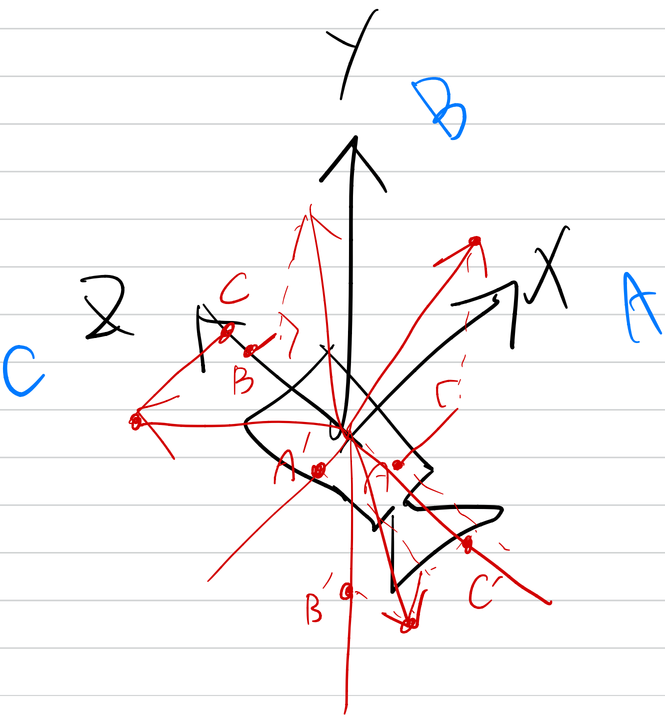
A

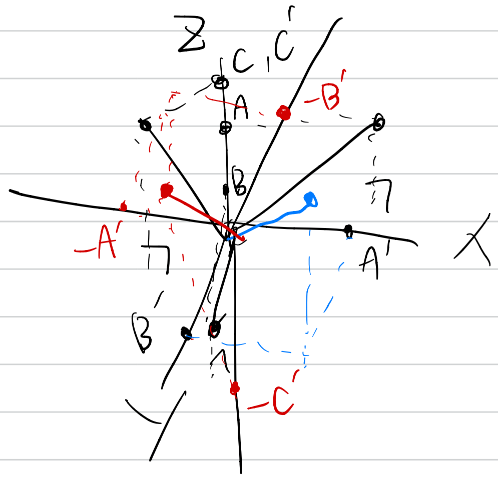
$$\begin{array}{c} x축 \\ y축 \\ z축 \end{array}\begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \begin{array}{c} \to x \\ \to y \\ \to z \end{array} = \begin{bmatrix} -R_{13} \\ -R_{23} \\ -R_{33} \end{bmatrix} \begin{array}{c} \to x \\ \to y \\ \to z \end{array}$$

B

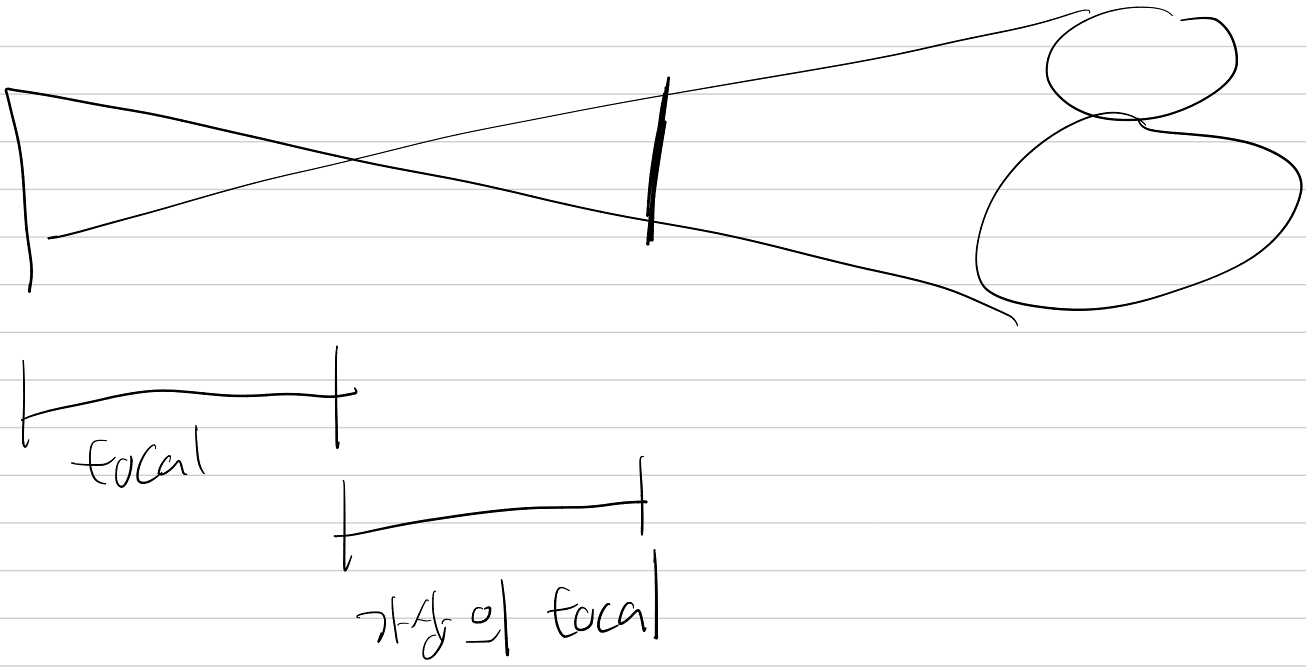
C

3×1　　　3×1





<NDC, 실제 물체와 이미지 간 차이 정규화>



< Find origin and direction of ray through every pixel and camera origin >

모든 픽셀과 카메라의 원점을 통해 ray의 원점과 방향 찾기

meshgrid

2개의 2차원 텐서

width = 100

[0, 1, . . . , 99]

↓

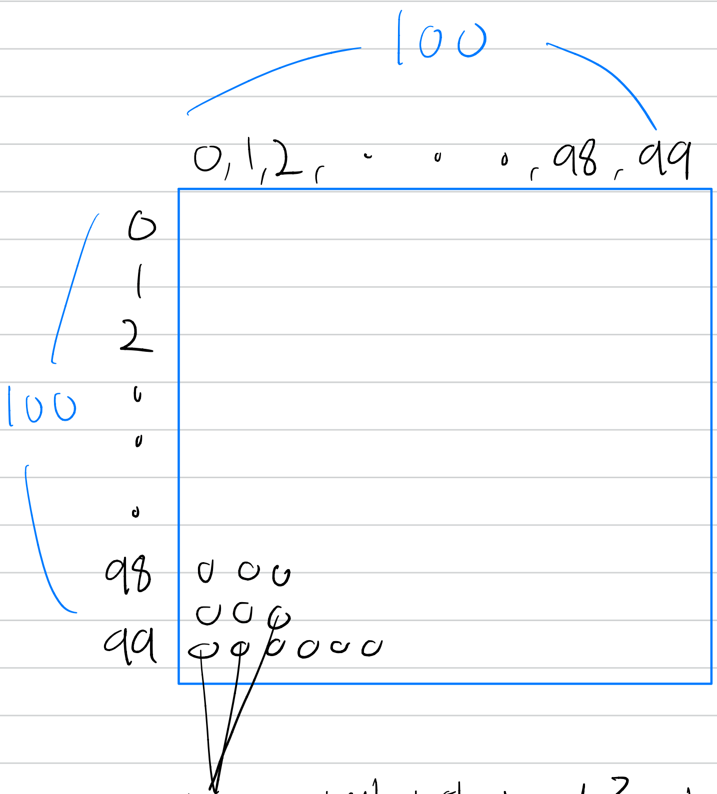
[0, 0, . . . , 0]
[1, 1, . . . , 1]
[99, 99, . . . , 99]

height = 100

[0, 1, . . . , 99]

↓

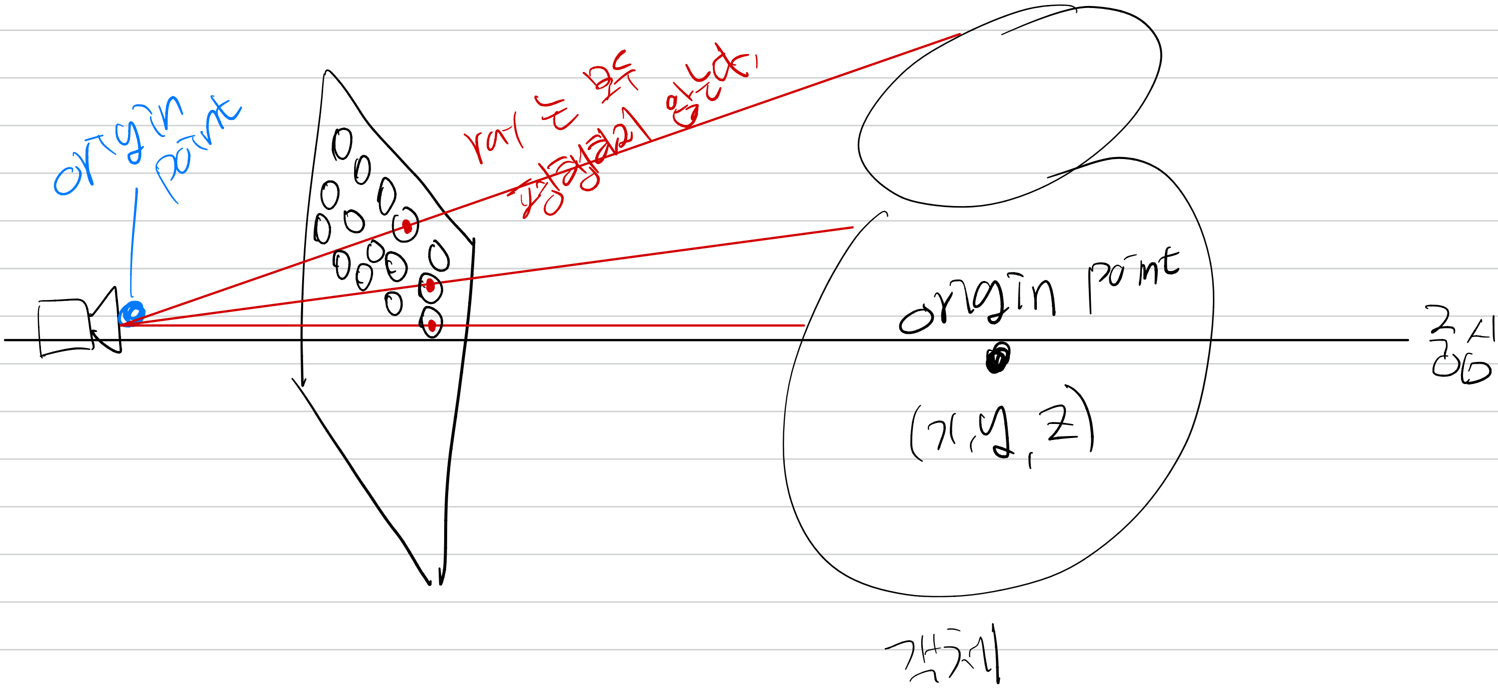
[0, 0, . . . , 0]
[1, 1, . . . , 1]
[99, 99, . . . , 99]



각각의 픽셀마다 ray를 가지고 있기 때문에 총 10,000개의 픽셀을 정의하는 과정이다.

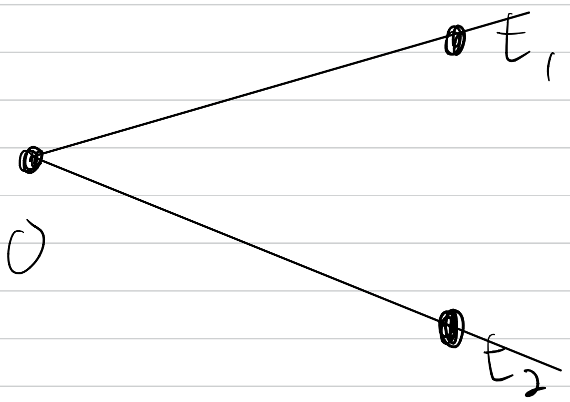
ray 구하는 과정에서 나오는 origin point는 객체의 위치인 원점을 의미하는 것인가? 카메라의 위치(렌즈의 중심)를 의미하는 것인가?

↳ 객체의 위치인 것으로 추측됨



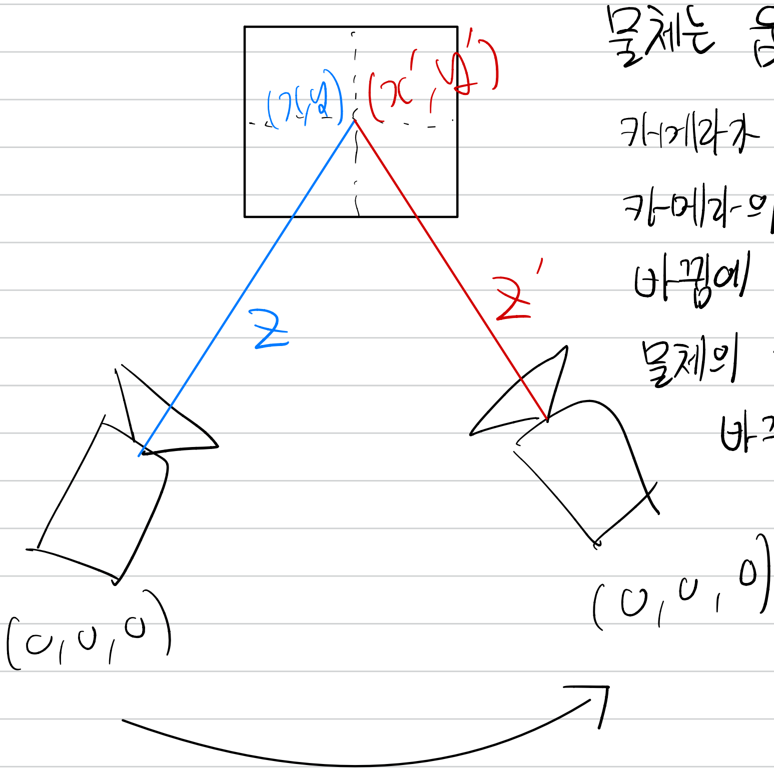
$$ray = o + td$$

o = camera pose,
d = direction
t = 픽셀 좌표



-0.05 , 3.84 , 1.20

Camera pose



물체는 움직이지 않았지만,
카메라가 움직였기 때문에
카메라의 기준 좌표인 (0, 0, 0)의 위치가
바뀜에 따라 카메라가 보고 있는
물체의 좌표도 (x, y, z)에서 (x', y', z')로
바뀌었다.

$(x, y, z) \longleftrightarrow (x', y', z')$

T 행렬 = Camera Pose
(회전 행렬)

3D 좌표에서 회전 변환 행렬 계산 $(x, y, z) \xleftrightarrow{T} (x', y', z')$

$$\begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} xR_{11} + yR_{12} + zR_{13} \\ xR_{21} + yR_{22} + zR_{23} \\ xR_{31} + yR_{32} + zR_{33} \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

$$= \begin{bmatrix} xR_{11} + yR_{12} + zR_{13} + t_x \\ xR_{21} + yR_{22} + zR_{23} + t_y \\ xR_{31} + yR_{32} + zR_{33} + t_z \end{bmatrix}$$

$$\therefore \begin{bmatrix} xR_{11} + yR_{12} + zR_{13} + t_x \\ xR_{21} + yR_{22} + zR_{23} + t_y \\ xR_{31} + yR_{32} + zR_{33} + t_z \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

호모지니어스 표현식 계산

$$\underbrace{\begin{bmatrix} R_{11} & R_{12} & R_{13} & t_x \\ R_{21} & R_{22} & R_{23} & t_y \\ R_{31} & R_{32} & R_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} xR_{11} + yR_{12} + zR_{13} + t_x \\ xR_{21} + yR_{22} + zR_{23} + t_y \\ xR_{31} + yR_{32} + zR_{33} + t_z \\ 0 + 0 + 0 + 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

↳ T 행렬 (회전, 평행이동 반영됨)

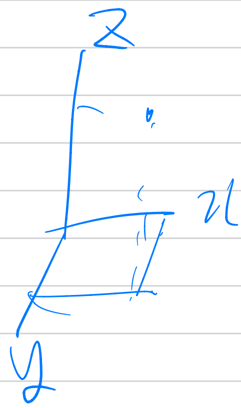
$= [R \mid t]$ 행렬

NeRF에서의 카메라 3D 좌표 행렬

$$\begin{bmatrix} R_1 & R_2 & R_3 & t_x \\ R_4 & R_5 & R_6 & t_y \\ R_7 & R_8 & R_9 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\sqrt{t_x^2 + t_y^2 + t_z^2}$$

= 평행이동 거리



$$[t_x^1, t_y^1, t_z^1]$$

$$[t_x^2, t_y^2, t_z^2]$$

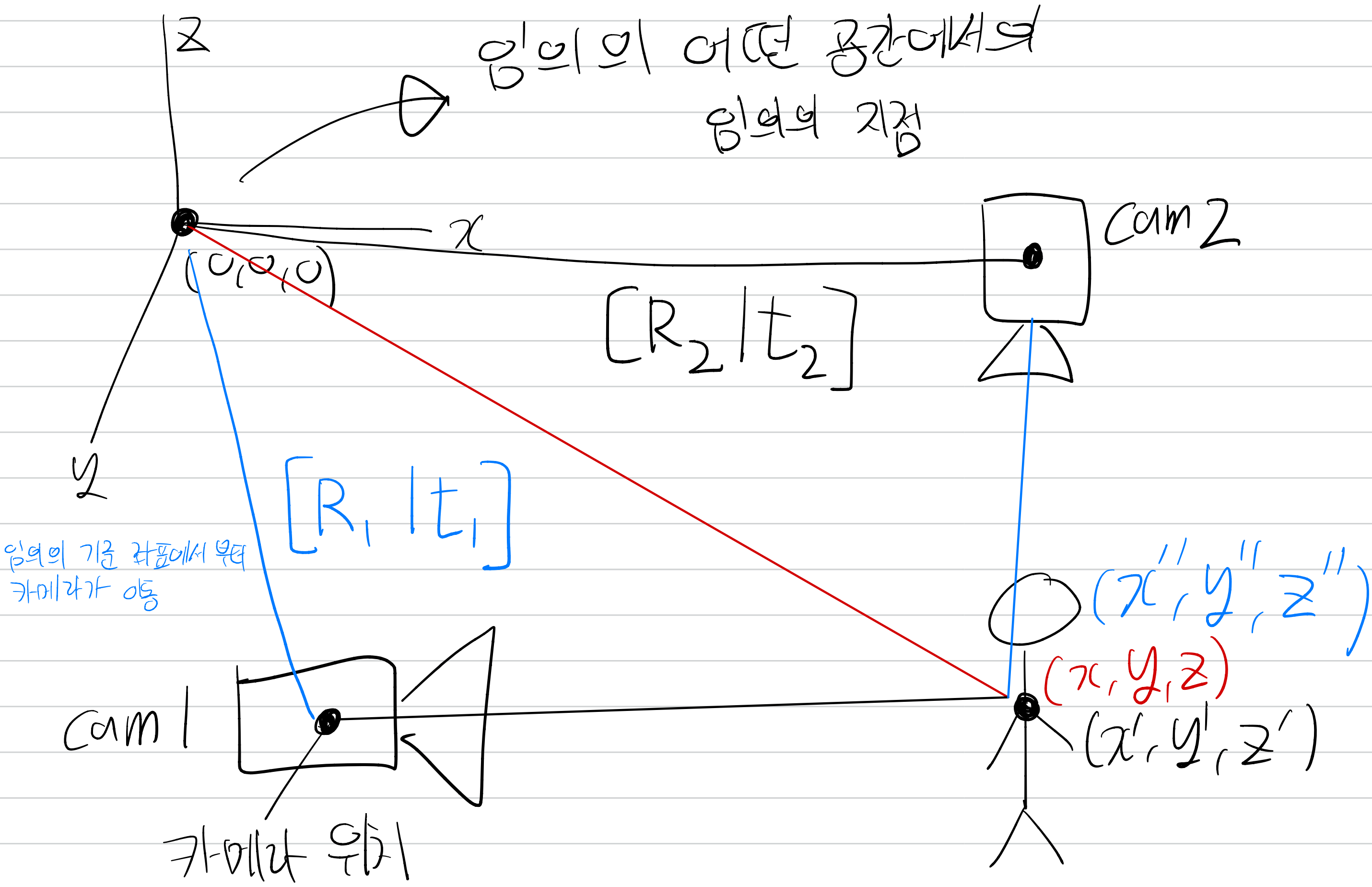
$$\vdots$$

$$[t_x^n, t_y^n, t_z^n]$$

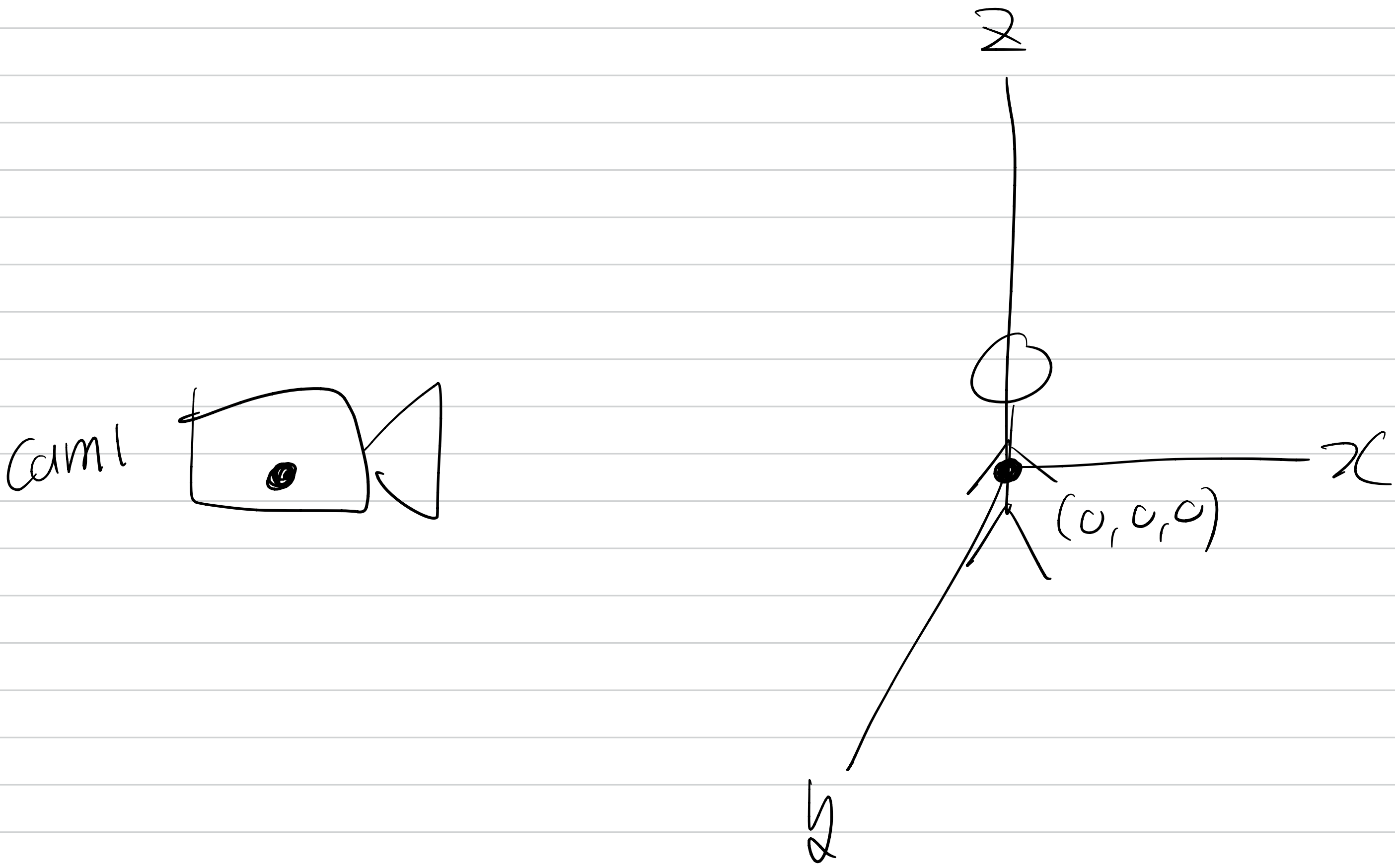
$$\rightarrow \quad \begin{matrix} [t_x^1, t_x^2, \cdots, t_x^n] \\ [t_y^1, t_y^2, \cdots, t_y^n] \\ [t_z^1, t_z^2, \cdots, t_z^n] \end{matrix}$$

Camera Pose는 한 개의 카메라가 동일한 물체를 찍는 과정에서 물체의 좌표는 고정된 상태로 두고, 카메라 위치가 바뀌면서 변화하는 좌표들 간의 변환행렬 [R | t]인데, NeRF에서는 카메라의 기준 좌표를 물체의 좌표로 설정하고, 카메라가 어떠한 위치에서 물체를 찍었을 때의 좌표 간의 차이에 대한 [R | t]를 Camera Pose로 보는 것인가?

아니면, 대상 물체나 카메라의 위치가 아닌 어떤 임의의 한 지점을 기준으로 설정하고, 그 좌표로부터 이동한 카메라의 좌표에 대한 차이를 [R | t]로 삼는 것인가?

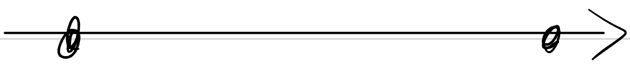


$$[R \mid t] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$


cam1
z
x
y
(0,0,0)

$1,0 - 0,1429$

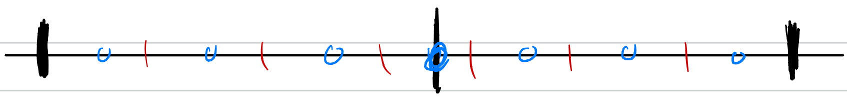
$= 0,8571$



$2,0$ $\qquad\qquad$ $6,0$

t_n $\qquad\qquad$ t_f

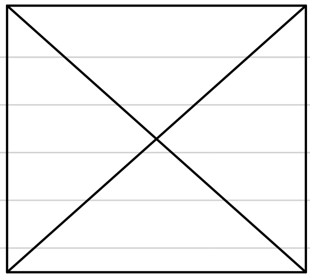
$1,0 \sim 0$ $\qquad\qquad$ $0 \sim 1,0$



100

0, 1, 2, ..., 99
0, 1, 2, ..., 99
⋮
0, 1, 2, ..., 99

100



(i-50)/focal_length

-(j-50)/focal_length

100

0, 0, ..., 0
1, 1, ..., 1
⋮
99, 99, ..., 99

100

-50, -49, ..., 49

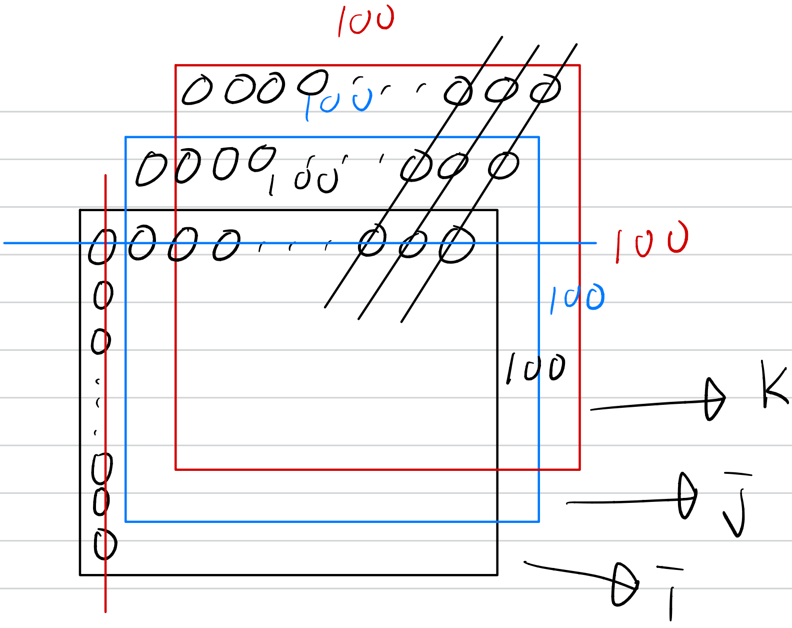
50, ..., 50
49, ..., 49

-49, ..., -49

[[1, 2], [3, 4]]

[[5, 6], [7, 8]]

[[1, 5], [2, 6]]
[[3, 7], [4, 8]]

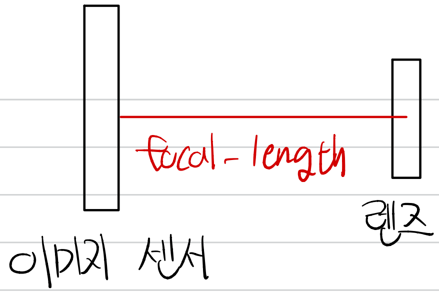


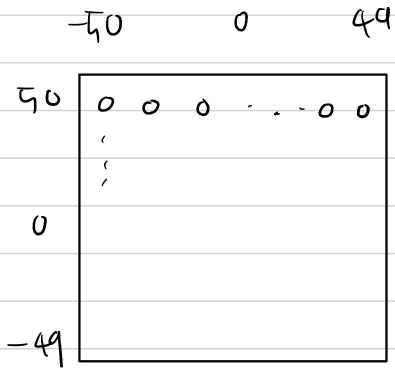
shape (100, 100, 3)

$\Uparrow$

stack((i, j, k), axis=-1)

(axis=2)





이미지

var : 10,000개

Camera Pose

$$\begin{bmatrix} r_{11}, & r_{12}, & r_{13} \\ r_{21}, & r_{22}, & r_{23} \\ r_{31}, & r_{32}, & r_{33} \end{bmatrix}$$

3

-23 0 23

-7 0 7

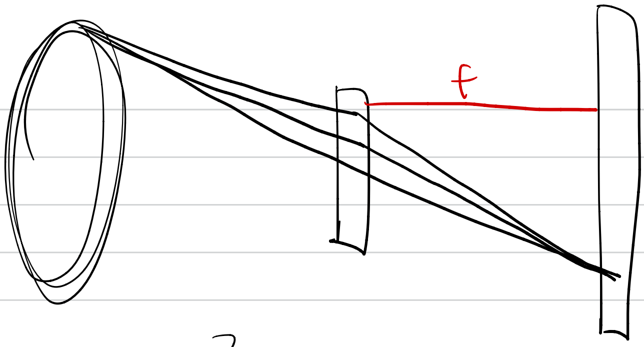
$$\Big[\boxed{[[-0.36,\ 0.36,\ -1,\ 0]]} \times \boxed{\begin{matrix}[[-r_{11}, & r_{12}, & -r_{13}] \\ [-r_{21}, & -r_{22}, & r_{23}] \\ [-r_{31}, & r_{32}, & r_{33}]]\end{matrix}}$$

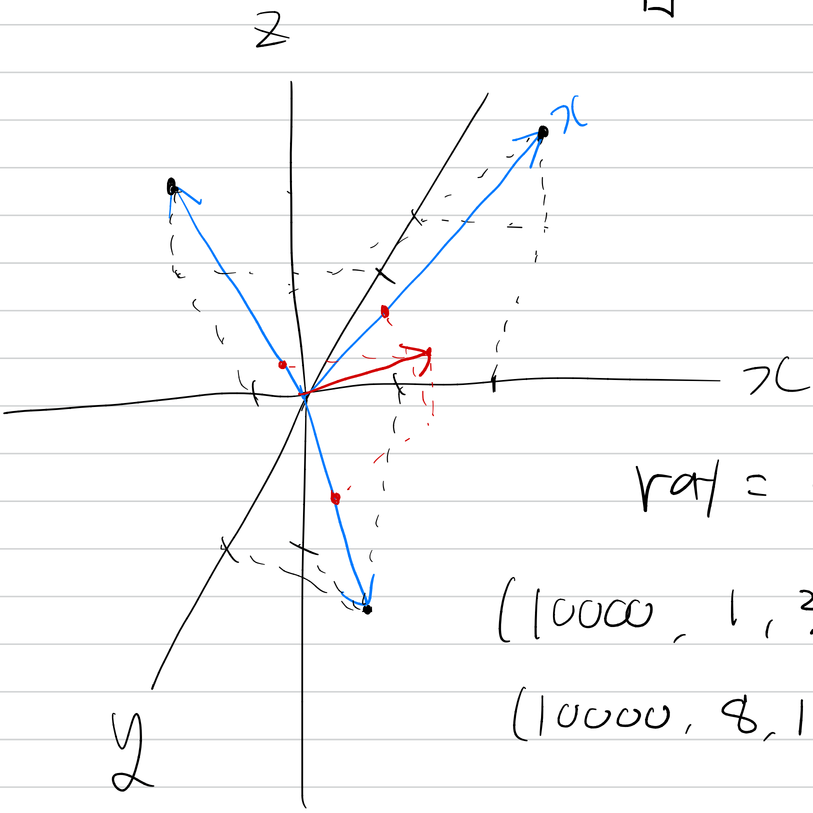
$[[-0.35,\ 0.36,\ -1,\ 0]]$

$[[-0.34,\ 0.36,\ -1,\ 0]]$

⋮

[[ ]]] 100, 100, 1, 3





$$ray = o + td$$

$(10000, 1, 3)$

$(10000, 8, 1)$

$$10000 \left(\begin{matrix} [[x, y, z]] \\ \vdots \end{matrix} \right) \qquad \begin{matrix} [[t_1] \\ [t_2] \\ \vdots \\ [t_8]] \end{matrix}$$

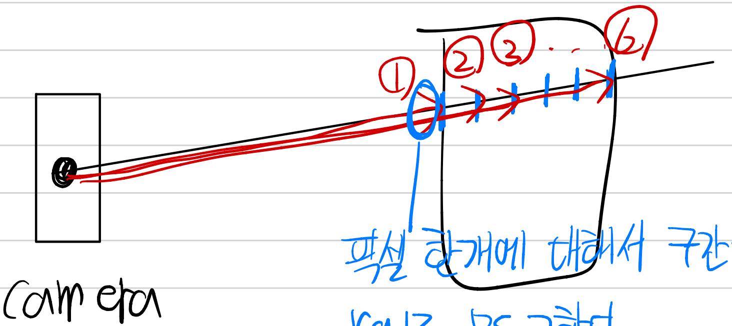
$$\underset{d}{[[x, y, z]]} \qquad \underset{t}{[[t_1],[t_2],[t_3],[t_4],[t_5],[t_6],[t_7],[t_8]]}$$

$(10000, 8, 3)$

$$[[t_1x, t_1y, t_1z], [t_2x, t_2y, t_2z], \dots, [t_8x, t_8y, t_8z]]$$

0,8305	−2,3418	0,0970
0,8569	−2,4161	0,1001

$ray = o + td$



픽셀 한개에 대해서 구간별 샘플링된 점까지의 ray를 모두 구한다.

$(2^0, 2^{4-1}, 4)$

$= (1, 8, 4)$

$[1, \cdots, 8]$

Positional Encoding $(\sin 2^{L-1}, \cos 2^{L-1})$

$0 \sim L-1$

원본데이터(입력 데이터)

증가된 데이터 (sin 10개 cos 10개)

$3 \times (1 + 2 \times 10) = 63$ output

좌표계 원소개수(x,y,z)

21

$3 \times (1 + 2 \times 4) = 27$ output

9

(x', y', z')

좌표 원소 1개당 $2 \times L$ 만큼 증가됨

$2^0 \sim 2^9$ $\qquad 2$

$1 \sim 512$

$\sin \pi p, \cos \pi p, \circ\circ\circ, \sin 512\pi p, \cos 512\pi p$

$[\overbrace{1, 57}^{56}, \ldots, 512]$

$\sin 2^0 \pi p$ $\qquad \sin 2^{L-1} \pi p$

2^8

2^9

2^{10}

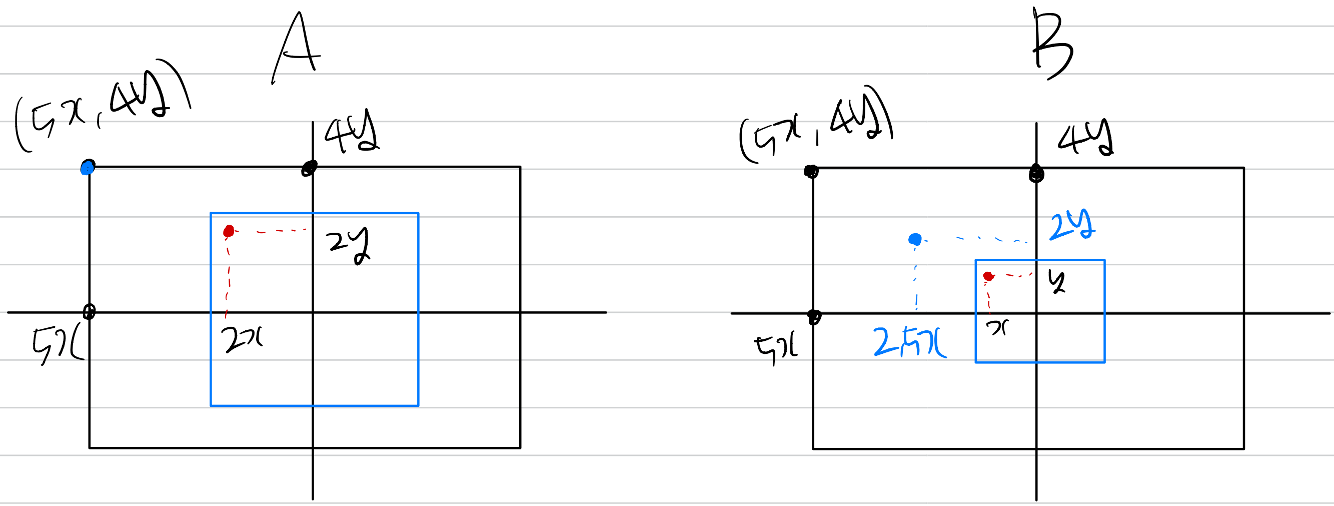
$\sin \pi p$ $\qquad \sin 1024 \pi p$

1 $\qquad 1024$

11

$2^{L-1}, L=4$

초점거리가 짧을 수록 화각이 넓어진다.



픽셀좌표 $(2x, 2y)$

focal length $2L$

픽셀좌표 (x, y)

focal length L

A

$$\left(\frac{2x}{2L}, \frac{2y}{2L}\right)$$

B

$$\left(\frac{x}{L}, \frac{y}{L}\right)$$

	A	B
$L=1$,	(x, y)	, (x, y)
$L=2$,	$\left(\frac{1}{2}x, \frac{1}{2}y\right)$	, $\left(\frac{1}{2}x, \frac{1}{2}y\right)$
$L=0.3$	, $\left(\frac{10}{3}x, \frac{10}{3}y\right)$	, $\left(\frac{10}{3}x, \frac{10}{3}y\right)$

A	B
$(5x, 4y)$, $2L$	$(5x, 4y)$, L

$L=1$, $\left(\frac{5}{2}x, 2y\right)$ $\quad$ $(5x, 4y)$

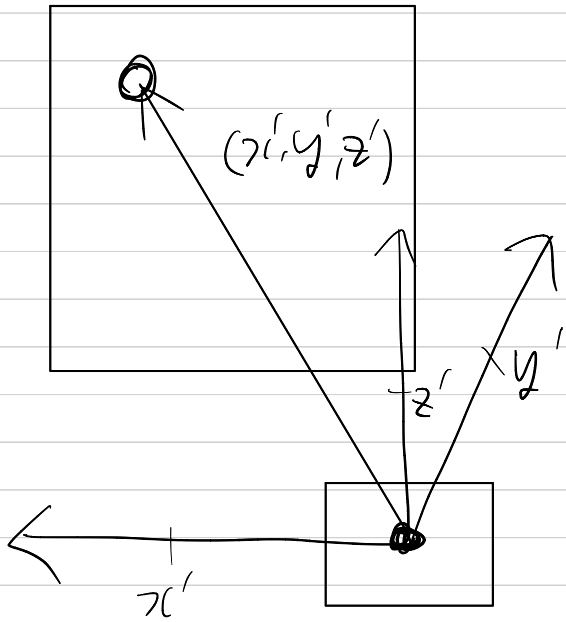
$L=2$, $\left(\frac{5}{4}x, y\right)$ $\quad$ $\left(\frac{5}{2}x, 2y\right)$

A	B
$(5x, 4y)$, $2L$	$\left(\frac{5}{2}x, 2y\right)$, L

$L=1$, $\left(\frac{5}{2}x, 2y\right)$ $\quad$ $\left(\frac{5}{2}x, 2y\right)$

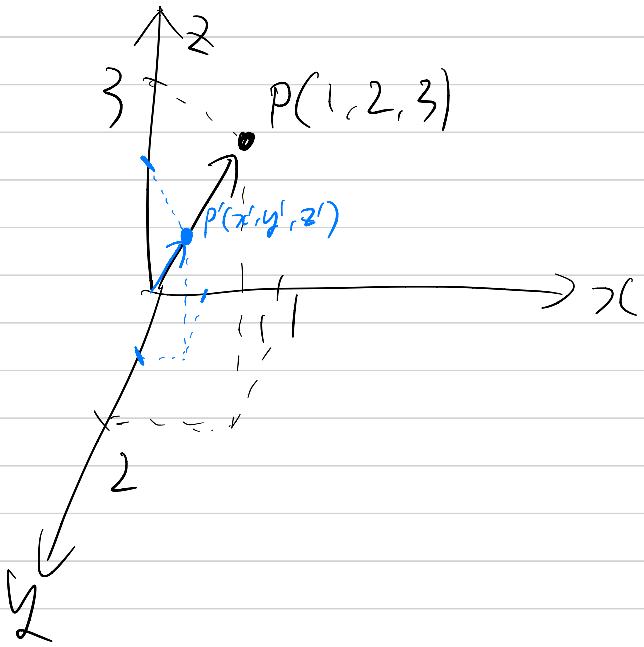
$L=2$, $\left(\frac{5}{4}x, y\right)$ $\quad$ $\left(\frac{5}{4}x, y\right)$

Positional Encoding에 viewing direction 좌표 입력하기 전에
Cartesian Vector인 direction 좌표 (x', y', z')를
normalization하여 각 원소마다 나누어주는 이유는?



$$\frac{x}{x+y+z}, \quad \frac{y}{x+y+z}, \quad \frac{z}{x+y+z}$$

$$\sqrt{x'^2 + y'^2 + z'^2} = \text{norm}$$



$$\sqrt{1+4+9} = \sqrt{14}$$

$$P'\left(\frac{\sqrt{14}}{14}, \frac{\sqrt{14}}{7}, \frac{3\sqrt{14}}{14}\right)$$

$$\frac{\sqrt{14}}{14} \doteqdot 0{,}2672$$

numpy, torch cumprod

[[1, 2, 3], [4, 5, 6]]

→ axis = 0, 행 방향

1	4
2	5
3	6

↓ axis = 1
열 방향

cumprod, axis = 0

[[1, 2, 3], [4, 10, 18]]

cumprod, axis = 1

[[1, 2, 6], [4, 20, 120]]

$$\overset{0}{[x \quad y \quad z]} \qquad \overset{1}{[x \quad y \quad z]}$$

$$\big((a_1, b_1, c_1), (a_2, b_2, c_2)\big)$$

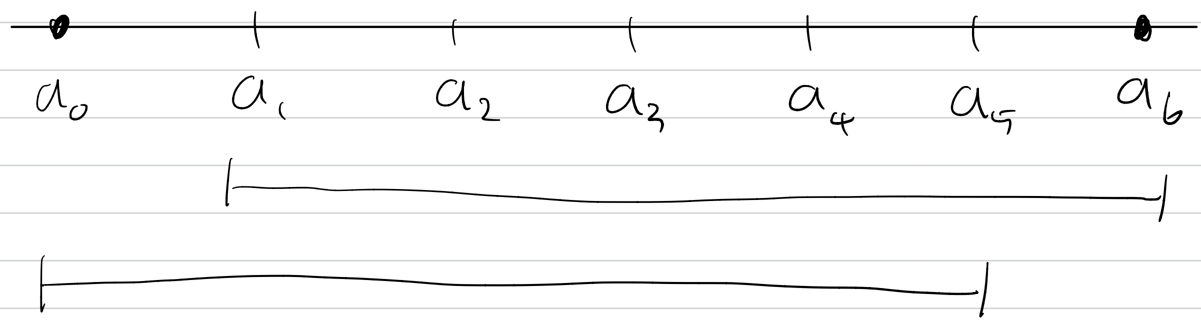
	0	1
x	a_1	a_2
y	b_1	b_2
z	c_1	c_2

→ 열, axis=0

↓ 행, axis=1

axis = -1, → axis = 1

Volume Rendering



$$[a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6]$$

$$[a_0 \quad a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5]$$

$$\underline{a_1 - a_0},\ \underline{a_2 - a_1},\ \underline{a_3 - a_2},\ \underline{a_4 - a_3},\ \underline{a_5 - a_4},\ \underline{a_6 - a_5}$$

$$\left[\left[[0, 1, 2, 3], [4, 5, 6, 7] \right], \left[[\quad], [\quad] \right] \right]$$

$(10000, 1)$

$(10000, 8)$

$$\left[[a], [b] \ldots [\ ] \right] \qquad \left[[\alpha_1, \alpha_2, \cdots, \alpha_8], [\beta_1, \beta_2, \cdots, \beta_8] \cdots \right]$$

$[a\alpha_1, a\alpha_2, \cdots, a\alpha_8]$

$$alpha = 1 - \exp(-\sigma_i \times \delta_i)$$

$$1.0 - (1.0 - \exp(-\sigma_i \delta_i)) + 0.0000000001$$

$$= \underbrace{\exp(-\sigma_i \delta_i)}_{= \alpha = density = \sigma(r(t))} + 1e-10$$

$$C(r) = \int_{t_n}^{t_f} T(t) \times \sigma(r(t)) \times c(r(t), d)\, dt$$

$\downarrow$ 수식 변환

$$\hat{C}(r) = \sum_{i=1}^{N} T_i (1 - \exp(-\sigma_i \delta_i)) c_i$$ $\leftarrow$ Optical Models for Direct Volume Rendering 참고

$$T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right), \quad T(t) = \exp\left(-\int_{t_n}^{t} \sigma(r(s))\, ds\right)$$

$$\delta_i = t_{i+1} - t_i$$

$$t_i \sim \mathcal{U}\left[t_n + \frac{i-1}{N}(t_f - t_n),\ t_n + \frac{i}{N}(t_f - t_n)\right]$$

$\exp(-\sigma(r(t)) \to$ cumprod, dim=-1, 열 방향 (한 행끼리)

$$[[\sigma(t_n)], [\sigma(t_1)], [\sigma(t_2)], \ldots [\sigma(t_f)]]$$

dim=-1,