

2DGS中的参数化

初识 2DGS

2DGS 是一份对 3DGS 的相当好的改进工作，其实从以往的工作中也能找到发展的脉络，即从 EWA Splatting 到 Perspective Accurate Splatting。

2DGS 最吸引人的无疑是其高质量的 3D重建结果，其引入 2D Gaussian 的表征使得光线与Primitive求交具有确切解，保证了深度信息的多视角一致性，并且还天然地引入法线，不用再对其进行单独的建模。

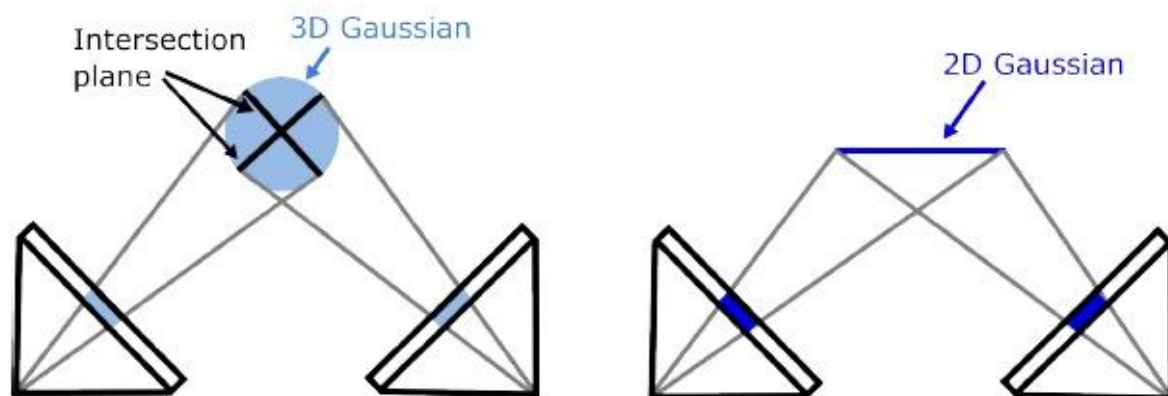


Fig. 2. Comparison of 3DGS and 2DGS. 3DGS utilizes different intersection planes for value evaluation when viewing from different viewpoints, resulting in inconsistency. Our 2DGS provides multi-view consistent value evaluations.

知乎 @ 景页

2DGS 最后的Mesh是通过 TSDF 这种深度法线融合的方式出的，它并没有去构建一个全局的几何，例如 SDF，因此个人认为它并不是一个严格意义上的 Geometry-Aware Reconstruction 的工作。尽管如此，其在场景上的重建结果也足够让人满意了。

2DGS中的变换

另一点值得注意的是，2DGS 解决了 3DGS 中由于仿射变换带来的近似误差，即将世界坐标系中的 3D Gaussian 近似为成像平面上的 2D Gaussian 过程中所引入的误差，这一误差在越远离屏幕中心时会越来越明显，最近的 GS++: Error Analyzing and Optimal Gaussian Splatting 分析了这一误差。

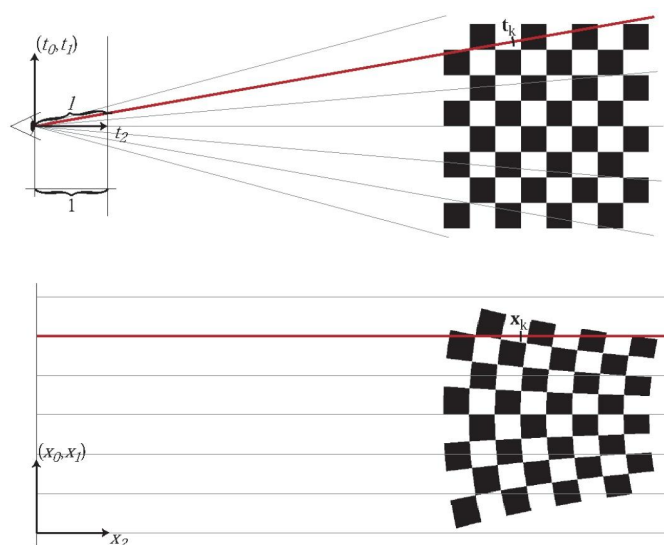


Fig. 8. Transforming the volume from camera to ray space. Top: camera space. Bottom: ray space.

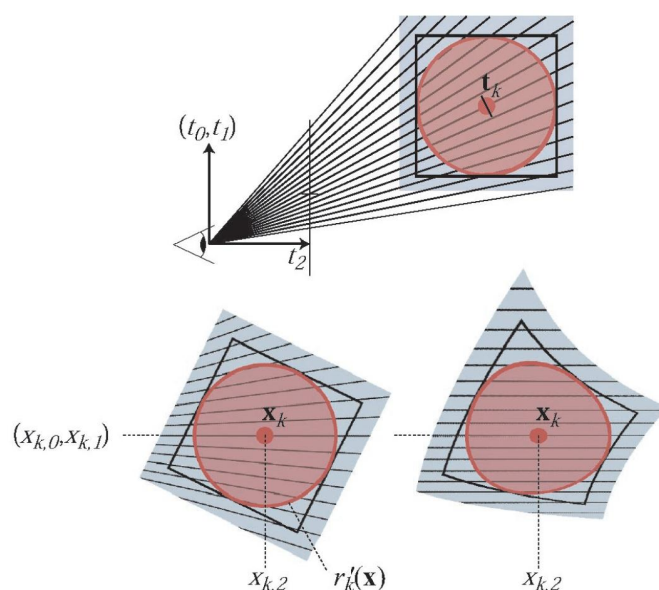


Fig. 9. Mapping a reconstruction kernel from camera to ray space. Top: camera space. Bottom: ray space. Left: local affine mapping. Right: exact mapping.

2DGS 采用了 Perspective Accurate Splatting 一样的策略来避免上述误差，具体的做法在文章的 4.2 章节 Splatting 中提及。补充材料中提到的效果也非常令人满意。

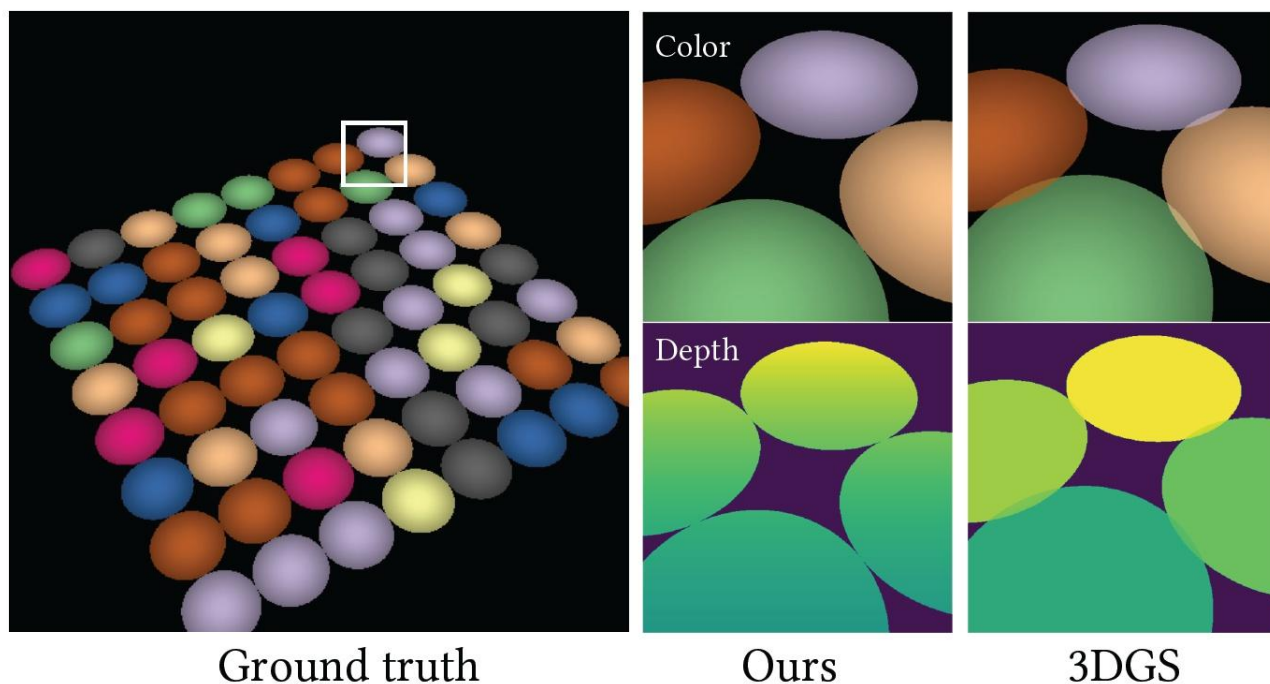


Fig. 7. Affine-approximation adopted in [Zwicker et al. 2001b][Kerbl et al. 2023] causes perspective distortion and inaccurate depth. 知乎 @ 景页

其中 2DGS 为了避免对矩阵的求逆操作，作者采用了 **Ray-splat Intersection** 的方式进行 UV 的计算，其中涉及到参数化的过程，这也是这篇文章中最令人费解的地方。

在参数化过程中，给定一个像素 (x, y) ，作者认为该像素对应的光线是两个平面（x-plane 和 y-plane）的交点，其中 x-plane 的齐次表示为 $\mathbf{h}_x = (-1, 0, 0, x)$ ，y-plane 的齐次表示为 $\mathbf{h}_y = (0, -1, 0, y)$ 。为什么可以这么表示呢？为什么是 -1 而不是 1 呢？这部分文章给了两个引用，而再具体一点是文章 GPU-Based Ray-Casting of Quadratic Surfaces (Quadratic) 的 Ray-Quadric Intersection 以及书 Geometric Algebra for Computer Graphics 中的 Homogeneous coordinates 章节。

10.7	Homogeneous coordinates	184
10.7.1	Introduction	184
10.7.2	Representing 2D lines in 3D homogeneous space	186
10.7.3	Intersection of two lines in \mathbb{R}^2	192
10.7.4	Representing 3D lines in 4D homogeneous space	195
10.7.5	Representing lines and planes in 4D homogeneous space	196

其中文章 Quadratic 提到的 Ray-Quadric Intersection 还是通过求逆的方式求解的，和2DGS中的公式7没有本质区别。而书中这一章节似乎和原文没有直接推到的关系（也有可能我没有找对或者理解透），其实关系更大的应该是Perspective Accurate Splatting这篇文章。同时文中的 $\mathbf{h}_u, \mathbf{h}_y, \mathbf{h}_u, \mathbf{h}_v$ 似乎存在转置符号缺失。

解析 2DGS 中的参数化

为了更加直观地理解参数化这部分，这里我给出了自己的推导过程，如果各位有更好更直观地推导或者能够指出作者引用指出，欢迎指明。

由于作者使用了 u, v 作为 2D 高斯中的坐标，因此，作者在文章中使用了 x, y 作为成像空间的坐标。文中由于基本不涉及三维欧式空间，因此不大会造成符号的混淆。这里会提到三维欧式空间，因此要注意一下，不要和文章中的符号混淆。

以 x-plane 为例，在欧式空间中我们认为只要满足 $x = x_0$ 的点就处在同一 x-plane 上，这些点的坐标可以表示为 (x_0, y, z) ，其中 y, z 为任意值。在齐次坐标系中这些点则表示为 $(x_0, y, z, 1)$ ，同时我们发现齐次坐标系中存在一个巧妙的关系：

$$(-1, 0, 0, x_0) \cdot (x_0, y, z, 1)^\top = -x_0 + x_0 = 0, \quad \forall y, z \in \mathbb{R}$$

因此，在齐次坐标系中 $x = x_0$ 的 x-plane 可表示为

$\mathbf{h}_x = (-1, 0, 0, x_0)$ ，同理可以得到 $y = y_0$ 的 y-plane 可表示为 $\mathbf{h}_y = (0, -1, 0, y_0)$ 。诚然，像素所在的空间应该是屏幕空间 (screen space)，而不是欧式空间，但这一结论在屏幕空间中依旧适用。

那么回到文章的描述中，给定像素 (x_0, y_0) ，x-plane 的齐次表示为 $\mathbf{h}_x = (-1, 0, 0, x_0)$ （为了更好的阐述，我们添加了0作为后缀标识），y-plane 的齐次表示为 $\mathbf{h}_y = (0, -1, 0, y_0)$ ，我们就可以得到两个平面的交线，即像素对应的光线 \mathbf{r} 。而光线 \mathbf{r} 上的点需要同时落在 x-plane 和 y-plane 上，即光线 \mathbf{r} 上的点 $(x, y, z) \in \mathbf{r}$ 满足：

$$(-1, 0, 0, x_0) \cdot (x, y, z, 1)^\top = (0, -1, 0, y_0) \cdot (x, y, z, 1)^\top = 0$$

其实这在欧式坐标中的表现就是 $x = x_0, y = y_0$ ，至于为什么要写成这个样子，则是为了进行后续的参数化计算从而得到文中的公式 9：

$$\mathbf{h}_u \cdot (u, v, 1, 1)^\top = \mathbf{h}_v \cdot (u, v, 1, 1)^\top = 0$$

从而实现 **Ray-splat Intersection** 求得光线与 2D Gaussian 的交点及 UV。

可以看到文中的公式 9 和我们上面推到的光线 \mathbf{r} 的齐次表示在形式上已非常相近了。其实我们通过非常简单的方式能够将上面两条公式进行转换。首先是，根据相机姿态矩阵 \mathbf{T} 和投影矩阵 \mathbf{P} 对世界空间 (world space) 中的目标点（即2D高斯的中心点） (x_w, y_w, z_w) 进行变换，而文章里将这两个矩阵合并称为 \mathbf{W}

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{PM} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \mathbf{W} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

而文章的公式4和5又提供了2D高斯的切平面上定义的 (u, v) 到世界空间 (x_w, y_w, z_w) 的参数化过程，这里写作：

$$\begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} u \\ v \\ 1 \\ 1 \end{bmatrix}$$

结合上面两个式子，2D高斯的切平面上定义的 (u, v) 到屏幕空间 (x, y, z) 的参数化过程则为：

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{W} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \mathbf{WH} \begin{bmatrix} u \\ v \\ 1 \\ 1 \end{bmatrix}$$

我们将其代入

$$(-1, 0, 0, x_0) \cdot (x, y, z, 1)^\top = (0, -1, 0, y_0) \cdot (x, y, z, 1)^\top = 0$$

中则得到：

$$\mathbf{h}_x^\top \mathbf{WH} \begin{bmatrix} u \\ v \\ 1 \\ 1 \end{bmatrix} = \mathbf{h}_y^\top \mathbf{WH} \begin{bmatrix} u \\ v \\ 1 \\ 1 \end{bmatrix} = 0$$

这里令 $\mathbf{h}_x = (-1, 0, 0, x_0)^\top$, $\mathbf{h}_y = (0, -1, 0, y_0)^\top$,

转置一下就得到了:

$$\begin{bmatrix} u & v & 1 & 1 \end{bmatrix} (\mathbf{WH})^\top \mathbf{h}_x = \begin{bmatrix} u & v & 1 & 1 \end{bmatrix} (\mathbf{WH})^\top \mathbf{h}_y = 0$$

以上, 我们通过简单和直观地推导就得到了上述结果。



图穷币见 我关注的人

...

Hi, 感谢校友给出更详细解释, 不过我们的写法应该没有错😅。(1) 公式 9 我们实际上使用的是两个列向量的点乘, 所以应该是与你提及的这种写法一致的, $\mathbf{h}_u^T [u, v, 1, 1] = 0$ 。(2) 你那个 x-plane 的解释也是对的👍👍👍 我们是通过 平面方程 $Ax + By + Cz + D = (A, B, C, D) \cdot (x, y, z, 1) = 0$ 导出的🤖。(3) 不过确实有一个地方有typo, 例如, 由于PAS 使用 3d 的齐次坐标, 为了兼容相机 translation, 我们使用的是 4d 齐次坐标, 造成一些错误, 公式7有一个x 应该参数化为 $(xz, yz, z, 1)$ 而不是 (xz, yz, z, z) 。😅

04-12 · 上海

❤️ 2



景页 作者

...

本尊来了。确实, 公式7中两个z比较迷惑, 感谢解答。顺便一问打算什么时候开源😂



04-12 · 广东

🗨️ 回复 ❤️ 喜欢



景页 作者

...

还有就是不知道公式8中算不算typo呢, 因为 \mathbf{h}_u 和 \mathbf{h}_x 都算行向量的话, 这个矩阵乘法似乎是不成立的。

04-12 · 广东

🗨️ 回复 ❤️ 喜欢



图穷币见 我关注的人 ▶ 景页

...

\mathbf{h}_u 和 \mathbf{h}_x 都是列向量哈。正文里应该漏写了 T 🤔🤔

04-12 · 上海

🗨️ 回复 ❤️ 喜欢



景页 作者 ▶ 图穷币见

...

是捏😂我也觉得



图穷币见 我关注的人 ▶ 图穷币见

...

$hx = (-1, 0, 0, x)^T$, $hy = (0, -1, 0, y)^T$ 🤔

04-12 · 上海

● 回复 ❤️ 喜欢



毛毛

...

可以详细解释一些公式为什么是 $(xz, yz, z, 1)$ 吗，按照上面推导不应该是投影坐标 $(x, y, z, 1)$ 吗 🤔

04-23 · 陕西

● 回复 ❤️ 喜欢



图穷币见 我关注的人 ▶ 毛毛

...

表示成 $(x, y, z, 1)$ 来表示一个3d点也是可以的，用 $(xz, yz, z, 1)$ 方便表示 x, y 为像素点 z 为depth。

04-27 · 上海

● 回复 ❤️ 喜欢