

3D 到 2D 的投影过程 (3D point $\rightarrow$ 2D pixel)

1. 世界坐标系到相机坐标系

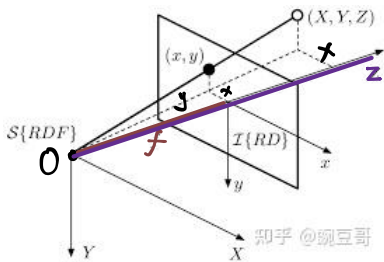
$$P_w = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix}$$

$$P_c = P_w R + t \quad R, t \text{ 外参 } \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} R + t$$

2. 相机到成像平面 (透视投影)

小孔成像



$$\frac{x_n}{f} = \frac{X_c}{Z_c} \quad \frac{y_n}{f} = \frac{Y_c}{Z_c}$$

焦距归一化 $f=1$

$$x_n = \frac{X_c}{Z_c} \quad y_n = \frac{Y_c}{Z_c}$$

相机坐标系: $P_c = (X_c, Y_c, Z_c)$

归一化成像平面:
$$P_n = \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{X_c}{Z_c} \\ \frac{Y_c}{Z_c} \\ 1 \end{bmatrix}$$

3. 成像平面到像素平面

实际图像在像平面上的坐标:

$$p = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

映射:

$$p = K p_n$$

K 内参

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{x_c}{z_c} \\ \frac{y_c}{z_c} \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} f_x \frac{x_c}{z_c} + c_x \\ f_y \frac{y_c}{z_c} + c_y \\ 1 \end{bmatrix}$$

齐次坐标写法:

直接像素坐标与相机坐标映射:

$$T_{11}$$

$$T_{12}$$

$$S \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$$

S 尺度因子

K 内参

$$S \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$$

$$= \begin{bmatrix} f_x x_c + c_x z_c \\ f_y y_c + c_y z_c \\ z_c \end{bmatrix}$$

S 就是 z_c ，由深度 z_c 引起的透视尺度因子，
相机越远， $z_c \uparrow$ ，即透视效应

$$u = \frac{f_x x_c + c_x z_c}{z_c} = f_x \frac{x_c}{z_c} + c_x$$

$$v = \frac{f_y y_c + c_y z_c}{z_c} = f_y \frac{y_c}{z_c} + c_y$$

联系 3D Gaussian 投影

$$(3D \text{ Gaussian } (\mu, \Sigma) \rightarrow 2D \text{ Gaussian } (\mu', \Sigma'))$$

① 其中 μ 变换与 3D point 投影一样:

$$P_w \rightarrow P_c = R P_w + t \rightarrow p_w = K P_c$$

② Σ 变换 (只需要 R)

1) 世界坐标系 $\rightarrow$ 相机坐标系 ($\Sigma_w \rightarrow \Sigma_c$)

优化 3DGS 的那些参数都是在世界坐标系下

协方差定义: $\Sigma = E[(X - \mu)(X - \mu)^T]$

$$\Sigma_w = E[(X_w - \mu_w)(X_w - \mu_w)^T]$$

$$= Q \Lambda Q^T$$

$$= R_g S_g S_g^T R_g^T$$

$$S_g S_g^T = \Lambda = \begin{bmatrix} s_1^2 & & \\ & s_2^2 & \\ & & s_3^2 \end{bmatrix}$$

旋转: $(X_w - \mu_w) \rightarrow R(X_w - \mu_w)$

$$\Sigma_c = R_c \Sigma_w$$

$$= E[R_c(X_w - \mu_w)(R_c(X_w - \mu_w))^T]$$

$$= R_c \Sigma_w R_c^T$$

$$= R_c R_g \Lambda R_g^T R_c^T$$

$$\text{令 } \Lambda = S_g S_g^T$$

$$R = R_c R_g$$

2) 相机坐标系 $\rightarrow$ 像平面 $= R S_g S_g^T R^T$

由于真实的相机是透视变换, 不能直接用 R/A (否则) 直接投影过去。

不是 $u = X$

$v = Y$

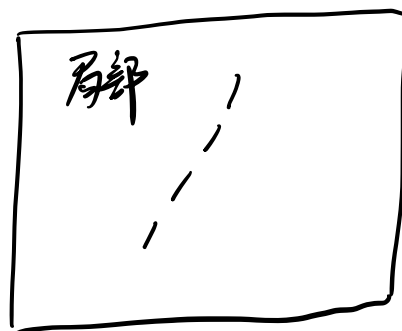
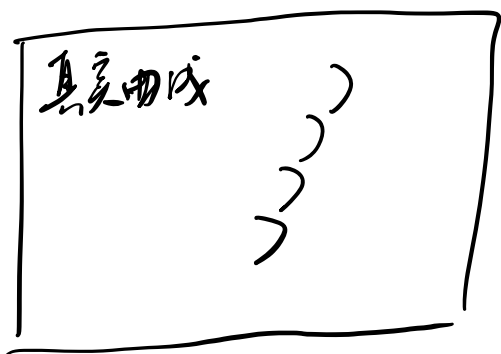
而是 $u = f_x \frac{X}{Z} + c_x$

$v = f_y \frac{Y}{Z} + c_y$

Δ 导致非线性

那怎么办呢?

由于 Gaussian 很小, 虽然整个投影过程是非线性的, 但是 GS 局部是近似线性的



对于 Gaussian, 不是一个点而是一团

$$X = \mu + E \quad (E \text{ 表示扰动})$$

投影到 2D:

$$p = \pi(X)$$

$$\therefore \text{近似 } \Delta p = J \Delta X$$

补充 Σ 的旋转与投影:

旋转 $R_{3 \times 3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

椭圆 $\rightarrow$ 椭圆 $\Sigma_c = R \Sigma_w R^T$

投影 $A_{2 \times 3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$

椭圆 $\rightarrow$ 椭圆 $\Sigma_y = A \Sigma_x A^T$

补充: Jacobian

它表示 3D 空间的小变化, 经过投影后,

在 2D 图像中的变化

3D 空间移动: $\Delta X = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}$

∴ 三维扰动 Σ_c 投影到 2D:

$$\Sigma_{2D} = J \Sigma_c J^T \\ = J R S_g S_g^T R^T J^T$$

优化 $\Sigma \rightarrow$ 优化 R, S

即: $\Sigma_w \xrightarrow{R \Sigma R^T} \Sigma_c \xrightarrow{J \Sigma J^T} \Sigma_{2D}$

投影后: $\Delta p = \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}$

扰动性质: $\Delta p = J \Delta X$

J 根据相机内参: $\begin{bmatrix} \frac{f_x}{z} & 0 & -\frac{f_x x}{z^2} \\ 0 & \frac{f_y}{z} & -\frac{f_y y}{z^2} \end{bmatrix} \Delta X$
 Gaussian 扰动. 相机参数
 计算出基于中间变量 x 方向移动 y

具体 $u = f_x \frac{x}{z} + c_x$

求偏导 $\frac{\partial u}{\partial x} = \frac{f_x}{z}$ 即 x 方向移动一点, 会造成多少像素变化

额外问题 ① 为什么不直接优化 Σ 的 9 个数 $\begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \end{bmatrix}$?

答: 1. 因为 Σ 必须是正定的: 所有特征值 > 0

2. 直接优化增加了复杂度

额外问题 ② 那如何优化 R 呢? 为什么不能直接优化 $R = \begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \end{bmatrix}$?

答: 1. R 需满足 $R R^T = I$, 梯度下降会破坏这些约束

2. 直接优化增加了复杂度

$\Rightarrow$ 用四元数表示 R

四元数: $q = (\underbrace{q_r}_{\text{实}}, \underbrace{q_i, q_j, q_k}_{\text{虚}})$

要求 $\|q\| = 1$, 即 $q_r^2 + q_i^2 + q_j^2 + q_k^2 = 1$

4-1=3 个自由度

四元数表示旋转:

$$R(q) = \begin{bmatrix} 1 - 2(q_j^2 + q_k^2) & 2(q_i q_j - q_r q_k) & 2(q_i q_k + q_r q_j) \\ 2(q_i q_j + q_r q_k) & 1 - 2(q_i^2 + q_k^2) & 2(q_j q_k - q_r q_i) \\ 2(q_i q_k - q_r q_j) & 2(q_j q_k + q_r q_i) & 1 - 2(q_i^2 + q_j^2) \end{bmatrix}$$

天然满足: $R^T R = I$

$\Rightarrow$ 3DGS 优化的参数: m, q, s, α, c