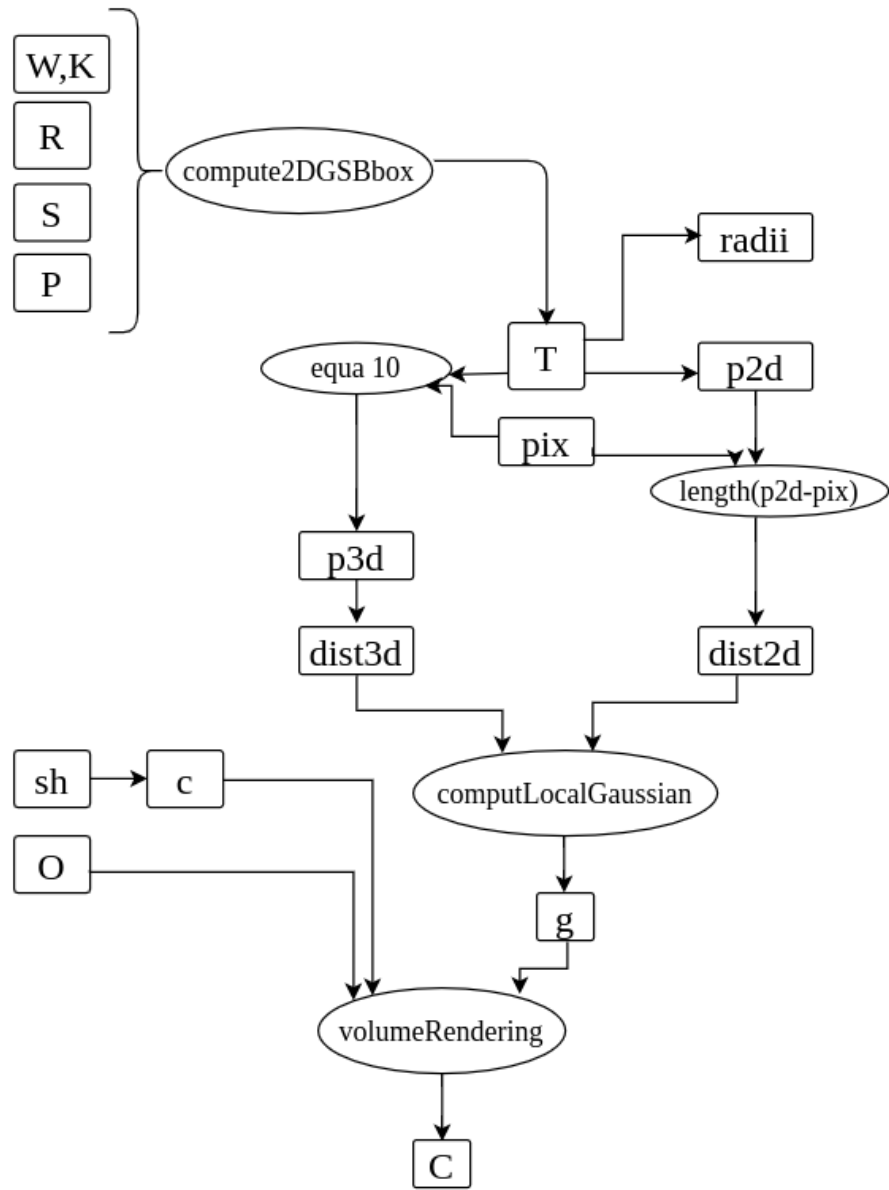


2DGS

计算图如下：



前向

初始化 t_u, t_v :

利用open3d对稀疏点云求法向，得到单位向量 t_n

在训练过程中2D椭圆图元用四元数 q 与两个尺度因子 s_u, s_v 表达，其中 q 代表的旋转矩阵的第三列初始化为 t_n ，前两列通过正交化得到，在后续的优化过程中对传到前两列的梯度计算并传递到四元数与scale上

compute2DGSBBox.forward

2DGS在计算图像上高斯的投影点 `point_image` 和 `radii` 时和3DGS不一样，2DGS不是forward地将高斯投影到图像上，而是通过在图像上划定一个bbox `B_1`，将其映射到高斯局部平面上，得到另一个bbox `B_2`（`B_2`不一定是长方形，但一定是平行四边形），通过约束 `B_2` 的各边到原点的距离为1，计算出 `B_1` 的 `x_1, x_2, y_1, y_2`，并取 `B_1` 的中心为投影点，过程如下：

$$W = W2C, \quad H = \begin{bmatrix} s_u t_u & s_v t_v & 0 & p_k \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad K = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & 0 & c_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$T = (KWH)^T = [t_0 \quad t_1 \quad t_2 \quad t_2]_{3 \times 4}$$

代码中的 `T` 即为公式中的 T ，均为 3×4 的矩阵（省略法向方向的变换）；

由论文， KWH 本身将高斯局部平面点的齐次坐标映射到相机平面上，则 $(KWH)^T$ 将相机平面上的 **平面齐次坐标** 映射到高斯局部平面上。

2DGS 在相机平面通过 x-plane 和 y-plane 确定像素点 (x, y) ，分别将 x-plane 和 y-plane 通过 $(KWH)^T$ 映射到局部平面上，两个面相交的线与局部平面上的交点即为“采样点”

设像素坐标系为 xy 坐标系，高斯局部坐标系为 uv 坐标系，在 xy 坐标系下的 x, y 平面齐次坐标分别为：

$$hx = \begin{bmatrix} -1 \\ 0 \\ 0 \\ x \end{bmatrix}, hy = \begin{bmatrix} 0 \\ -1 \\ 0 \\ y \end{bmatrix}$$

则 T 右乘 hx, hy 可将 xy 下的平面齐次坐标映射到 uv 下，即（上标表示向量第几个元素）：

$$[t_0 \quad t_1 \quad t_2 \quad t_3] h_x = \begin{bmatrix} -t_0^0 + t_3^0 x \\ -t_0^1 + t_3^1 x \\ -t_0^2 + t_3^2 x \end{bmatrix} = h_u$$

应用距离公式：

$$\frac{|h_u^2|}{\sqrt{(h_u^0)^2 + (h_u^1)^2}} = 1$$

注意 aligned axis 的 x-plane 变换到 uv 后不一定 aligned 到 uv axis；因此需要用距离公式做约束，得到一个一元二次方程

两边平方移项可得：

$$(t_0^0)^2 + (t_0^1)^2 - (t_0^2)^2 + x^2[(t_3^0)^2 + (t_3^1)^2 - (t_3^2)^2] - 2x(t_0^0 t_3^0 + t_0^1 t_3^1 - t_0^2 t_3^2) = 0$$

根 x_1, x_2 分别表示了 uv 平面上 bbox 的 u-plane 映射到 xy 平面上 x-plane 的位置，则 `center` 可表示为两根之和/2，也即：

$$\frac{x_1 + x_2}{2} = -\frac{b}{a} = \frac{t_0^0 t_3^0 + t_0^1 t_3^1 - t_0^2 t_3^2}{(t_3^0)^2 + (t_3^1)^2 - (t_3^2)^2}$$

两根之差除以 $2 = \sqrt{b^2 - 4ac} / 2a$ 即为图像上 bbox 的半径，也即 splatting 过程中的 `my_radius`

为什么不直接用 `T.t` 将 $(0, 0, 1, 1)^T$ 变换到相机平面得到中心点？

因为投影变换不是仿射的，空间 gs 椭球/椭圆面投影到图像上不一定是对称的，2dgs 使用 backward 的采样方式实际上是为了规避雅可比近似带来的误差，因此在确定相机平面上 2dgs 的 center 时不能直接用透视变换投影 2dgs 质心，而要用 **bbox 确定中心** 保证采样的对称性

通过以上方法得到 **投影点** 和 **半径** 后，即可使用 3D gaussians splatting 的 tile-based 排序方法

如果距离为 3 (3σ)

则：

$$\frac{|h_u^2|}{\sqrt{(h_u^0)^2 + (h_u^1)^2}} = 3$$

$$9(t_0^0)^2 + 9(t_0^1)^2 - (t_0^2)^2 + x^2[9(t_3^0)^2 + 9(t_3^1)^2 - (t_3^2)^2] - 2x(9t_0^0 t_3^0 + 9t_0^1 t_3^1 - t_0^2 t_3^2) = 0$$

如果有滤波，则需要计算高斯在图像上的投影点：

滤波的前向为 $KWH \Rightarrow p_{2d}$ ，注意 $T_t = KWH$ ，由于 K 最后一行是齐次位，故 T_t 第四行和第三行一样

$$T_t = \begin{bmatrix} T_{00} & T_{01} & T_{02} \\ T_{10} & T_{11} & T_{12} \\ T_{20} & T_{21} & T_{22} \end{bmatrix}$$

$$a = \sigma^2 T_{20}^2 + \sigma^2 T_{21}^2 - T_{22}^2$$

$$b = -2(\sigma^2 T_{00} T_{02} + \sigma^2 T_{01} T_{12} - T_{02} T_{22})$$

$$c = \sigma^2 T_{00}^2 + \sigma^2 T_{01}^2 - T_{02}^2$$

$$ax^2 + bx + c = 0$$

则两根之和

$$p_{2d} \rightarrow x = \frac{(x_1 + x_2)}{2} = -\frac{b}{2a} = \frac{T_{00}T_{02} + T_{10}T_{12} - T_{20}T_{22}}{T_{02}^2 + T_{12}^2 - T_{22}^2}$$

computeLocalGaussian.forward

该函数目的是知道像素点与某个高斯参数，获得该高斯对像素点的权重。该过程有两种方式，第一种发生在**三维空间**，也即在2DGS局部空间上：将像素点变换到uv坐标系，查询高斯权重；第二种发生在**二维空间**，也即在相机平面上：高斯投影点对该像素点的贡献（也即文中的滤波），如果有滤波，则输入需要加入高斯的投影点 `point_image`

输入：`KWH_t`（3x4），`pix_xy`，`point_image`

三维空间：

$$Tt = KWH_{4 \times 3}$$

$$k = -Tt_{0,:} + xTt_{3,:} = \begin{bmatrix} -T_{00} + xT_{30} \\ -T_{01} + xT_{31} \\ -T_{02} + xT_{32} \end{bmatrix}$$

$$l = -Tt_{1,:} + yTt_{3,:} = \begin{bmatrix} -T_{10} + xT_{30} \\ -T_{11} + xT_{31} \\ -T_{12} + xT_{32} \end{bmatrix}$$

$$p = k \times l$$

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} k_2 l_3 - k_3 l_2 \\ k_3 l_1 - k_1 l_3 \\ k_1 l_2 - k_2 l_1 \end{bmatrix}$$

$$d = \left(\frac{p^1}{p^3}\right)^2 + \left(\frac{p^2}{p^3}\right)^2$$

$$g = \exp\left(-\frac{d}{2}\right)$$

二维空间：

令投影点为 p_{2d}

$$\hat{g} = \exp\left(-\frac{(p_{2d} - \begin{bmatrix} x \\ y \end{bmatrix})^2}{2\sigma^2}\right), \quad \sigma = \frac{\sqrt{2}}{2}$$

反向

对于

$$\hat{c} = \sum_{i=1}^N c_i \bar{\alpha}_i T_i \quad \text{where } \bar{\alpha}_i = \alpha_i g_i, T_i = \prod_{j=1}^{i-1} (1 - \bar{\alpha}_j)$$

梯度传递路径如下：

$$\begin{cases} c_i \rightarrow sh_i \\ \alpha_i \rightarrow o_i \\ T_i \rightarrow \begin{cases} \alpha_j \rightarrow o_j \\ g_j \rightarrow R, S \end{cases} \\ g_i \rightarrow R, S \end{cases}$$

则梯度 $\frac{\partial L}{\partial \alpha}, \frac{\partial L}{\partial g}$ 计算如下：

$$\begin{aligned} \frac{\partial \hat{c}}{\partial \bar{\alpha}_i} &= c_i T_i - \frac{\sum_{j=i+1}^N c_j \bar{\alpha}_j T_j}{1 - \bar{\alpha}_i} \\ &= (c_i - \frac{\sum_{j=i+1}^N c_j \bar{\alpha}_j T_j}{T_{i+1}}) T_i \\ &= (c_i - A_i) T_i \end{aligned}$$

其中 A_i 是有递推公式的，因此可以节省计算时间

$$\begin{aligned} A_i &= \frac{\sum_{j=i+1}^N c_j \bar{\alpha}_j T_j}{T_{i+1}} \\ &= c_{i+1} \bar{\alpha}_{i+1} + \frac{\sum_{j=i+2}^N c_j \bar{\alpha}_j T_j}{T_{i+1}} \\ &= c_{i+1} \bar{\alpha}_{i+1} + \frac{\sum_{j=i+2}^N c_j \bar{\alpha}_j T_j}{T_{i+2}} * (1 - \bar{\alpha}_{i+1}) \\ &= c_{i+1} \bar{\alpha}_{i+1} + A_{i+1} (1 - \bar{\alpha}_{i+1}) \end{aligned}$$

则可以从后往前递推 A_i 以及 $\partial L / \partial \bar{\alpha}_i$

从而 $\partial L / \partial \alpha_i = \partial L / \partial \bar{\alpha}_i * g_i, \partial L / \partial g_i = \partial L / \partial \bar{\alpha}_i * \alpha_i$

当xy平面上的投影点 p_g 到像素点 p_I 的距离 \hat{d} 小于 d 时，则使用xy平面的高斯投影点计算高斯权重，此时会产生 `dL_dmean2D` 以及对 KWH_t 的梯度

$$\begin{aligned} \hat{d} &= p_I - p_{2d} \\ g &= \exp\left(-\frac{\hat{d}^2}{2\sigma^2}\right) \end{aligned}$$

从而：

$$\frac{dL}{dp_{2d}} = \begin{bmatrix} -\frac{1}{\sigma^2} L_g g (p_{2d} \cdot x - p_I \cdot x) \\ -\frac{1}{\sigma^2} L_g g (p_{2d} \cdot y - p_I \cdot y) \end{bmatrix}$$

computeLocalGaussian.backward

由 g 到 T

由 `dL_dp2d` 到 T

计算高斯权重：

$$\frac{dL}{dg_i} = \alpha_i \frac{dL}{d\bar{\alpha}_i}$$

如果有滤波，由于投影点是由bbox的中心点得到的，因此存在一条由 `dL_dg` 到 `dL_dp2d` 到 `dL_dT` 的传播路径

投影点计算公式为：

$$\begin{aligned} p_{2d} \rightarrow x &= \frac{(x_1 + x_2)}{2} = -\frac{b_x}{2a} = \frac{T_{00}T_{20} + T_{01}T_{21} - T_{02}T_{22}}{T_{20}^2 + T_{21}^2 - T_{22}^2} \\ p_{2d} \rightarrow y &= \frac{(y_1 + y_2)}{2} = -\frac{b_y}{2a} = \frac{T_{10}T_{20} + T_{11}T_{21} - T_{12}T_{22}}{T_{20}^2 + T_{21}^2 - T_{22}^2} \end{aligned}$$

令

$$\begin{aligned}a &= \sigma^2 T_{20}^2 + \sigma^2 T_{21}^2 - T_{22}^2 \\ b_x &= -2(\sigma^2 T_{00} T_{20} + \sigma^2 T_{01} T_{21} - T_{02} T_{22}) \\ b_y &= -2(\sigma^2 T_{10} T_{20} + \sigma^2 T_{11} T_{21} - T_{12} T_{22})\end{aligned}$$

从而

$$\begin{aligned}\frac{dL}{da} &= L_{p_{2d}}^0 \frac{dp_{2d} \cdot x}{da} + L_{p_{2d}}^1 \frac{dp_{2d} \cdot y}{da} = L_{p_{2d}}^0 \frac{b_x}{2a^2} + L_{p_{2d}}^1 \frac{b_y}{2a^2} \\ \frac{dL}{db_x} &= -L_{p_{2d}}^0 \frac{1}{2a} \\ \frac{dL}{db_y} &= -L_{p_{2d}}^1 \frac{1}{2a}\end{aligned}$$

由于 c 不参与投影点计算，故

$$\begin{aligned}\frac{dL}{dT} &= \frac{dL}{da} \frac{da}{dT} + \frac{dL}{db_x} \frac{db_x}{dT} + \frac{dL}{db_y} \frac{db_y}{dT} \\ &= \frac{dL}{da} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 2\sigma^2 T_{20} & 2\sigma^2 T_{21} & -2T_{22} \end{bmatrix} + \frac{dL}{db_x} \begin{bmatrix} -2\sigma^2 T_{20} & -2\sigma^2 T_{21} & 2T_{22} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -2\sigma^2 T_{00} & -2\sigma^2 T_{01} & 2T_{02} \end{bmatrix} + \frac{dL}{db_y} \begin{bmatrix} 0 & 0 & 0 \\ -2\sigma^2 T_{20} & -2\sigma^2 T_{21} & 2T_{22} \\ 0 & 0 & 0 \\ -2\sigma^2 T_{10} & -2\sigma^2 T_{11} & 2T_{12} \end{bmatrix}\end{aligned}$$

另一条路径是由 `dL_dg` 到 `dL_dp3d` 到 `dL_dT`：

$$\begin{aligned}\frac{dg_i}{dd_i} &= -\frac{g}{2} \\ \frac{dd_i}{dp} &= \begin{bmatrix} 2p_1/p_3^2 \\ 2p_2/p_3^2 \\ -2((p_1^2+p_2^2)/p_3^3) \end{bmatrix} \\ \frac{\partial p}{\partial \boldsymbol{k}} &= \begin{bmatrix} 0 & l_3 & -l_2 \\ -l_3 & 0 & l_1 \\ l_2 & -l_1 & 0 \end{bmatrix} \\ \frac{\partial p}{\partial \boldsymbol{l}} &= \begin{bmatrix} 0 & -k_3 & k_2 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix}\end{aligned}$$

$$\begin{aligned}\frac{dk}{dTt_{0,:}} &= \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} & \frac{dk}{dTt_{3,:}} &= \begin{bmatrix} x & 0 & 0 \\ 0 & x & 0 \\ 0 & 0 & x \end{bmatrix} \\ \frac{dl}{dTt_{1,:}} &= \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} & \frac{dl}{dTt_{3,:}} &= \begin{bmatrix} y & 0 & 0 \\ 0 & y & 0 \\ 0 & 0 & y \end{bmatrix}\end{aligned}$$

从而

$$\begin{aligned}
\frac{dL}{dT_{0,:}} &= \frac{dL}{dd} \frac{dd}{dp} \frac{dp}{dT_{0,:}} \\
&= \frac{dL}{dd} \frac{dd}{dp} \begin{bmatrix} 0 & -T_{12} + yT_{32} & T_{11} - yT_{31} \\ T_{12} - yT_{32} & 0 & -T_{10} + yT_{30} \\ -T_{11} + yT_{31} & T_{10} - yT_{30} & 0 \end{bmatrix} \\
&= \frac{dL}{dd} \frac{dd}{dp} \begin{bmatrix} 0 & l_3 & -l_2 \\ -l_3 & 0 & l_1 \\ l_2 & -l_1 & 0 \end{bmatrix} \\
&= \frac{dL}{dd} \begin{bmatrix} dp_2 l_3 - dp_3 l_2 \\ -dp_1 l_3 + dp_3 l_1 \\ dp_1 l_2 - dp_2 l_1 \end{bmatrix}
\end{aligned}$$

同理

$$\begin{aligned}
\frac{dL}{dT_{1,:}} &= \frac{dL}{dd} \frac{dd}{dp} \frac{dp}{dT_{1,:}} \\
&= \frac{dL}{dd} \frac{dd}{dp} \begin{bmatrix} 0 & -k_3 & k_2 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix} \\
&= \frac{dL}{dd} \begin{bmatrix} -dp_2 k_3 + dp_3 k_2 \\ dp_1 k_3 - dp_3 k_1 \\ -dp_1 k_2 + dp_2 k_1 \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
\frac{dL}{dT_{3,:}} &= \frac{dL}{dd} \frac{dd}{dp} \frac{dp}{dT_{3,:}} \\
&= \frac{dL}{dd} \frac{dd}{dp} \begin{bmatrix} 0 & -x(-T_{12} + yT_{32}) + y(-T_{02} + xT_{32}) & x(-T_{11} + yT_{31}) - y(-T_{01} + yT_{31}) \\ x(-T_{12} + yT_{32}) - y(-T_{02} + xT_{32}) & 0 & -x(-T_{10} + yT_{30}) + y(-T_{00} + xT_{30}) \\ -x(-T_{11} + yT_{31}) + y(-T_{01} + xT_{31}) & x(-T_{10} + yT_{30}) - y(-T_{00} + xT_{30}) & 0 \end{bmatrix} \\
&= \frac{dL}{dd} \frac{dd}{dp} \begin{bmatrix} 0 & -xl_3 + yk_3 & xl_2 - yk_2 \\ xl_3 - yk_3 & 0 & -xl_1 + yk_1 \\ -xl_2 + yk_2 & xl_1 - yk_1 & 0 \end{bmatrix} \\
&= \frac{dL}{dd} \begin{bmatrix} dp_2(xl_3 - yk_3) + dp_3(-xl_2 + yk_2) \\ dp_1(-xl_3 + yk_3) + dp_3(xl_1 - yk_1) \\ dp_1(xl_2 - yk_2) + dp_2(-xl_1 + yk_1) \end{bmatrix}
\end{aligned}$$

compute2DGSBBox.backward

输入 `dL_KWH_t` (3×4)

由 T 到 R, S, p

前向为: $R, S, W, K \Rightarrow KWH_t$

$$\begin{aligned}
K &= \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}^T \\
H &= SR \\
Y &= HW_R^T \\
p_v &= p_w^T W_R^T + W_T^T \\
M &= \begin{bmatrix} Y_{0,:} & 0 \\ Y_{1,:} & 0 \\ p_v^T & 1 \end{bmatrix} \\
T &= MK
\end{aligned}$$

展开可得：

$$Y = \begin{bmatrix} H_{00}W_{R00} + H_{01}W_{R10} + H_{02}W_{R20} & H_{00}W_{R01} + H_{01}W_{R11} + H_{02}W_{R21} & H_{00}W_{R02} + H_{01}W_{R12} + H_{02}W_{R22} \\ H_{10}W_{R00} + H_{11}W_{R10} + H_{12}W_{R20} & H_{10}W_{R01} + H_{11}W_{R11} + H_{12}W_{R21} & H_{10}W_{R02} + H_{11}W_{R12} + H_{12}W_{R22} \\ H_{20}W_{R00} + H_{21}W_{R10} + H_{22}W_{R20} & H_{20}W_{R01} + H_{21}W_{R11} + H_{22}W_{R21} & H_{20}W_{R02} + H_{21}W_{R12} + H_{22}W_{R22} \end{bmatrix}$$

$$T = \begin{bmatrix} f_x M_{00} + c_x M_{02} & f_y M_{01} + c_y M_{02} & M_{02} & M_{02} \\ f_x M_{10} + c_x M_{12} & f_y M_{11} + c_y M_{12} & M_{12} & M_{12} \\ f_x M_{20} + c_x M_{22} & f_y M_{21} + c_y M_{22} & M_{22} & M_{22} \end{bmatrix}$$

则：

$$\frac{dL}{dM} = \begin{bmatrix} f_x L_{KWH_t}^{00} & f_y L_{KWH_t}^{01} & c_x L_{KWH_t}^{00} + c_y L_{KWH_t}^{01} + L_{KWH_t}^{02} + L_{KWH_t}^{03} & 0 \\ f_x L_{KWH_t}^{10} & f_y L_{KWH_t}^{11} & c_x L_{KWH_t}^{10} + c_y L_{KWH_t}^{11} + L_{KWH_t}^{12} + L_{KWH_t}^{13} & 0 \\ f_x L_{KWH_t}^{20} & f_y L_{KWH_t}^{21} & c_x L_{KWH_t}^{20} + c_y L_{KWH_t}^{21} + L_{KWH_t}^{22} + L_{KWH_t}^{23} & 0 \end{bmatrix}$$

由 M 到 Y 和 p_v ：

$$\frac{dL}{dp_w} = [L_M^{20} \quad L_M^{21} \quad L_M^{22}]^T$$

$$\frac{dL}{dY} = \begin{bmatrix} f_x L_{KWH_t}^{00} & f_y L_{KWH_t}^{01} & c_x L_{KWH_t}^{00} + c_y L_{KWH_t}^{01} + L_{KWH_t}^{02} + L_{KWH_t}^{03} \\ f_x L_{KWH_t}^{10} & f_y L_{KWH_t}^{11} & c_x L_{KWH_t}^{10} + c_y L_{KWH_t}^{11} + L_{KWH_t}^{12} + L_{KWH_t}^{13} \\ 0 & 0 & 0 \end{bmatrix}$$

由 Y 到 H （注意这里 W 矩阵为 $W2C.T$ ）

$$\frac{dL}{dH} = \begin{bmatrix} L_Y^{00}W_{R00} + L_Y^{01}W_{R01} + L_Y^{02}W_{R02} & L_Y^{00}W_{R10} + L_Y^{01}W_{R11} + L_Y^{02}W_{R12} & L_Y^{00}W_{R20} + L_Y^{01}W_{R21} + L_Y^{02}W_{R22} \\ L_Y^{10}W_{R00} + L_Y^{11}W_{R01} + L_Y^{12}W_{R02} & L_Y^{10}W_{R10} + L_Y^{11}W_{R11} + L_Y^{12}W_{R12} & L_Y^{10}W_{R20} + L_Y^{11}W_{R21} + L_Y^{12}W_{R22} \\ 0 & 0 & 0 \end{bmatrix}$$

由 H 到 R 和 S

注意这里 R 每一行是一个向量/轴

$$H = SR \quad R = \begin{bmatrix} R_{00} & R_{01} & R_{02} \\ R_{10} & R_{11} & R_{12} \\ R_{20} & R_{21} & R_{22} \end{bmatrix} \quad S = \begin{bmatrix} s_0 & 0 & 0 \\ 0 & s_1 & 0 \\ 0 & 0 & s_2 \end{bmatrix}$$

$$H = \begin{bmatrix} s_0 R_{00} & s_0 R_{01} & s_0 R_{02} \\ s_1 R_{10} & s_1 R_{11} & s_1 R_{12} \\ s_2 R_{20} & s_2 R_{21} & s_2 R_{22} \end{bmatrix}$$

则

$$\frac{dL}{dS} = \begin{bmatrix} L_H^{00}R_{00} + L_H^{01}R_{01} + L_H^{02}R_{02} \\ L_H^{10}R_{10} + L_H^{11}R_{11} + L_H^{12}R_{12} \\ 0 \end{bmatrix}$$

四元数到旋转矩阵如下（每一行是一个向量，第三行是没用的因为 dL/dH 第三行为0）：

$$q = [r, x, y, z]$$

$$R = \begin{bmatrix} 1 - 2(y^2 + z^2) & 2(xy + rz) & 2(xz - ry) \\ 2(xy - rz) & 1 - 2(x^2 + z^2) & 2(yz + rx) \\ 2(xz + ry) & 2(yz - rx) & 1 - 2(x^2 + y^2) \end{bmatrix}$$

从而：

$$\frac{dL}{dq} = \begin{bmatrix} 2z(L_R^{01} - L_R^{10}) - 2yL_R^{02} + 2xL_R^{12} \\ 2y(L_R^{01} + L_R^{10}) + 2zL_R^{02} - 4xL_R^{11} + 2rL_R^{12} \\ -4yL_R^{00} + 2x(L_R^{01} + L_R^{10}) - 2rL_R^{02} + 2zL_R^{12} \\ -4z(L_R^{00} + L_R^{11}) + 2r(L_R^{01} - L_R^{10}) + 2xL_R^{02} + 2yL_R^{12} \end{bmatrix}$$

Depth Distortion

Normal Consistency