交互平面揭示三维线映射

Zeran Ke*、Bin Tan*、Gui-Song Xia、Yujun Shen、Nan Xue†

摘要—基于多视角 RGB 图像的三维线映射为场景提供了一种紧凑且结构化的视觉表示。我们从物理和拓扑视角研究这一问题：三维直线最自然地表现为有限三维平面片的边缘。我们提出 LiP-Map，一种线—平面联合优化框架，显式建模可学习的线基元和平面基元。这种耦合在保持高效性的同时，实现了准确且细致的三维线映射（通常每个场景仅需 3 至 5 分钟即可完成重建）。LiP-Map 首次将平面拓扑融入三维线映射，并非施加成对共面约束，而是显式构建平面基元与线基元之间的交互，从而为人工环境中的结构化重建提供了一条有原则的路径。在来自 ScanNetV2、ScanNet++、Hypersim、7Scenes 和 Tanks&Temple 的 100 多个场景上，LiP-Map 在准确度和完整性方面均优于最先进方法。除线映射质量外，LiP-Map 还显著推进了线辅助视觉定位，在 7Scenes 上取得了优异性能。我们的代码已发布于 https://github.com/calmke/LiPMAP，以支持可复现研究。

索引词—三维 线映射、三维平面基元、交互平面、结构化视觉几何、相机重定位

arXiv:2602.01296v1 [cs.CV] 1 Feb 2026

1 引言

利用简单基元 [1], [2], [3], [4], [5], [6], [7], [8] 从多视图图像中进行结构化三维重建颇具吸引力，因为这与人类直观地从点、线、曲线和平面等基本元素构建复杂三维几何结构的方式相似，并借助几何组合的简约性。结构化三维场景表示具有符号性质，可以说能够增强空间智能，从而更高效地理解我们的三维世界。然而，这一观点颇具争议，因为许多业内人士仍不认同，认为当前方法的重建质量有限。本文聚焦于两种关键基元，矩形平面和线段，旨在为人工环境实现准确、完整且细致的三**维线映射**[1] 。

三维线映射问题已通过双视图匹配 [9], [10] 和对二维线段进行多视图跟踪得到了广泛研究，其流程与三维点建图类似。然而，由于线段与点之间存在根本差异，建立可靠的线对应关系仍然具有挑战性。近期的一种方法 LIMAP [1], 尝试通过考虑排名前 K 的线匹配，而非单一匹配，来缓解这一问题，从而提高找到正确对应关系的概率，同时剔除错误对应关系。尽管 LIMAP [1] 的表现显著优于 Line3D++ [3], 但要成功进行线三角化，它仍然严重依赖密集的视点。其他研究 [2], [4] 旨在利用神经隐式场绕过二维线匹配，展现了无需匹配的方案的潜力，但由于依赖神经场，这些方法的优化过程较为缓慢。

要根据多视图中的二维直线检测结果构建可靠的三维线地图，需要两个关键要素：稳健的跨视图对应关系，以及对重建三维直线进行准确的空间定位。一种常见策略是利用基于已知相机位姿的多视图输入生成的现有表面重建，并使用深度图将检测到的二维直线投影到表面上。然而，正如图 1 所示，这种方法在实际应用中往往会失败。一方面，视角相关视差可能导致同一条图像直线对应于不同深度处的多条三维线段。另一方面，深度图中的不一致可能引入虚假的结构线。由于真实的三维直线与结构表面边缘本质上不可分割，因此必须联合恢复表面和边缘。为此，我们提出了一种新的三维线映射方法，明确建模平面与线段之间的几何关系及其相互作用。

平面表示提供了一个很有前景的机会。PlanarSplatting [8] 通过优化一组可学习平面基元来重建结构化场景表面。这种表述很有吸引力，因为平面边界天然由线构成，从而揭示了表面与三维线结构之间直接的几何与拓扑联系。然而，PlanarSplatting仅面向表面重建而设计，并未将多视图二维线检测结果与所学基元显式关联。因此，我们探索如何将二维线段与三维平面边缘关联起来，并利用这些关联重建可靠、一致的三维线图。为此，我们提出 LiP-Map，它显式利用平面与线之间的交互来恢复可靠且一致的三维线图。该名称代表用于三维线映射的线—平面联合优化。

我们的方法利用线段与三维平面基元边缘之间的几何协同。通过将任意现有的二维线段检测器 [11], [12],

（*：贡献相同。 †：通讯作者）

- *Z.-R. Ke* 就职于武汉大学计算机科学学院，中国武汉 *430072*。他目前在蚂蚁集团担任研究实习生。
- *Gui-Song Xia* 就职于武汉大学计算机科学学院和人工智能学院，中国武汉 *430072*。
- *B. Tan*、*Y. Shen* 和 *N. Xue* 就职于蚂蚁集团，中国杭州 *310000*。

1. 这一术语借鉴自 LIMAP [1], , 这项任务也称为三维直线重建。

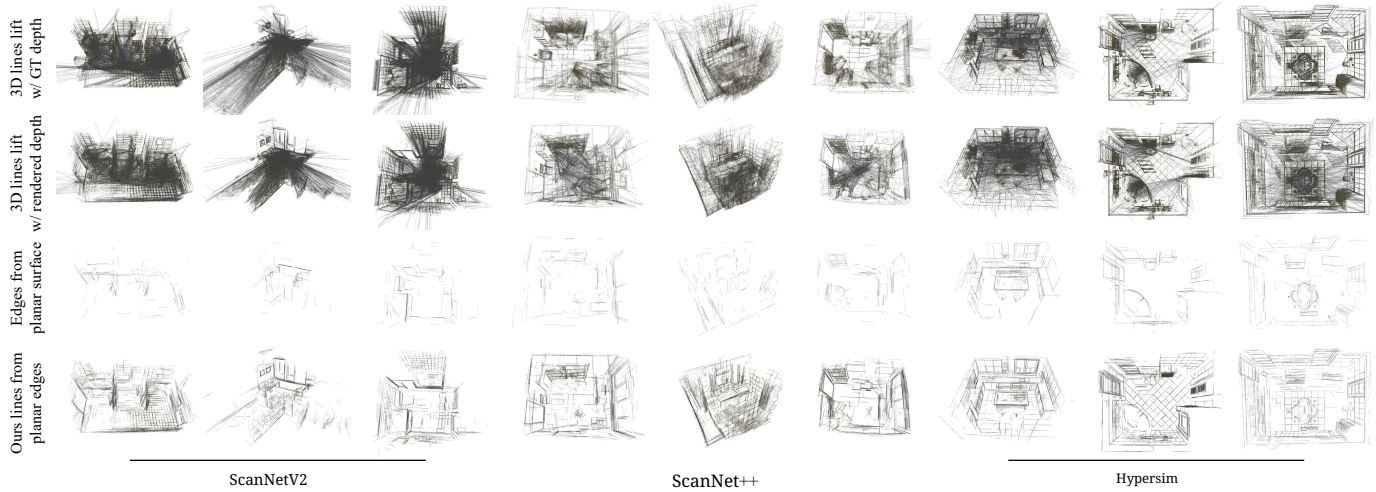


图 1：三个公共数据集场景上不同三维线图的比较。第一行：使用由真实网格渲染的深度，将二维检测线提升为三维直线。第二行：使用由 PlanarSplatting [8] 渲染的深度，将二维检测线提升为三维直线。第三行：从 PlanarSplatting [8] 提取平面边。最后一行：我们的三维线图。二维线段由 DeepLSD [11] 检测。“提升”是指使用传感器／预测深度图将二维检测线反投影到三维。

[13], [14] 应用于多视图输入图像，可优化一组三维平面，优化目标包括以下两个关键学习目标：

- 三维平面应尽可能与输入图像的 2.5D 深度／法线图对齐。
- 三维平面的边缘（或边界）应与观测到的二维线段保持一致。

如图 2 所示，给定一组具有位姿的多视图图像，我们的 *LiP-Map*通过优化一组可学习的三维平面基元，并利用深度图和法线图进行监督，从而重建场景的平面表面。同时，在多视图检测到的二维线段的监督下，场景平面表面模型的几何结构逐步得到细化。

通过构建联合优化来约束上述两个目标，LiP-Map 在三维线映射中实现了最先进的准确度和完整性。在公共基准上进行的大量实验，包括来自 ScanNetV2 [15]数据集的 50 个场景、来自 ScanNet++ [16]数据集的 30 个场景，以及来自 Hypersim [17] 数据集的 10 个场景，均展示了其优势。此外，在 7Scenes [18]和 Tanks&Temples [19] 上使用 VGGT [20] 提供的相机位姿进行的定性比较，突显了 LiP-Map 在无位姿输入下的鲁棒性。除重建之外，我们的三维线图还为线辅助视觉定位带来了实际收益：提升后的三维基元能够提供稳定且具有几何意义的对应关系，补充点特征，并在无纹理或重复纹理区域保持鲁棒性。在 7Scenes [18] 数据集上，将这些基元集成到标准的纯点定位流程中，相较于纯点基线方法和现有的点线联合方法，显著提高了位姿精度和鲁棒性。此外，LiP-Map 还能提升平面三维重建的质量，进一步彰显了其多功能性。

2 相关工作

三维线映射。将多视图图像中的二维特征映射到三维空间，是一项基础任务，具有广泛的应用——涵盖三维视觉、增强现实和机器人学等领域 [21], [22], [23], [24]。然而，在结构占主导地位且表面往往缺乏纹理的室内场景中，传统的基于二维关键点的建图方法 [25], [26], [27], [28]往往表现欠佳。因此，三维线映射已成为传统三维点建图的一种颇具吸引力的替代方案。近期，二维线段检测 [11], [13], [14], [29], [30], [31]和匹配 [10], [32] 技术取得进展，使得三维点建图流程能够应用于三维线映射 [1], [3], [33]。然而，由于二维线特征与点特征存在根本差异，匹配过程往往会引入固有的不确定性，有时甚至导致失败。为解决这一问题，一些研究 [1], [3], [34] 利用几何约束来搜索线对应关系，从而改善重建效果。近年来，基于神经场的方法 [2], [4]则尝试消除显式的线匹配，取得了更好的映射结果。

结构化三维表面重建。除了三维线映射之外，许多研究还探索使用不同类型的基元进行三维重建，例如表面元素 [35], [36],[37]，平面 [5], [6], [8], [38], [39], [40], [41], [42], [43],多边形网格 [44], [45], [46], [47], 草图 [48], 和隐式曲面 [49], [50], [51], 等。其中，PlanarSplatting [8] 提出的平面三维重建在室内场景中展现出巨大潜力，它以超快速度显式优化由 2.5D线索推导出的三维平面所构成的表面几何。与其他方法相比，对平面表面几何进行显式建模和优化有几项优势：(1) 无需代理表示，即可始终以可微的方式获取预期几何结构；(2) 平面表面具有确定的几何含义，并以线性结构近似场景几何。

我们的研究属于免匹配解决方案这一类，并采用三维平面泼溅 [8] ，因为它计算效率高，且平面与线表示之间存在内在联系。据我们所知，这

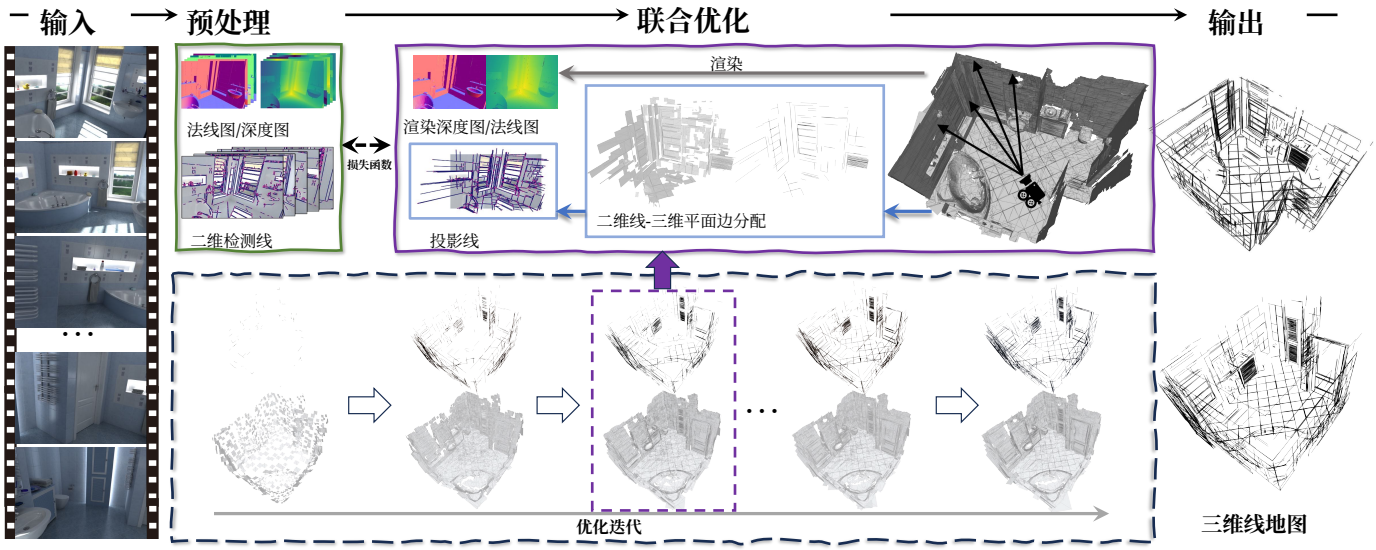


图 2：我们提出的重建流程概览，这是一种通过探索三维平面与线之间结构协同关系的新型三维线映射方法。

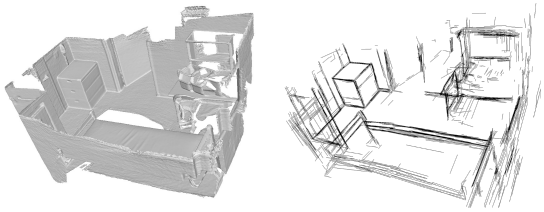


图 3：ScanNetV2 场景 [15] 中表面（左）与三维直线（右）的示意图。三维直线与物理表面边界紧密对齐，凸显了这两种结构之间的强相关性。

是首项明确优化三维线段与平面之间协同关系的工作，为结构化三维重建提供了全新视角。

3 *LiP-Map*：线-平面联合映射

给定一组带有位姿的多视图图像，以及单目二维和 2.5 维草图，

$$\mathcal{I} = \{(\mathbf{I}_i, K_i, \xi_i, \mathbf{D}_i, \mathbf{N}_i, \mathbf{L}_i)\}_{i=1}^{N} \tag{1}$$

其中 $\mathbf{I}_i$ 是第 i 张图像， K_i 和 ξ_i 分别是内参矩阵和外参矩阵， $\mathbf{D}_i$ 和 $\mathbf{N}_i$ 分别是深度图和法线图（可为真实值，也可由预训练的三维基础模型 [52], [53]预测得到）， $\mathbf{L}_i$ 是由任意线段检测器 [11], [12],[13], [14], 检测到的二维线段集合，记为

$$\mathbf{L}_i = \left\{ \mathbf{l}_{i,j}^{\text{det}} = (\mathbf{x}_{i,j}^1, \mathbf{x}_{i,j}^2) \in \mathbb{R}^2 \times \mathbb{R}^2 \right\}. \tag{2}$$

我们的目标是利用这些2D/2.5D几何线索重建对应场景的三维线地图。最终的线映射结果表示为

$$\mathbb{L} = \left\{ \mathbf{l}_i^{3d} = (\mathbf{u}_i^{3d}, \mathbf{v}_i^{3d}) \in \mathbb{R}^3 \times \mathbb{R}^3 \right\}. \tag{3}$$

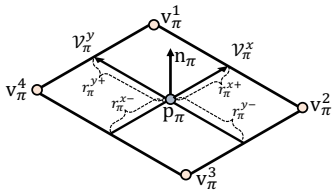


图 4：具有可学习形状参数的三维矩形平面的表示。

3.1 平面表面的表示

我们遵循近期的PlanarSplatting [8],，该方法使用一组三维矩形平面 $\mathbf{\Pi} = \{\pi_i\}_{i=1}^{K}$从 $\mathcal{I}$重建场景。这一做法的动机来自如下观察（见图3）：三维矩形平面天然地将表面与三维直线联系起来：平面内部近似场景表面，而其边涵盖目标三维直线。

三维平面参数化。 如图 4 所示，且如 PlanarSplatting [8], 中所述，每个三维矩形平面π 都配有可学习参数，包括平面中心$\mathbf{p}_\pi \in \mathbb{R}^3$、平面旋转$\mathbf{q}_\pi \in \mathbb{R}^4$ （以四元数表示）以及平面半径$\mathbf{r}_\pi = \{r_\pi^{x+}, r_\pi^{x-}, r_\pi^{y+}, r_\pi^{y-}\} \in \mathbb{R}^4$。其中，符号“+/−”表示矩形 X轴/Y轴的“正/负”方向。随后，可以进一步计算三维矩形平面的 X轴正方向 $\mathcal{V}_\pi^x \in \mathbb{R}^3$ 和 Y轴正方向 $\mathcal{V}_\pi^y \in \mathbb{R}^3$:

$$\mathcal{V}_\pi^x = \mathbf{R}(\mathbf{q}_\pi)[1,0,0]^\top, \quad \mathcal{V}_\pi^y = \mathbf{R}(\mathbf{q}_\pi)[0,1,0]^\top, \tag{4}$$

其中，$\mathbf{R}(\mathbf{q}_\pi) \in \mathbb{R}^{3\times3}$ 是由$\mathbf{q}_\pi$转换得到的旋转矩阵。三维矩形平面$\mathbf{n}_\pi \in \mathbb{R}^3$ 的法向量可计算如下:

$$\mathbf{n}_\pi = \mathbf{R}(\mathbf{q}_\pi)[0,0,1]^\top. \tag{5}$$

这些可学习的三维平面 $\mathbf{\Pi}$ 通过 2.5D 几何线索（即单目深度和法线图）进行优化，并借助可微光栅化重建场景。更多详情请参阅 [8]或我们的附录 A。

作为直线的平面边。基于上述表示，我们进一步依次定义三维平面 π 的四个平面顶点集合$\mathbf{V}_\pi = \{\mathbf{v}_\pi^i \in \mathbb{R}^3\}_{i=1}^4$：

$$\begin{aligned}\mathbf{v}_\pi^1 &= \mathbf{p}_\pi + r_\pi^{x+}\mathcal{V}_\pi^x + r_\pi^{y+}\mathcal{V}_\pi^y,\\ \mathbf{v}_\pi^2 &= \mathbf{p}_\pi + r_\pi^{x+}\mathcal{V}_\pi^x - r_\pi^{y-}\mathcal{V}_\pi^y,\\ \mathbf{v}_\pi^3 &= \mathbf{p}_\pi - r_\pi^{x-}\mathcal{V}_\pi^x - r_\pi^{y-}\mathcal{V}_\pi^y,\\ \mathbf{v}_\pi^4 &= \mathbf{p}_\pi - r_\pi^{x-}\mathcal{V}_\pi^x + r_\pi^{y+}\mathcal{V}_\pi^y.\end{aligned} \tag{6}$$

那么，三维平面 π 的四条平面边$\mathbf{E}_\pi = \{\mathbf{e}_\pi^i\}_{i=1}^4$ 可表示为：

$$\begin{aligned}e_\pi^1 &= (\mathbf{v}_\pi^1, \mathbf{v}_\pi^2), \quad e_\pi^2 = (\mathbf{v}_\pi^2, \mathbf{v}_\pi^3),\\ e_\pi^3 &= (\mathbf{v}_\pi^3, \mathbf{v}_\pi^4), \quad e_\pi^4 = (\mathbf{v}_\pi^4, \mathbf{v}_\pi^1).\end{aligned} \tag{7}$$

对于由 K 3D 个平面 $\mathbf{\Pi} = \{\pi_i\}_{i=1}^K$表示的场景，目标三维直线 $\mathbb{L}$ 是所有三维平面边缘的子集：

$$\mathbb{L} \subseteq \{\mathbf{E}_{\pi_i}\}_{i=1}^K. \tag{8}$$

这将三维线映射简化为一项任务：优化与二维直线检测结果对齐的平面边缘，同时优化平面表面。

3.2 二维直线到三维平面边缘的分配

为了在平面边缘 $\{\mathbf{E}_{\pi_i}\}_{i=1}^K$中重建三维直线 $\mathbb{L}$，我们必须找出可能对应于三维直线的平面边缘。为避免歧义，我们将这类边缘称为“直线边”，记作 $\mathcal{E} = \{e_{\pi_i}^j\}$，其中 $i \in \{1, 2, ..., K\}$、$j \in \{1, 2, 3, 4\}$，且 $\mathcal{E} \subseteq \{\mathbf{E}_{\pi_i}\}_{i=1}^K$。我们使用任意直线检测器检测到的二维线段 $\mathbf{L}_i$ 来寻找“直线边”。由于所有平面都处于优化过程中，直接将任何平面边缘 $\mathbf{E}_{\pi_i}$ 与二维观测 $\mathbf{l}_i^{\text{det}}, _j \in \mathbf{L}_i$ 关联起来会很脆弱。我们从二维线段 $\mathbf{L}_i$的吸引场表示 [13] 入手解决这一问题。记 $\mathcal{R}_{i,j}$ 为线段 $\mathbf{l}_i^{\text{det}}, _j$ 的 1像素区域，其中任意像素 $\mathbf{p} \in \mathcal{R}_{i,j}$ 到 $\mathbf{l}_i^{\text{det}}, _j$ 的距离不超过 1 像素；然后，我们逐像素地根据像素 $\mathbf{p}$ 及其所支持的线段 $\mathbf{l}_i^{\text{det}}, _j$，关联最可能的平面基元。图 5 展示了使用 DeepLSD [11] 检测到的线段，以及检测结果的所有 1 像素区域。

从$\mathbf{p}$ 位于$\mathbf{l}_i^{\text{det}}, _j$的1像素区域中的一个像素出发，我们逐像素投射一条对应射线，该射线始于图像$\mathbf{I}_i$的相机中心，并找到射线首先命中的三维平面 π。接下来，为了确定三维平面 π 的哪条三维边是目标“直线边”，我们将四条三维边$\mathbf{E}_\pi = \{e_\pi^v\}_{v=1}^4$ 全部投影到二维图像上。投影后的二维边可定义为：

$$\mathbf{E}_\pi^{2d} = \left\{e_{\pi,v}^{2d}\right\}_{v=1}^4. \tag{9}$$

由于三维平面是在运行过程中实时优化的，在收敛之前，投影二维边$\mathbf{E}_\pi^{2d}$可能无法与检测到的二维线段$\mathbf{l}_i^{\text{det}}, _j$ 很好地对齐。为了解决这个问题，我们首先比较$\mathbf{E}_\pi^{2d}$中的边与对应检测二维直线之间的角距离，并丢弃角距离最大的两条边。我们
然后计算剩余边与对应检测二维直线之间的正交距离，以选择最佳候选项 $e_{\pi, v^*}^{2d} \in \mathbf{E}_\pi^{2d}$。对于最佳候选项 e_{π, v^*}^{2d}

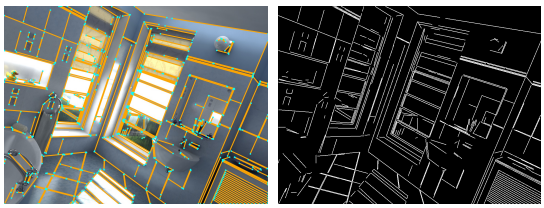


图 5：DeepLSD [11] 在 Hypersim [17] 场景“ai 001 001”的单视图中检测到的二维线段及其对应的单像素区域示意图。左：检测到的二维直线。右：对应的单像素区域。– –

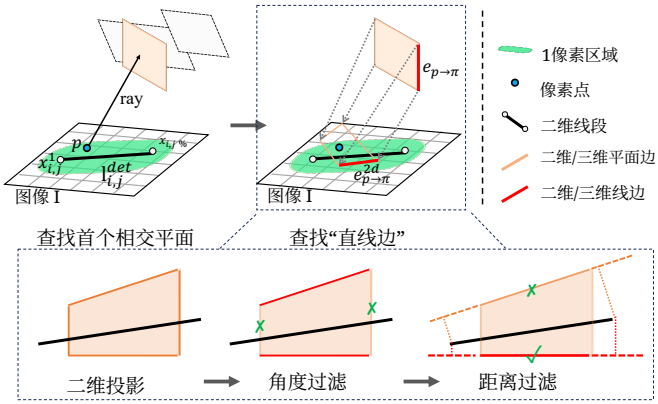


图 6：二维线到三维平面边缘分配示意图。根据角度和距离筛选出最佳的“直线边”候选项。

对于 $\mathbf{l}_i^{\text{det}}, _j$，我们将像素 $\mathbf{p}$处选定的二维/三维“直线边”定义为

$$e_{\mathbf{p}\to\pi}^{2d} := e_{\pi,v^*}^{2d} = (\mathbf{v}_{\pi,v^*}^1, \mathbf{v}_{\pi,v^*}^2), \quad e_{\mathbf{p}\to\pi} := e_\pi^{v^*}, \tag{10}$$

其中$(\mathbf{v}_{\pi,\ v^*}^1, \mathbf{v}_{\pi,\ v^*}^2)$ 是投影二维边 $e_{\pi,\ v^*}^{2d}$ 的两个顶点。

现在，我们从像素 p 得到二维线段与二维/三维“直线边”之间的分配，记为$S(p) = (\mathbf{l}_i^{\text{det}}, _j, e_{\mathbf{p}\to\pi}^{2d}, e_{\mathbf{p}\to\pi})$。在图像 $\mathbf{I}_i$ 上遍历所有检测到的二维线段$\mathbf{L}_i$ 的所有 1 像素区域后，我们最终得到了所有二维直线与三维平面边之间对应于有效像素的分配。随后，我们利用这些分配联合优化三维直线（“直线边”）和三维平面，相关内容将在下一节中介绍。

为清晰起见，我们在图7中说明了这一过程。如图所示，从单像素区域中的所有像素投射射线，以找到最先相交的平面基元。为此，我们首先计算所有射线与平面基元所在平面之间的交点。然后，我们检查每个交点是否位于该基元的边界内，以及其深度值是否为正，从而识别被射线有效相交的基元。在这些有效交点中，我们沿每条射线选择深度值最小的最近平面基元。通过这种方式，我们识别出可能包含三维线段的三维平面。最后，如第3.3节所述，对所选平面基元进行联合优化，以重建相应的三维直线。它们的边

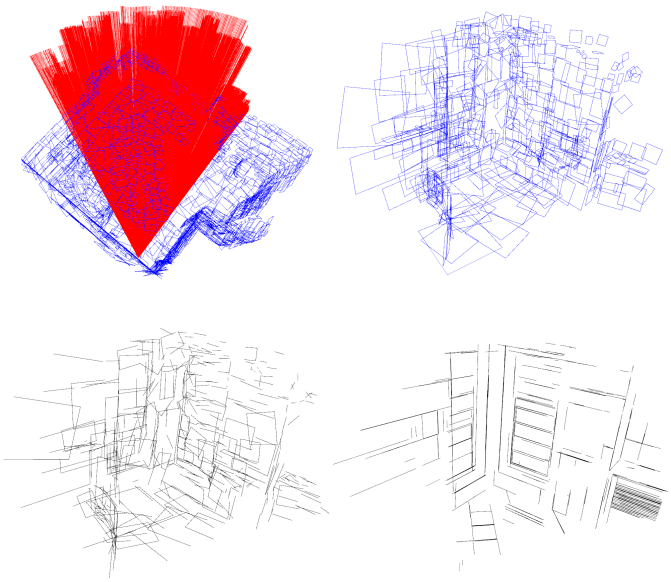


图 7：从一个视图进行分配过程的示意图。**红色**：射线。**蓝色**：平面基元。**黑色**：平面边。**左上**：射线与平面基元的关联。**右上**：选定的平面基元。**左下**：选定的平面边。**右下**：优化后的平面边。

与检测到的二维线良好对齐，从而实现可靠且合理的重建。

3.3 平面-直线优化

给定一幅图像 $\mathbf{I}$ 及其二维直线到三维平面边的分配$\mathcal{S}_{\mathbf{I}}$，我们的目标是优化分配得到的三维“直线边”，以实现线映射。我们通过在三维和二维空间中约束“直线边”来实现这一目标。

在三维空间中，我们必须确保这些“直线边”位于场景表面上。请注意，“直线边”本身是三维平面的一部分。因此，可以通过使用由 2.5D 单目深度和法线监督的平面渲染损失 $\mathcal{L}_{\mathbf{\Pi}}$ 来优化三维平面，从而直接实现表面约束，其形式如下：

$$\mathcal{L}_{\mathbf{\Pi}} = \alpha_D \mathcal{L}_{\mathbf{D_{epth}}} + \alpha_N \mathcal{L}_{\mathbf{N_{ormal}}}, \tag{11}$$

其中 $\alpha_D = 5$和 $\alpha_N = 1$。这一渲染过程已在 PlanarSplatting [8],中介绍，我们也在附录 A 中进行了介绍。

在二维空间中，我们需要确保“直线边”与图像 $\mathbf{I}$ 上检测到的二维线段对齐。具体而言，给定所有二维直线到三维平面边的分配$\mathcal{S}_{\mathbf{I}}$，二维对齐损失可计算为：

$$\mathcal{L}_{\text{2d}} = \mathcal{L}^{2d}_{\text{euc}} + \mathcal{L}^{2d}_{\text{ort}}, \tag{12}$$

二维端点损失 $\mathcal{L}^{2d}_{\text{euc}}$ 和二维直线损失 $\mathcal{L}^{2d}_{\text{ort}}$ 的计算方式如下：

$$\begin{aligned}\mathcal{L}^{2d}_{\text{euc}} &= \sum_{\mathbf{l}^{\text{det}}_{i,j} \in \mathbf{L}_i} \sum_{\mathbf{p} \in \mathcal{R}(\mathbf{l}^{\text{det}}_{i,j})} \mathcal{D}_{\text{euc}}(\mathbf{l}^{\text{det}}_{i,j}, e^{2d}_{\mathbf{p}\to\pi}),\\ \mathcal{L}^{2d}_{\text{ort}} &= \sum_{\mathbf{l}^{\text{det}}_{i,j} \in \mathbf{L}_i} \sum_{\mathbf{p} \in \mathcal{R}(\mathbf{l}^{\text{det}}_{i,j})} \mathcal{D}_{\text{ort}}(\mathbf{l}^{\text{det}}_{i,j}, e^{2d}_{\mathbf{p}\to\pi}),\end{aligned} \tag{13}$$

其中

$$\begin{aligned}\mathcal{D}_{\text{euc}}(\mathbf{l}^{\text{det}}_{i,j}, e^{2d}_{\mathbf{p}\to\pi}) = \min(&\|\mathbf{x}^1_{i,j} - \mathbf{v}^1_{\pi,v^*}\| + \|\mathbf{x}^2_{i,j} - \mathbf{v}^2_{\pi,v^*}\|,\\ &\|\mathbf{x}^1_{i,j} - \mathbf{v}^2_{\pi,v^*}\| + \|\mathbf{x}^2_{i,j} - \mathbf{v}^1_{\pi,v^*}\|),\end{aligned} \tag{14}$$

$$\begin{aligned}\mathcal{D}_{\text{ort}}(\mathbf{l}^{\text{det}}_{i,j}, e^{2d}_{\mathbf{p}\to\pi}) = &\frac{\|(\mathbf{v}^1_{\pi,v^*} - \mathbf{x}^2_{i,j}) \times (\mathbf{x}^1_{i,j} - \mathbf{x}^2_{i,j})\|}{\|\mathbf{x}^1_{i,j} - \mathbf{x}^2_{i,j}\|}\\ &+ \frac{\|(\mathbf{v}^2_{\pi,v^*} - \mathbf{x}^2_{i,j}) \times (\mathbf{x}^1_{i,j} - \mathbf{x}^2_{i,j})\|}{\|\mathbf{x}^1_{i,j} - \mathbf{x}^2_{i,j}\|}.\end{aligned} \tag{15}$$

这里，$(\mathbf{x}^1_{i,j},\mathbf{x}^2_{i,j})$ 是二维线段$\mathbf{l}^{\text{det}}_{i,j}$ 的两个端点，而$(\mathbf{v}^1_{\pi,v^*},\mathbf{v}^2_{\pi,v^*})$ 是投影 “ 直线边 ” $e^{2d}_{\mathbf{p}\to\pi}$ 的两个端点。

除了上述两个分别用于约束二维和三维空间中每条“直线边”的损失函数外，我们还引入群组损失 $\mathcal{L}_{\text{group}}$ ，以鼓励分配给同一条二维线段的所有“直线边”在三维空间中彼此足够接近，其计算方式为：

$$\mathcal{L}_{\text{group}} = \sum_{\forall \mathbf{l}^{\text{det}}_{i,j} \in \mathbf{L}_i} \sum_{\forall \mathbf{p_1}, \mathbf{p_2} \in \mathcal{R}(\mathbf{l}^{\text{det}}_{i,j})} \mathcal{D}_{\text{ort}}(e_{\mathbf{p_1}\to\pi}, e_{\mathbf{p_2}\to\pi}), \tag{16}$$

¨其中， p_1， p_2 是线段 l“1像素区域” $\mathcal{R}(l)$ ¨内的像素。随后可计算最终损失：

$$\mathcal{L} = \alpha_{\mathbf{\Pi}} \mathcal{L}_{\mathbf{\Pi}} + \alpha_L \mathcal{L}_{\mathbf{E}}, \tag{17}$$

其中，在我们的实验中， $\mathcal{L}_{\mathbf{E}} = (\mathcal{L}_{\text{2d}} + \mathcal{L}_{\text{group}})$、 $\alpha_{\mathbf{\Pi}} = 10$ 和 $\alpha_L = 0.1$ 。

3.4 线映射定稿

优化过程收敛后，提取三维直线十分简单，因为每条直线都对应于一个优化后的三维平面的边缘。为确保映射结果的准确度，我们通过计算角距离 $d_{\text{ang}} = \mathcal{D}_{\text{ang}}(\mathbf{l}^{\text{det}}_{i,j}, e^{2d}_{\mathbf{p}\to\pi})$ 和正交距离 $d_{\text{orth}} = \mathcal{D}_{\text{ort}}(\mathbf{l}^{\text{det}}_{i,j}, e^{2d}_{\mathbf{p}\to\pi})$ 来筛选内点，这两个距离均在二维投影$e^{2d}_{\pi,v}$ 与其对应的二维线检测$\mathbf{l}^{\text{det}}_{i,j}$ 之间计算。如果 $d_{\text{ang}} \le \tau_\alpha$且 $d_{\text{orth}} \le \tau_d$，则将三维平面边缘 e^v_π 视为对检测到的二维线段$\mathbf{l}^{\text{det}}_{i,j}$的正确解释，并将其保留在最终映射结果中。我们在实验中设置 $\tau_d = 1$像素和 $\tau_\alpha = 0.01$ 。

线跟踪构建器。获得最终的三维线图后，我们随后在 $\mathbb{L}$ 与图像集 $\mathcal{I}$ 中的所有二维检测结果之间建立三维—二维线对应关系。记为 $\mathcal{T}(\mathbf{l}^{3d}_i) = \{(v_1, \imath_1) \ldots, (v_T, \imath_T)\}$ 的$\mathbf{l}^{3d}_i$的跟踪二维线段，任意检测到的二维线段$\mathbf{l}^{\text{det}}_{v_k,\imath_k} \in \mathbf{L}_{v_k}$都应满足以下条件：二维投影$\mathbf{l}^{v_k}_i$与$\mathbf{l}^{3d}_i$以及$\mathbf{l}^{\text{det}}_{v_k,\imath_k}$之间，

- 角距离 $d_{\text{ang}} = \mathcal{D}_{\text{ang}}(\mathbf{l}_{\text{det}v_k,\imath_k}, \mathbf{l}^{v_k}_i)$ 应小于 $\tau_a = 0.01$;
- 正交距离 d_{dist} 在$\mathbf{l}_{\text{det}v_k,\imath_k}$ 与$\mathbf{l}^{v_k}_i$之间应小于 $\tau_d = 2$ 像素;
- 重叠率 d_{overlap} 在$\mathbf{l}_{\text{det}v_k,\imath_k}$ 与$\mathbf{l}^{v_k}_i$之间应大于 $\tau_o = 0.2$。

我们在附录 B 中提供了用于计算 d_{ang}、 d_{dist} 和 d_{overlap} 的示意图和数学公式。

表1：使用4种不同的检测器检测出的线，在ScanNetV2 [15] 数据集和ScanNet++ [16] 数据集上进行三维线映射的定量结果。所有指标均以5毫米为单位报告，并给出了重建线段的平均数量。

ScanNetV2 [15]：50个场景		线段级指标					交汇点级指标					统计
检测器	方法	ACC-L↓	COMP-L↓	PREC-L↑	RECAL-L↑	F-SCORE-L↑	ACC-J↓	COMP-J↓	PREC-J↑	RECAL-J↑	F-SCORE-J↑	线条数↑
LSD [12]	LIMAP [1]	0.0748	0.4094	0.5912	0.0807	0.1436	0.0753	0.4183	0.5873	0.0502	0.0927	328
	带深度的 LIMAP [1]	0.1083	0.5499	0.4853	0.0253	0.0454	0.1137	0.5540	0.4904	0.0161	0.0298	241
	CLMAP [34]	**0.0692**	0.3326	0.5893	0.1333	0.2228	**0.0702**	0.3494	0.5837	0.0728	0.1302	529
	Ours	0.0765	**0.1260**	**0.7069**	**0.3271**	**0.4438**	0.0774	**0.1433**	**0.7068**	**0.2083**	**0.3183**	**2974**
HAWPv3	LIMAP [1]	0.0791	0.4510	0.6699	0.0628	0.1139	0.0811	0.4636	0.6547	0.0293	0.0555	172
	带深度的 LIMAP [1]	0.0833	0.4092	0.5402	0.0553	0.0960	0.0853	0.4285	0.5326	0.0231	0.0426	237
	CLMAP [34]	0.0782	0.3955	0.7027	0.0895	0.1575	0.0799	0.4090	0.6927	0.0472	0.0873	389
	Ours	**0.0735**	**0.1168**	**0.7424**	**0.3521**	**0.4744**	**0.0743**	**0.1350**	**0.7398**	**0.2156**	**0.3301**	**2942**
DeepLSD	LIMAP [1]	0.0733	0.3673	0.6451	0.1064	0.1814	0.0736	0.3816	0.6436	0.0571	0.1038	397
	带深度的 LIMAP [1]	0.0783	0.3469	0.5097	0.0804	0.1297	0.0786	0.3615	0.5165	0.0420	0.0732	658
	CLMAP [34]	**0.0748**	0.2963	0.6624	0.1610	0.2585	**0.0758**	0.3185	0.6569	0.0798	0.1407	613
	Ours	0.0776	**0.1202**	**0.7308**	**0.3504**	**0.4650**	0.0781	**0.1388**	**0.7310**	**0.2158**	**0.3292**	**3015**
ScaleLSD	LIMAP [1]	0.0828	0.3217	0.6413	0.1294	0.2155	0.0843	0.3356	0.6326	0.0710	0.1263	499
	带深度的 LIMAP [1]	0.0829	0.2782	0.5437	0.1182	0.1840	0.0840	0.2977	0.5411	0.0589	0.1004	733
	CLMAP [34]	0.0827	0.2711	0.6939	0.1765	0.2822	0.0840	0.2831	0.6863	0.1121	0.1915	1186
	Ours	**0.0742**	**0.1384**	**0.7422**	**0.3399**	**0.4430**	**0.0743**	**0.1537**	**0.7431**	**0.2078**	**0.3204**	**2934**
ScanNet++ [16]：30 个场景		线段级指标					交汇点级指标					统计
检测器	方法	ACC-L↓	COMP-L↓	PREC-L↑	RECAL-L↑	F-SCORE-L↑	ACC-J↓	COMP-J↓	PREC-J↑	RECAL-J↑	F-SCORE-J↑	线条数↑
LSD	LIMAP [1]	0.0718	1.1532	0.5283	0.0281	0.0515	0.0721	1.1604	0.5396	0.0141	0.0266	73
	带深度的 LIMAP [1]	0.1084	1.5770	0.4355	0.0025	0.0050	0.1038	1.5790	0.4593	0.0014	0.0029	16
	CLMAP [34]	**0.0648**	0.8684	0.5308	0.0689	0.1183	**0.0633**	0.8821	0.5403	0.0344	0.0689	218
	Ours	0.0759	**0.2585**	**0.5653**	**0.2900**	**0.3833**	0.0744	**0.2718**	**0.5583**	**0.1432**	**0.2279**	**3795**
HAWPv3	LIMAP [1]	0.0870	1.2963	0.4876	0.0194	0.0375	0.0878	1.3043	0.4902	0.0081	0.0160	41
	带深度的 LIMAP [1]	0.1337	0.5728	0.3699	0.0343	0.0617	0.1333	0.5918	0.3743	0.0144	0.0273	228
	CLMAP [34]	**0.0783**	1.0479	0.5578	0.0346	0.0635	**0.0792**	1.0561	0.5635	0.0186	0.0360	141
	Ours	0.0890	**0.2428**	**0.5724**	**0.2664**	**0.3636**	0.0906	**0.2568**	**0.5761**	**0.1741**	**0.2673**	**2691**
DeepLSD	LIMAP [1]	0.0785	1.0347	0.4114	0.0416	0.0771	0.0769	1.0465	0.4278	0.0185	0.0356	102
	带深度的 LIMAP [1]	0.1145	0.8208	0.3629	0.0151	0.0282	0.1114	0.8335	0.3685	0.0071	0.0137	108
	CLMAP [34]	**0.0729**	0.8035	0.5034	0.0863	0.1512	**0.0715**	0.8220	0.5179	0.0396	0.0741	280
	Ours	0.0828	**0.2466**	**0.5770**	**0.3056**	**0.3996**	0.0801	**0.2621**	**0.5669**	**0.2181**	**0.3157**	**3948**
ScaleLSD	LIMAP [1]	0.0825	1.0197	0.4507	0.0402	0.0752	0.0824	1.0280	0.4498	0.0195	0.0376	110
	带深度的 LIMAP [1]	0.1201	0.3798	0.3907	0.0920	0.1453	0.1191	0.4039	0.3965	0.0445	0.0782	740
	CLMAP [34]	0.0813	0.8639	0.5663	0.0708	0.1272	0.0812	0.8714	0.5677	0.0433	0.0807	382
	Ours	**0.0801**	**0.2520**	**0.5889**	**0.2895**	**0.3882**	**0.0811**	**0.2665**	**0.5959**	**0.2039**	**0.3038**	**2749**

4 实验

本节展示了在公开数据集上的实验结果和消融实验，以证明我们的方法及其组成部分的有效性。更多详细信息和补充结果见附录。

4.1 实现细节

初始化。我们首先利用所有视图提供的深度图生成粗略场景网格。然后从该网格中随机采样 2,000 个点，作为三维平面基元的初始中心。每个基元的初始半径 π 设为其π 最近邻基元距离的一半。平面旋转则使用粗略网格的表面法线进行初始化。

优化。我们的 *LiP-Map*使用 PyTorch [54]实现，并通过 Adam [55]进行优化。对于每个场景，我们从头开始联合优化线和曲面，训练 60 个轮次，每个轮次遍历所有可用视点一次。可学习参数（包括平面中心、半径和旋转）的学习率固定为 0.001。

平面剪枝与分裂。在优化过程中，我们采用 Planar-Splatting [8], 中提出的平面分裂操作，并根据平面半径的梯度优化，以更好地拟合场景几何结构。如果沿 X轴方向的平均半径梯度（r_π^{x+}和 r_π^{x-}）超过 0.2，则沿 Y轴方向分裂该平面。类似地，如果沿 Y轴方向的半径梯度（r_π^{y+}和 r_π^{y-}）大于 0.2，则沿 X轴方向分裂该平面。与 PlanarSplatting 不同，

表2：在ScanNetV2 [15] 数据集和ScanNet++ [16]数据集上，使用4种不同的检测器检测到的线进行三维线映射的定量结果。 R_τ 和 P_τ 在5 毫米、10毫米、50毫米处给出，并附带支持图像/线段的平均数量。

ScanNetV2 [15]：50 个场景								
检测器	方法	R5	R10	R50	P5	P10	P50	支持数量
LSD	LIMAP [1]	1.04	6.41	31.07	25.8	**45.4**	80.2	5.6 / 5.8
	带深度的 LIMAP [1]	0.19	1.24	13.97	7.3	14.4	54.9	4.2 / 4.5
	CLMAP [34]	1.50	14.20	85.05	24.2	44.6	79.6	8.6 / 8.8
	Ours	**6.23**	**40.23**	**240.29**	**27.6**	43.0	**81.7**	**10.7** / **14.1**
HAWPv3	LIMAP [1]	0.90	5.61	28.25	28.5	**48.4**	82.0	7.1 / 12.1
	带深度的 LIMAP [1]	0.80	2.94	19.11	17.6	27.8	64.7	4.4 / 5.8
	CLMAP [34]	1.63	14.80	80.68	28.3	47.7	81.1	**8.5** / **9.1**
	Ours	**5.26**	**33.12**	**183.61**	**30.9**	47.6	**84.6**	6.3 / 8.1
DeepLSD	LIMAP [1]	1.67	10.41	53.78	30.0	**48.3**	81.3	6.4 / 6.8
	带深度的 LIMAP [1]	0.96	6.45	46.18	10.8	19.2	60.1	4.3 / 4.5
	CLMAP [34]	1.97	18.81	118.22	24.5	46.2	81.8	**9.4** / **9.6**
	Ours	**4.63**	**29.74**	**168.16**	**31.7**	47.1	**85.6**	6.2 / 6.4
ScaleLSD	LIMAP [1]	2.08	12.94	64.40	**31.3**	49.9	86.2	7.2 / 10.2
	带深度的 LIMAP [1]	1.78	11.49	54.33	15.0	25.3	64.5	4.5 / 4.4
	CLMAP [34]	4.09	38.01	204.94	27.6	**50.7**	85.0	**8.4** /9.1
	Ours	**6.74**	**42.65**	**224.81**	30.4	48.6	**87.4**	8.2 / **12.8**
ScanNet++ [16]：30 个场景								
检测器	方法	R5	R10	R50	P5	P10	P50	支持数量
LSD	LIMAP [1]	2.07	5.69	17.27	33.8	50.1	69.1	4.7 / 5.0
	带深度的 LIMAP [1]	0.07	0.23	1.03	12.3	18.8	51.6	4.1 / 4.2
	CLMAP [34]	5.63	16.29	49.65	**36.8**	52.4	71.5	**5.8** / **6.2**
	Ours	**12.87**	**48.64**	**61.20**	29.1	**64.0**	**74.1**	4.9 / 5.4
HAWPv3	LIMAP [1]	1.33	3.78	11.23	33.5	58.2	78.9	5.1 / 6.2
	带深度的 LIMAP [1]	0.24	0.69	3.06	18.1	27.1	57.1	4.2 / 4.3
	CLMAP [34]	4.09	11.86	33.80	32.4	59.3	**82.8**	**5.8** /6.4
	Ours	**4.33**	**16.90**	**40.67**	**34.3**	**72.0**	81.5	4.7 /**7.1**
DeepLSD	LIMAP [1]	3.05	8.96	28.36	38.8	53.8	74.7	5.2 / 5.9
	带深度的 LIMAP [1]	0.41	1.42	7.11	15.4	22.6	58.7	4.0 / 4.2
	CLMAP [34]	8.37	24.13	76.47	38.1	63.6	**79.5**	**6.3** / **6.6**
	Ours	**11.04**	**45.22**	**59.61**	**39.9**	**73.8**	76.6	4.9 / 5.3
ScaleLSD	LIMAP [1]	2.81	7.99	24.36	39.8	56.6	80.3	5.3 / 6.1
	带深度的 LIMAP [1]	0.61	1.93	9.50	16.7	23.1	59.4	4.1 / 4.0
	CLMAP [34]	10.04	28.73	80.88	**40.4**	67.3	**83.1**	**5.9** / 6.6
	Ours	**14.31**	**43.91**	**99.01**	36.5	**71.7**	81.5	4.6 / **6.7**

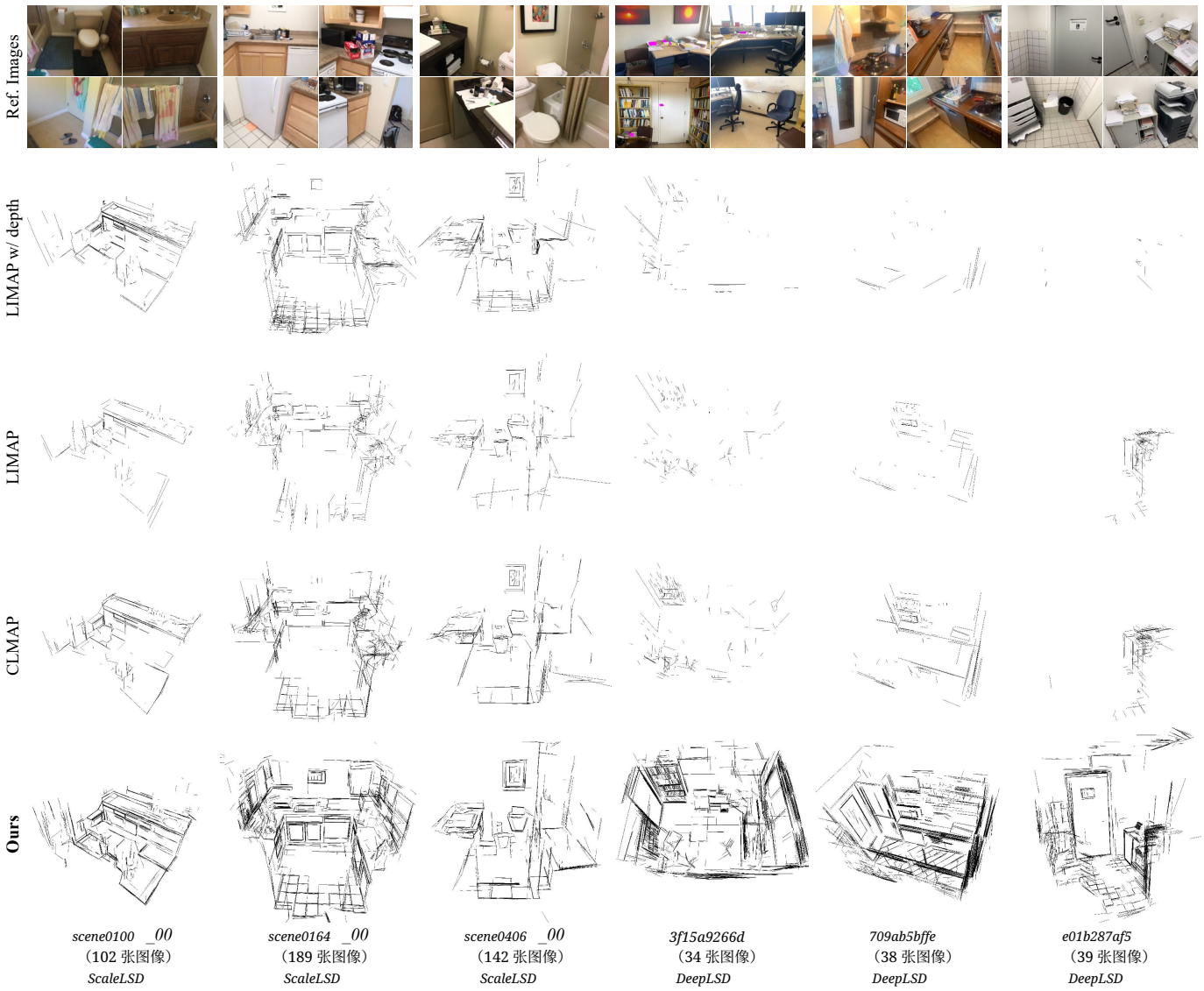


图 8：在 ScanNetV2数据集 [15]（左侧 3 个场景使用 ScaleLSD [14] 检测器）和 ScanNet++数据集 [16] （右侧 3 个场景使用 DeepLSD [11]检测器）上的定性比较。第一行：带深度的 LIMAP [1]。第二行：LIMAP [1]。第三行：CLMAP [34]。最后一行：我们的方法。

其每1,000次迭代执行一次剪枝和分裂，而我们每个训练轮次执行一次这些操作。

4.2 数据集、指标与基线方法

数据集。 我们在 ScanNetV2数据集 [15]、ScanNet++ 数据集 [16] 和 Hypersim数据集 [17]上测试了我们的方法。ScanNetV2 和 ScanNet++ 是两个提供带位姿视频的真实室内数据集。按照 PlanarSplatting [8],的测试协议，我们从 ScanNetV2数据集中随机抽取 50 个场景，从 ScanNet++ 数据集中随机抽取 30 个场景，并从每个场景的视频中每隔 10 帧抽取一张图像进行评估。与 PlanarSplatting [8],类似，我们使用 Metric3Dv2 [53] 进行深度估计，并使用 Ominidata [52] 进行法线估计。Hypersim 是一个用于整体室内场景理解的照片级真实感合成数据集。该数据集中的每个场景最多提供 100 张带位姿的图像、深度图和法线图。按照 LIMAP [1] 和 CLMAP [34],的方法，我们选取前十个场景进行评估。

指标。 由于不存在真实值（GT）三维直线，我们使用真实值网格模型或点云来评估三维线映射。为全面评估重建的三维线图的质量，我们考虑两项评估指标：一项是广泛使用的、用于评估重建三维模型的标准基准，另一项是 LIMAP [1] 提出的评估基准。

- 三维重建中广泛采用的指标（M1） [4], [56], [57], 包括准确率（ACC）、完整性（COMP）、精确率（PREC）、召回率（RECAL）和 F分数。我们在交点级（-J）和线段级（-L）评估这五项指标。对于交点级结果，我们评估三维直线两个端点（交点）的性能。对于线段级结果，我们在每条线上均匀采样 100 个点来衡量性能。此外，我们

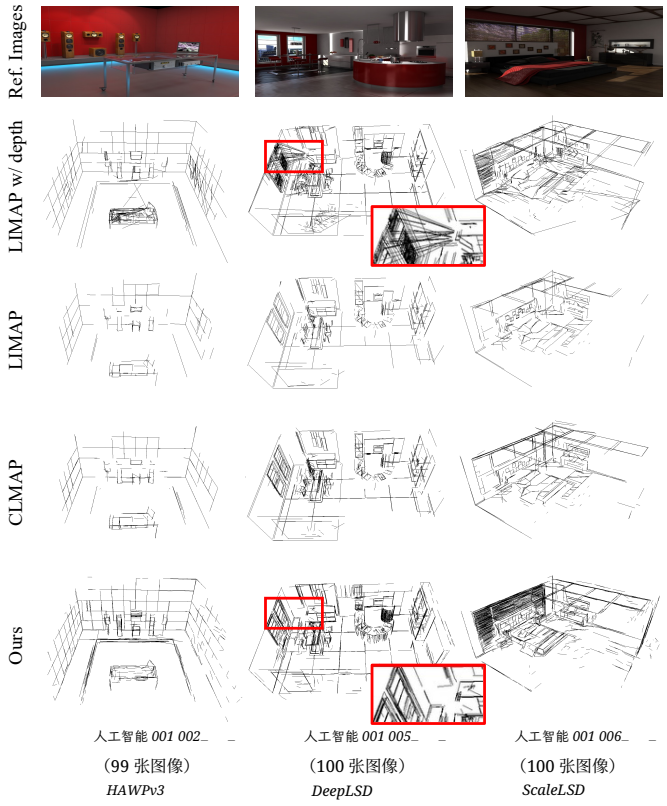


图 9：使用 3 种不同直线检测器在 Hypersim 数据集上进行的定性比较。第一行：带深度的 LIMAP [1]。第二行：LIMAP [1]。第三行：CLMAP [34]。最后一行：我们的方法。

将重建的三维线段数量作为衡量完整性和鲁棒性的统计指标。

- LIMAP 的评估指标（M2）基于线段跟踪，并使用以下指标：阈值为 τ （R_τ）时的长度召回率（单位：米），表示距离真实值模型 τ 毫米以内的线段部分的长度总和。阈值为 τ （P_τ）时的内点百分比，表示距离真实值模型 τ 毫米以内的轨迹所占百分比。平均支持数表示所有线轨迹中图像支持数和二维线段支持数的平均值。

评估指标的完整公式见附录B。

基线方法。 我们采用当前最先进的方法 LIMAP [1] 作为基线，该方法通过精心设计的评分和跟踪实现三维线映射，并自动识别和利用结构先验。由于我们使用深度图进行三维线映射，且 LIMAP 提供了基于深度图的优化，因此在所有实验中，我们都将其与基于深度的 LIMAP 进行比较。我们还与近期提出的 CLMAP [34], 进行比较，该方法基于 LIMAP 构建，旨在通过联合优化点、线和平面来提高线地图结构的一致性。与我们提出的可学习平面基元不同，CLMAP 静态地应用由 RSPD [58]检测到的平面，为三维直线施加共面约束。

表 3：在 Hypersim [17] 数据集上进行三维线映射的定量结果，其中线段由4种不同的检测器检测得到。所有指标均在 5 毫米处报告，并给出重建线段的平均数量。

Hypersim [17]：前10个场景		线段级指标					统计
检测器	方法	ACC-L↓	COMP-L↓	PREC-L↑	RECAL-L↑	F-SCORE-L↑	线条数↑
LSD [12]	LIMAP [1]	0.0731	0.5336	0.9302	0.1890	0.3085	886
	带深度的 LIMAP [1]	0.0088	0.5800	**0.9969**	0.2024	0.3297	890
	CLMAP [34]	0.1045	0.5010	0.9136	0.2028	0.3248	1209
	Ours	**0.0079**	**0.3989**	0.9921	**0.2908**	**0.4424**	**2538**
HAWPv3	LIMAP [1]	0.0435	0.7133	0.9030	0.0912	0.1648	263
	带深度的 LIMAP [1]	0.0097	0.7907	0.9896	0.1262	0.2214	238
	CLMAP [34]	0.0591	0.6678	0.9286	0.1087	0.1935	641
	Ours	**0.0077**	**0.3531**	**0.9923**	**0.2785**	**0.4295**	**2461**
DeepLSD	LIMAP [1]	0.0861	0.5174	0.8964	0.2154	0.3412	665
	带深度的 LIMAP [1]	0.0093	0.7168	**0.9934**	0.2465	0.3868	786
	CLMAP [34]	0.1088	**0.4745**	0.9044	0.2330	0.3622	1105
	Ours	**0.0076**	0.5269	0.9924	**0.3387**	**0.4963**	**3355**
ScaleLSD	LIMAP [1]	0.0520	0.6432	0.8774	0.1341	0.2296	458
	带深度的 LIMAP [1]	0.0096	0.5397	**0.9901**	0.1794	0.2982	439
	CLMAP [34]	0.0664	0.5602	0.9161	0.1593	0.2679	1225
	Ours	**0.0090**	**0.3410**	0.9857	**0.2944**	**0.4465**	**2968**

表 4：在 Hypersim [17] 数据集上进行三维线映射的定量结果，线由 4种不同的检测器检测得到。 R_τ 和 P_τ 的结果以 1 毫米、5 毫米和 10 毫米为阈值报告，同时给出了支持图像/线段的平均数量。

Hypersim [17]：前10个场景								
检测器	方法	R1	R5	R10	P1	P5	P10	支持数量
LSD	LIMAP [1]	70.5	282.5	351.5	65.7	84.0	89.6	15.6 / 17.7
	带深度的 LIMAP [1]	**268.0**	432.6	434.1	**84.0**	**99.9**	**100**	16.6 / **22.2**
	CLMAP [34]	112.3	403.0	492.7	61.9	80.6	87.9	**19.7** / 21.6
	Ours	245.3	**585.6**	**678.7**	81.7	94.6	98.7	12.3 / 20.8
HAWPv3	LIMAP [1]	15.4	66.1	95.1	65.0	81.0	88.4	12.3 / 23.5
	带深度的 LIMAP [1]	82.2	182.3	189.9	80.8	**99.7**	**100**	**15.8** / **33.0**
	CLMAP [34]	52.7	205.0	272.7	63.3	83.7	89.6	13.7 / 14.7
	Ours	**184.3**	**512.9**	**594.5**	**82.8**	95.7	99.2	9.3 / 18.2
DeepLSD	LIMAP [1]	65.3	244.2	304.4	66.4	80.7	85.6	17.2 / 23.5
	带深度的 LIMAP [1]	261.8	492.8	503.1	**84.2**	**99.8**	**100**	17.5 / **24.9**
	CLMAP [34]	138.0	455.8	551.1	63.8	80.7	87.5	**20.5** / 21.9
	Ours	**396.3**	**886.5**	**1010.5**	82.2	95.1	98.7	13.1 / 21.9
ScaleLSD	LIMAP [1]	24.5	100.6	142.4	64.2	80.9	87.0	14.0 / 29.3
	带深度的 LIMAP [1]	136.2	275.1	287.9	79.2	**99.7**	**100**	**15.6** / **35.6**
	CLMAP [34]	107.9	391.1	533.9	62.9	82.2	89.1	14.7 / 15.9
	Ours	**223.6**	**639.6**	**754.1**	**80.2**	95.4	98.7	11.0 / 32.0

4.3 三维线映射

检测器。为系统研究检测到的二维线段对最终三维线映射的影响，我们使用四种不同的直线检测器开展了全面实验：LSD [12],HAWPv3 [13],DeepLSD [11], 以及 ScaleLSD [14],，均在统一的定量评估中进行。

ScanNetV2 与 ScanNet++。 我们首先在 ScanNetV2 数据集 [15]和 ScanNet++数据集 [16]上评估我们的方法。我们使用这两个数据集提供的每个场景对应的平面网格来计算指标 M1 和 M2。指标 M1 的平均结果如表 1 所示。可以看出，我们的方法在重建三维直线的准确度和完整性之间实现了出色的平衡：完整性误差始终较低，准确性误差与其他方法相当，同时保持较高的精确率、召回率和 F1分数。总体而言，基于匹配的方法 LIMAP 和 CLMAP 能够稳健地重建更准确的三维线图，而我们的方法倾向于重建更完整的场景线结构，并得到数量多得多的三维直线。由于图像存在运动模糊，且 Metric3d 估计的深度图并不完美，带深度的LIMAP 会根据检测到的二维直线拟合出不可靠的三维直线，最终导致重建结果变差。不过，我们的方法可以大幅缓解这一问题，因为平面基元能够对每个区域的深度误差取平均，从而使我们的

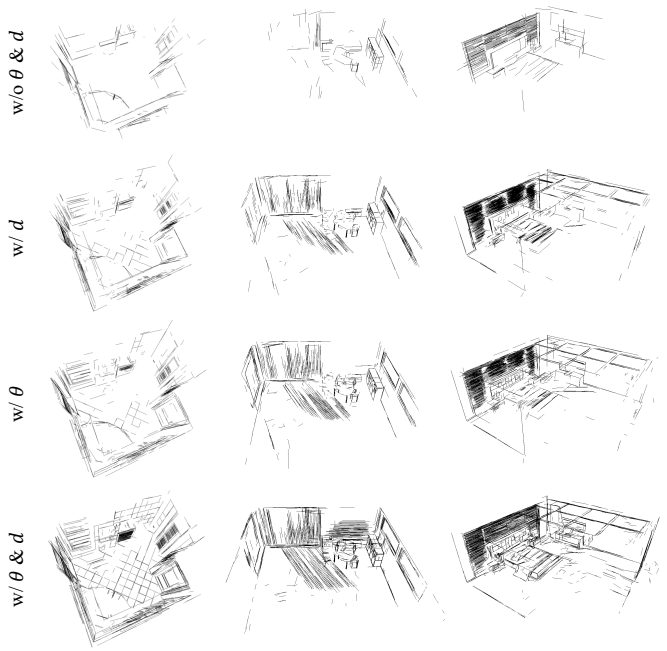


图 10：我们提出的二维直线—三维边分配消融示意图。第一行：不使用 θ 和 d。第二行：仅使用 θ。第三行：仅使用 d。最后一行：使用 θ 和 d。

方法的准确性误差低于带深度的LIMAP。指标 M2 的平均结果见表 2。我们方法的长度召回率显著高于其他方法，同时精确率也具有竞争力。LIMAP 和 CLMAP 更擅长构建较长的线轨迹，并且能够跟踪比我们的方法更多的图像支持和线支持。

图 8 展示了三维线映射的定性比较。每列下方标注了场景名称、使用的图像数量以及所用的线检测器。我们的方法不仅擅长捕捉精细结构，还能重建出整体上更完整的线图。在 ScanNetV2 数据集上，LIMAP 和 CLMAP 使用超过 100 张图像的稠密视图，基本能够重建整个场景的结构；但在视图相对稀疏的 ScanNet++数据集上，它们未能重建场景的完整结构。

Hypersim。 我们进一步展示了所有方法在 Hypersim数据集 [17]前十个场景上的比较。遵循 LIMAP [1], , 我们使用从每个场景的深度图中恢复的点云来计算指标 M1 和 M2。指标 M1 的平均结果见表 3。如图所示，由于 Hypersim 提供了准确的深度图，我们的方法和带深度的LIMAP在准确性误差更低、精确率更高方面优于 LIMAP 和 CLMAP。此外，我们的方法在完整性误差更低、召回率和 F1分数更高方面显著优于其他方法。我们的方法能够利用更多三维直线重建出更完整的结构。指标 M2 的平均结果见表 4。我们的方法也取得了高于其他方法的长度召回率，且精确率略低于带深度的LIMAP，但构建长线的能力较弱

表 5：在 Hypersim数据集 [15]上，使用 LSD [12]检测线段时，我们提出的损失函数在**指标 M1** 上的消融研究。所有指标均在 5 毫米处报告，并附有重建线段的平均数量。

$\mathcal{L}_{euc}^{2d}$	$\mathcal{L}_{ort}^{2d}$	$\mathcal{L}_{group}$	ACC-L↓	COMP-L↓	PREC-L↑	RECAL-L↑	F-SCORE-L↑	线条数↑
			0.0082	0.8447	0.9551	0.1269	0.2240	532
✓			0.0088	0.4648	0.9421	0.1512	0.2606	1358
	✓		0.0083	0.4472	0.9441	0.1619	0.2764	1654
		✓	0.0082	0.4689	0.9446	0.1330	0.2332	1301
✓	✓		0.0087	0.4516	0.9585	0.2180	0.3552	3109
✓		✓	0.0088	0.4232	0.9534	0.2011	0.3321	2929
	✓	✓	0.0081	0.4566	0.9662	0.2504	0.3977	**3439**
✓	✓	✓	**0.0079**	**0.3989**	**0.9921**	**0.2908**	**0.4424**	2538

表 6：在 Hypersim数据集 [15]上，使用 LSD [12]检测到的线段，对我们提出的损失函数进行消融研究，评估指标为**指标 M2** 。 R_τ 和 P_τ 报告了 1 毫米、5 毫米、10 毫米处的结果，以及支持图像/线段的平均数量。

$\mathcal{L}_{euc}^{2d}$	$\mathcal{L}_{ort}^{2d}$	$\mathcal{L}_{group}$	R1	R5	R10	P1	P5	P10	支持数量
			11.7	32.8	54.1	81.0	92.6	98.6	5.3 / 6.2
✓			31.5	99.4	121.1	81.1	92.0	97.5	10.0 / 11.3
	✓		39.5	117.6	141.8	81.4	92.5	97.9	10.4 / 11.8
		✓	29.7	97.2	118.5	80.9	91.9	98.2	10.2 / 11.4
✓	✓		143.7	476.7	569.2	81.5	93.0	98.4	10.8 / 13.1
✓		✓	135.3	468.0	566.2	81.7	93.2	98.1	10.9 / 13.1
	✓	✓	115.2	335.5	389.4	81.3	94.0	97.9	11.3 / 13.3
✓	✓	✓	**245.3**	**585.6**	**678.7**	**81.7**	**94.6**	**98.7**	**12.3 /20.8**

表 7：在**指标 M1** 上，对使用 LSD 检测的线 [12]在 Hypersim 数据集 [15] 中提出的二维线-三维边分配进行消融研究。所有指标均在 5 毫米处报告，同时给出重建线段的平均数量。

角度(θ)	距离(d)	ACC-L↓	COMP-L↓	PREC-L↑	RECAL-L↑	F-SCORE-L↑	线条数↑
		0.0082	0.8447	0.9551	0.1269	0.2240	532
✓		0.0087	0.4051	0.9879	0.2461	0.3866	2125
	✓	0.0082	**0.3542**	0.9900	0.2776	0.4260	**2561**
✓	✓	**0.0079**	0.3989	**0.9921**	**0.2908**	**0.4424**	2538

表 8：在 Hypersim数据集 上对所提出的二维线-三维边分配进行消融研究**指标 M2** ，使用 LSD 检测的线 [15] 在 [12]处的结果为 1 毫米、5 毫米、10 毫米， R_τ 并 P_τ 报告支持图像/线段的平均数量。

θ(θ)	$dist$(d)	R1	R5	R10	P1	P5	P10	支持数量
		11.7	32.8	54.1	81.0	92.6	98.6	5.3 / 6.2
✓		138.9	351.2	418.7	76.2	92.2	97.4	10.8 / 13.3
	✓	177.4	421.6	492.9	76.9	92.0	96.9	11.5 / 14.1
✓	✓	**245.3**	**585.6**	**678.7**	**81.7**	**94.6**	**98.7**	**12.3 / 20.8**

比 LIMAP 和 CLMAP 的轨迹更多。

图 9 展示了三维线映射的定性比较。我们选取了 3 个场景中由 3 种不同直线检测器生成的重建直线图进行说明。与其他方法相比，我们的方法能够重建出更清晰、更完整的线结构。带深度的LIMAP也优于 LIMAP 和 CLMAP。然而，带深度的LIMAP仍有可能重建出不合理的线，因为它可能会根据检测到的二维直线以及不正确的深度图，拟合出不可靠的三维直线，如图 9 中的红色方框所示。这类情况会增大带深度的LIMAP的重建误差。

4.4 消融实验

4.4.1 混合损失的设计

我们主要在Hypersim数据集 [17] 的前十个场景上进行消融实验，以验证

分割	行级指标 M1						行轨迹指标 M2						
阈值	ACC-L↓	COMP-L↓	PREC-L↑	RECAL-L↑	F-SCORE-L↑	线条数↑	R1	R5	R10	P1	P5	P10	支持数量
0.02	0.0083	0.5031	0.9899	0.3372	0.4933	8232	549.0	1336.0	2033.3	82.0	94.1	97.4	10.5 / 11.9
0.1	0.0079	0.5223	0.9862	0.3214	0.4766	4053	413.4	987.9	1364.1	78.8	91.5	95.7	11.9 / 16.5
0.2	0.0076	0.5269	0.9924	0.3387	0.4963	3355	396.3	886.5	1010.5	82.2	95.1	98.7	13.1 / 21.9
1.0	0.0104	0.6757	0.9284	0.2189	0.3442	1145	69.9	190.9	223.2	68.7	82.7	88.6	11.7 / 13.5
2.0	0.0118	0.5753	0.9127	0.1919	0.3066	888	51.2	147.1	173.2	67.1	81.5	87.9	11.6 / 13.4
10.0	0.0164	0.6148	0.8907	0.1740	0.2775	744	40.4	114.4	137.8	66.3	81.0	86.9	11.4 / 13.1

表10：Hypersim 上使用不同数量初始平面的重建定量结果 [17]。

初始	行级指标 M1						行轨迹指标 M2						
数量	ACC-L↓	COMP-L↓	PREC-L↑	RECAL-L↑	F-SCORE-L↑	线条数↑	R1	R5	R10	P1	P5	P10	支持数量
200	0.0098	0.9547	0.9284	0.2323	0.3571	2265	223.3	622.8	876.0	80.6	93.2	95.5	12.4 / 18.7
2000	0.0076	0.5269	0.9924	0.3387	0.4963	3355	396.3	886.5	1010.5	82.2	95.1	98.7	13.1 / 21.9
5000	0.0074	0.5685	0.9738	0.3085	0.4604	3337	368.2	873.2	940.1	81.6	94.9	98.1	12.9 / 21.6
10000	0.0072	0.5598	0.9746	0.3169	0.4701	3312	343.5	791.8	959.2	82.6	94.2	97.5	12.3 / 20.5
20000	0.0073	0.5471	0.9743	0.3232	0.4776	3614	377.6	802.3	969.4	82.0	94.7	98.3	11.9 / 19.7
40000	0.0072	0.5331	0.9948	0.3401	0.4988	4016	420.8	954.1	1213.6	83.3	95.7	98.9	12.8 / 18.4
100000	0.0071	0.5321	0.9956	0.3471	0.5068	4660	417.7	978.5	1343.0	82.6	95.2	98.9	12.6 / 18.2

表 11：定量结果在 Hypersim [17] 上使用不同损失权重组合进行重建的定量结果。

损失权重		行级指标 M1						行轨迹指标 M2						
α_{Π}	α_L	ACC-L↓	COMP-L↓	PREC-L↑	RECAL-L↑	F-SCORE-L↑	线条数↑	R1	R5	R10	P1	P5	P10	支持数量
0.1	0.1	0.0146	0.5882	0.9428	0.2453	0.3804	1348	74.3	155.0	177.3	86.7	96.5	98.5	10.4 / 12.2
1	0.1	0.0127	0.5188	0.9576	0.2098	0.3282	842	51.9	106.3	121.4	88.4	96.9	98.9	10.7 / 12.2
10	0.1	0.0076	0.5269	0.9924	0.3387	0.4963	3355	396.3	886.5	1010.5	82.2	95.1	98.7	13.1 / 21.9
10	1	0.0098	0.5421	0.9784	0.2791	0.4239	1876	109.7	218.8	246.6	87.6	96.9	99.1	11.2 / 13.0
10	10	0.0078	0.5749	0.9901	0.3415	0.5003	6141	678.6	1487.4	1842.8	84.2	96.2	98.6	10.2 / 15.8

我们混合损失各项设计的有效性，并通过受控对比加以检验。勾选标记表示使用了所设计的损失，而未采用任何损失项的情形则表示使用PlanarSplatting [8]的基线方法。指标M1下的对比结果见表5。可以看出，我们的损失设计有助于重建更多数量的三维线段，并显著降低整体完整性误差。完整性的提升直接体现为更高的召回率和F1分数。由于我们的方法建立在PlanarSplatting的基线之上，并利用深度图监督平面与线的位置，因此在重建误差和精确率方面取得了相当的性能。尽管如此，与基线相比，我们的方法在三维线映射上展现出更高的准确度。

我们还在表 6 中报告了评估指标 M2 下的对比结果。与表 5 中的结论一致，所有消融实验都取得了相近的重建精确率。然而，与 PlanarSplatting 的原始基线相比，每个提出的损失分量都显著提高了构建的线轨迹的长度召回率，并能够跟踪更长的图像轨迹和线段轨迹。此外，组合不同的损失函数可进一步提高长度召回率，并增加受支持的线段数量。以上消融结果验证了所提出的混合损失设计的有效性。

4.4.2 2D 线到 *3D* 平面的边缘分配设计

在第 3.2 节中，我们的分配策略同时采用投影平面边缘与检测到的线段之间的夹角，以及投影平面边缘端点到检测到的直线的垂直距离这两个互补指标，以确定最佳对应关系。在此，我们分别仅考虑夹角或仅考虑端点垂直距离进行消融实验，以验证结合这两种策略的必要性。未勾选任何选项表示使用原始 PlanarSplatting [8]。我们仅在逐场景优化过程中采用这些不同策略，最终仍采用标准的三维线分配步骤（如第 3.4 节所述）。指标 M1 的定量结果见表 7，指标 M2 的定量结果见表 8。结果表明，每种单独的分配策略都能提高最终三维线重建的准确性和完整性，同时促成长线轨迹的形成。结合角度一致性与几何邻近性的联合策略取得了最佳性能。图 10 展示了定性比较。可以看出，仅使用角度或仅使用距离构建对应关系，均能提升三维线重建的质量。我们通过同时考虑角度和空间距离关系来搜索二维直线与三维边之间的对应关系，从而得到最完整、最准确的三维线图。

4.4.3 对启发式方法和超参数的敏感性

我们在标准Hypersim数据集 [17]上，对我们的方法中使用的若干关键超参数和启发式方法进行了消融研究。所有结果均报告了M1和M2两项指标。不同平面分割阈值对最终线映射影响的定量结果见表9。可以看出，平面分裂阈值会直接影响最终线段数量，过大或过小的阈值都会降低重建质量。初始平面数量对最终线映射影响的定量结果报告于

表 12：7Scenes [18] 上视觉定位的逐场景结果。我们报告平移误差中位数（单位：厘米）、旋转误差中位数（单位：度），以及在 5 厘米/5 度阈值下的位姿精度。

	基于匹配的方法				基于学习的方法		
场景	Hloc^{point} [59]	PtLine$^{point\ \&\ line}$ [60]	Limap$^{point\ \&\ 线}$[1]	**我们的方法$^{point\ \&\ 线}$**	Pl2Map^{point} [61]	Pl2Map$^{point\ \&\ 线}$[61]	**我们的方法$^{point\ \&\ 线}$**
国际象棋	2.4 / 0.84 / 93.0	2.4 / 0.85 / 92.7	2.5 / 0.85 / 92.3	**2.3 / 0.82 / 93.2**	2.0 / 0.65 / 95.5	**1.9** / **0.63** / 96.0	**1.9** / **0.63** / **96.2**
Fire	2.3 / 0.89 / 88.9	2.3 / 0.91 / 87.9	2.1 / 0.84 / **95.5**	**2.0 / 0.83** / 93.4	2.0 / 0.81 / 93.3	**1.9** / **0.80** / **94.0**	**1.9** / **0.80** / 93.8
人头	1.1 / 0.75 / 95.9	1.2 / 0.81 / 95.2	1.1 / 0.76 / **95.9**	**1.0 / 0.74** / 95.4	1.2 / 0.74 / 97.8	**1.1** / **0.71** / 98.2	**1.1** / **0.71** / **98.5**
办公室	3.1 / 0.91 / 77.0	3.2 / 0.96 / 74.5	**3.0 / 0.89** / 78.4	**3.0** / 0.90 / **78.6**	2.8 / 0.78 / 82.3	**2.7** / 0.74 / **84.3**	**2.7** / **0.73** / 83.9
南瓜	5.0 / 1.32 / 50.4	5.1 / 1.35 / 49.0	**4.7 / 1.23** / **52.9**	**4.7**/ 1.24 / 52.7	3.5 / 0.96 / 63.1	**3.4** / 0.93 / 64.1	3.5 / **0.92** / **64.3**
红色厨房	4.2 / 1.39 / 58.9	4.3 / 1.42 / 58.0	**4.1**/ 1.39 / 60.2	**4.1 / 1.36 / 60.7**	3.8 / 1.13 / 66.7	**3.7** / **1.10** / **68.9**	**3.7** / **1.10** / 68.5
楼梯	5.2 / 1.46 / 46.8	4.8 / 1.33 / 51.9	**3.7** / 1.02 / **71.1**	3.9 / **0.97** / 65.9	8.5 / 2.4 / 27.8	7.6 / **2.0** / 33.3	**7.4** / 2.1 / **34.1**

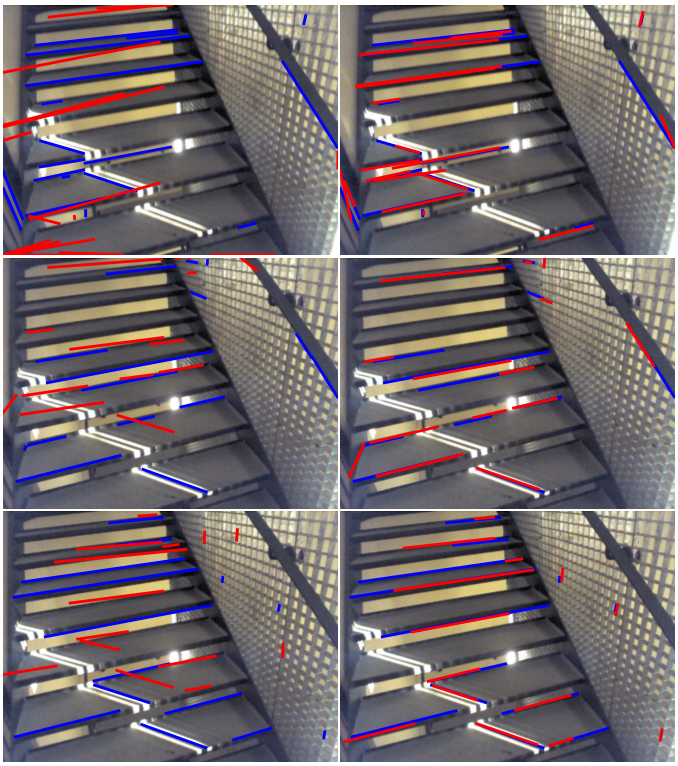


图 11：基于**Hloc** [59]的 7Scenes [18] 楼梯场景线辅助视觉定位。**蓝色**：检测到的二维线。**红色**：来自我们的三维线地图的投影二维线。**左**：由基于点估计的位姿得到的投影。**右**：由我们的线辅助估计位姿得到的投影。

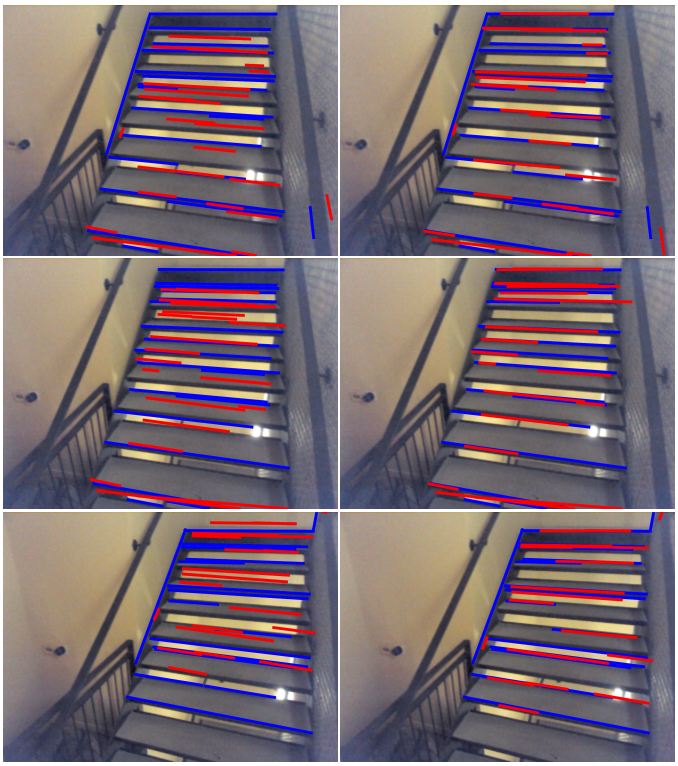


图 12：基于**Pl2Map** [61]的 7Scenes [18]楼梯场景线辅助视觉定位。**蓝色**：检测到的二维线。**红色**：来自我们的三维线地图的投影二维线。**左**：由基于点估计的位姿得到的投影。**右**：由我们的线辅助估计位姿得到的投影。

表10。可以看出，初始平面过少会对最终线段数量和重建质量产生显著影响。不同损失权重设置对最终线映射影响的定量结果见表 11。可以看出，两种损失权重都会影响最终线段数量，但重建质量受平面损失权重的影响更大。

4.5 线辅助视觉定位

本节将在 7Scenes 数据集上使用点线特征，展示我们在线辅助视觉定位方面的实验结果 [18]。我们采用经典的 LSD [12] 检测器来检测二维直线，以构建三维线地图。我们使用数据集提供的真值深度和位姿，以及由预训练 Omnidata [52] 模型预测的法线图进行三维重建。我们的视觉定位流程的输入是一组 2D-3D 点对应关系（来自基于点的方法，例如基于匹配的SfM模型或基于学习的模型）和 2D-3D 线对应关系（来自我们的 LiP-Map）。对于基于匹配的SfM模型，我们使用由 COLMAP [21] 中的点三角化器处理、并结合 HLoc [59]中的 SuperPoint [62] 获得的 2D-3D 点对应关系。对于基于学习的模型，我们采用来自 Pl2Map [61] 的预训练模型来获得2D–3D点对应关系。

对于相机位姿未知的查询图像，我们首先使用 PoseLib [63] 根据 2D-3D 点对应关系估计初始位姿。然后，我们使用初始位姿查询每张图像，从重建得到的三维线地图中生成 2D-3D 线对应关系（如第 3.2 节和第 3.4 节所述）。最后，我们使用 PoseLib [63] 估计优化后的

表13：在 Hypersim数据集 [17]上对时间和数量平面的分析。所有运行时间均以秒（s）为单位报告。

Hypersim	平面时间（/迭代）			直线时间（/迭代）			汇总（60 训练轮次）			
数据集	光栅化器	Loss	总计	分配	Loss	总计	平均值	总计	初始平面数	最终平面数
ai 001 001	0.0036	0.0029	0.0067	0.0221	0.0009	0.0241	5.3799	322.7921	2000	12900
人工智能 001 002	0.0036	0.0029	0.0066	0.0206	0.0010	0.0226	5.1115	306.6906	1955	7448
人工智能 001 003	0.0032	0.0028	0.0061	0.0192	0.0009	0.0211	4.9280	295.6779	1640	6337
人工智能 001 004	0.0041	0.0036	0.0078	0.0263	0.0010	0.0285	6.2292	373.7514	1897	19704
人工智能 001 005	0.0034	0.0031	0.0067	0.0223	0.0009	0.0242	5.7705	346.2297	1834	13549
人工智能 001 006	0.0040	0.0035	0.0076	0.0230	0.0010	0.0252	5.9670	358.0211	1767	6384
ai 001 007	0.0038	0.0032	0.0072	0.0224	0.0010	0.0245	5.6613	339.6774	1954	6888
ai 001 008	0.0036	0.0033	0.0070	0.0211	0.0010	0.0231	5.3834	323.0037	1794	6608
ai 001 009	0.0037	0.0029	0.0067	0.0214	0.0010	0.0235	5.6190	337.1397	1946	7105
ai 001 010	0.0033	0.0034	0.0068	0.0222	0.0009	0.0240	5.4591	327.5479	1943	13718
平均值	0.0036	0.0032	0.0069	0.0221	0.0010	0.0241	5.5509	333.0532	1873	10064

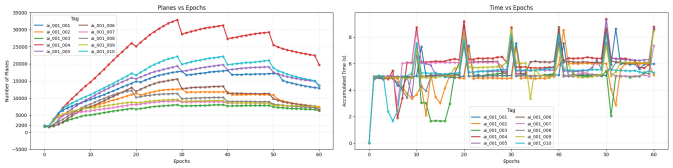


图13：Hypersim数据集 [17] 上的时间和平面数量分析。我们报告了平面数量和优化时间随优化轮次的变化情况。

通过 2D-3D 点对应关系和 2D-3D 线对应关系获取位姿。

各场景的定量结果汇总于表 12。我们报告了每个场景所有测试图像中估计位姿的平移误差中位数（单位：厘米）、旋转误差中位数（单位：度）以及位姿精度（定义为误差低于 5 厘米 / 5 度的位姿所占百分比）。如表所示，加入我们重建的三维线地图可显著降低位姿估计误差，并提高基于点的方法的准确度。与其他线辅助点线联合位姿估计方法相比，我们的 LiP-Map 也取得了极具竞争力的表现。

为清楚展示我们重建的三维线地图对视觉定位带来的提升，我们在图 11 和图 12 中展示了 7Scenes [18] 中最具挑战性的“楼梯”场景的更多细节。我们首先使用真实值（GT）位姿，从重建模型中检索对应的一组三维直线，然后分别使用基于点的方法估计的位姿（左图）和线辅助位姿（右图），将这些三维直线投影到图像平面上。基于点的方法估计的位姿通常会将三维直线投影到图像中的偏移位置；而通过我们的线辅助流程优化后的位姿能够实现明显更准确的投影，从而与观测到的场景结构具有更好的视觉一致性。

4.6 效率与鲁棒性分析

本节分析了我们的优化框架在 Hypersim 的前十个场景上的效率与鲁棒性 [17], , 使用 DeepLSD [11] 检测器，并在一台配备 48GB 内存的 NVIDIA RTX A6000 上运行。

逐场景优化的详细信息。我们首先将每个场景的初始平面基元数量固定为 2000，使用 100 个输入视图，并进行 60 轮训练。我们详细列出了两个核心模块在每次迭代中的平均运行时间开销，并报告了

表14：优化时间和平面数量随输入视图数量的变化。

输入视图	2	5	10	20	40	60	80	100
迭代次数	1000	1000	2000	2000	3000	3000	4000	5000
时间（秒）	32.6058	35.3922	101.3842	104.9645	165.1081	162.7495	221.6956	275.3702
初始平面数	1982	1947	1929	1925	1865	1890	1874	1875
最终平面数	2011	3065	4571	6102	7622	7747	8018	8676

表15：优化时间和平面数量随初始平面基元数量的变化。

数量	200	2000	5000	10000	20000	40000	100000
迭代次数	6000	6000	6000	6000	6000	6000	6000
时间（秒）	331.9121	333.0532	345.8118	346.6410	356.7424	391.7404	443.6078
初始平面数	184	1873	4811	9676	19315	38399	95517
最终平面数	7692	10064	11677	12579	14810	18940	27961

表13中各场景的优化总运行时间和每轮运行时间。表中“初始平面数量”表示初始裁剪步骤后剩余的平面基元数量，该步骤会移除在任何视点下都不可见的平面。“最终平面数”表示优化收敛后的平面基元数量。图13报告了各场景每轮训练的平面数量和运行时间。

输入视图数量的可扩展性。我们首先研究优化时间和平面数量如何随输入视图数量变化，结果见表14。需要注意的是，当输入视图过于稀疏时（例如仅有2个视图），60个训练轮次（120次迭代）不足以完成重建，因此我们会运行更多迭代（例如1000次）以获得高质量结果。实际运行时间主要取决于优化迭代次数，也会受到平面基元数量的一定影响。图14以“ai 001 001”场景为例展示了定性比较结果。可以看出，对于不同数量的输入视图，我们的方法都能稳健地完成重建。

– –

平面数量的可扩展性。我们还研究优化时间和平面数量如何随平面基元初始数量变化。表15显示，即使初始平面数量大幅增加，优化时间也不会急剧增长，而且收敛后的最终平面数量仍保持在相对稳定的范围内。图15以“ai 001 007”场景为例展示了定性比较结果。这些结果表明，我们的方法对于初始化时使用的平面数量具有稳健性。值得注意的是，虽然使用200个平面基元进行初始化会导致初始平面网格质量较差，但优化最终仍能收敛到相对完整的重建结果。

4.7 泛化到非平面结构

为了进一步研究我们的方法在通用非平面结构上的泛化能力，我们仅使用一些常见数据集中的场景多视图图像进行了实验，包括 DTU [64], Tanks and Temples [19], Blender MVS [65], 和 MipNerf360 [66]。为了尽可能全面地探索这一能力，我们使用 VGGT 预测的噪声位姿、深度图和法线图进行优化。如图 16 所示，这些场景

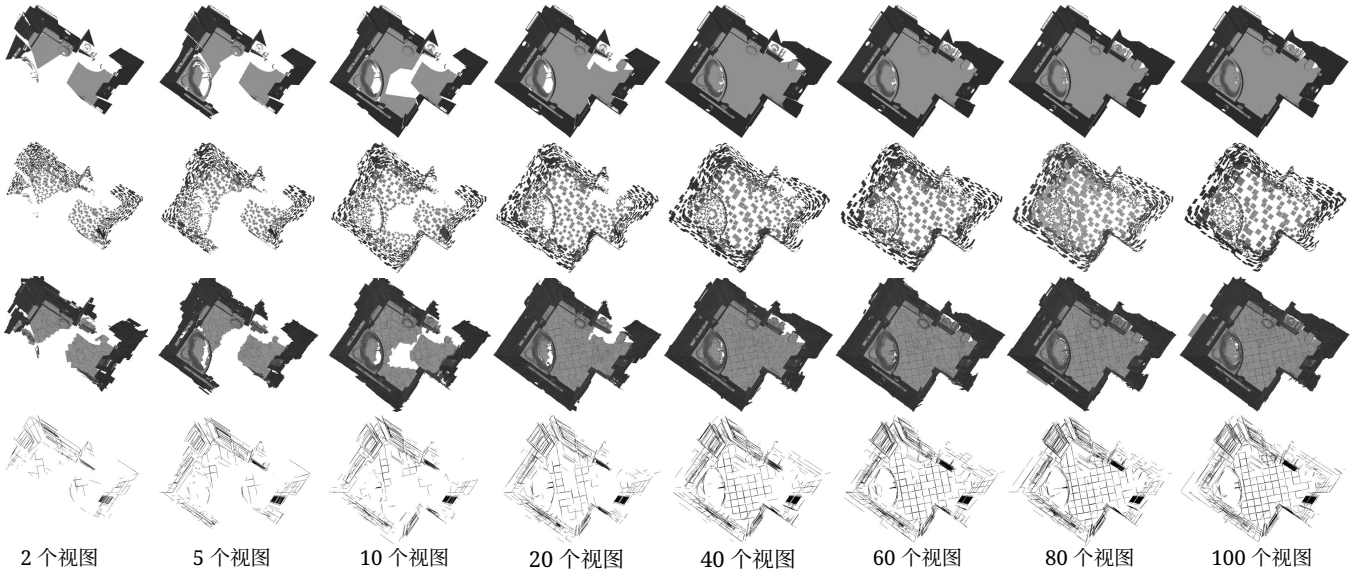


图 14：不同输入视图数量下结果的定性比较。第一行：初始网格。第二行：初始平面网格。第三行：最终平面网格。最后一行：最终三维线条。

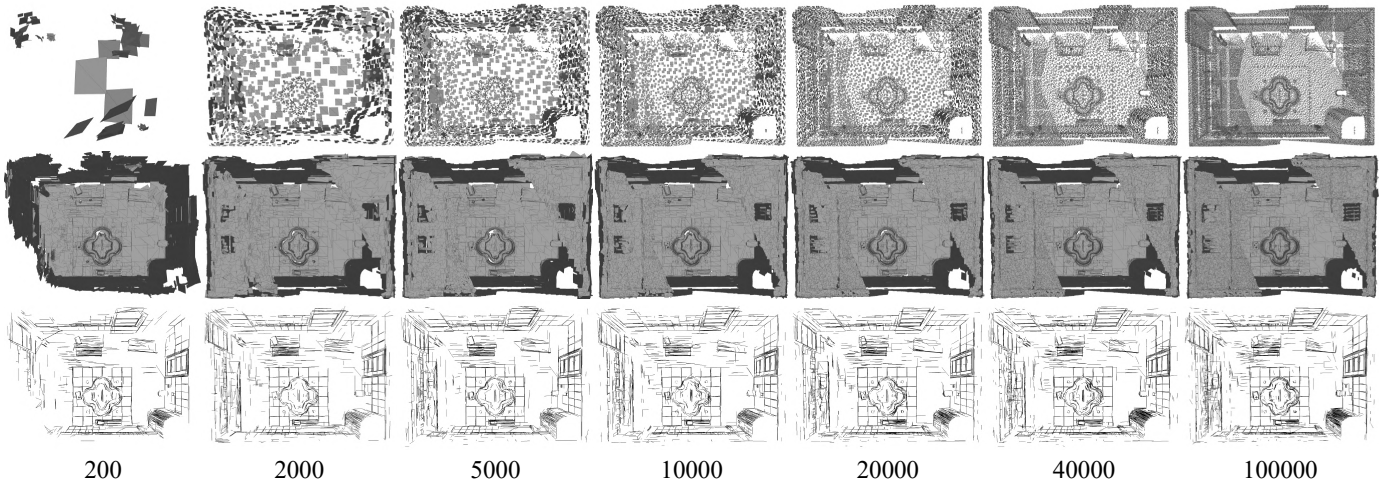


图 15：不同初始平面基元数量下结果的定性比较。第一行：初始平面网格。第二行：最终平面网格。最后一行：最终三维线条。

包含曲面椅子、圆桌、树木、细长的围栏栏杆、路灯、雕塑等非平面和不规则结构，以及其他几何结构复杂的物体。但我们的方法通过组织和优化大量平面之间的拓扑关系，基本能够满足这些场景的重建需求。尽管重建出的平面表面中有些平面可能不够准确，但分配策略中的严格阈值（如第 3.4 节所述）确保了二维直线与三维直线之间具有较强的几何一致性，从而能够过滤掉这些不准确的平面边缘。

4.8 使用 VGGT 的更多结果

我们还在 7Scenes 数据集 [18] 和 Tanks and Temples 数据集 [19]上进行了视觉实验。不同于使用真实姿态和深度图根据数据集提供的信息，我们采用最新的 VGGT [20]来回归位姿、深度图和法线图，以进行优化。在位姿和深度图不完美的情况下重建三维直线，这是一种更具挑战性的设置。由于我们的优化过程中所有超参数均具有尺度不变性，因此在集成 VGGT 时无需进行修改，因为整体框架在尺度变化下仍然兼容。我们在图 17 中展示了与 Pl2Map [61] 的重建结果视觉对比。即使使用 VGGT 的输出，我们的方法也能获得更加完整、准确的三维线图。我们还在图 18 中展示了我们在 7Scenes 数据集 [18] 和 Tanks and Temples 数据集 [19] 上的三维线图的其他结果。

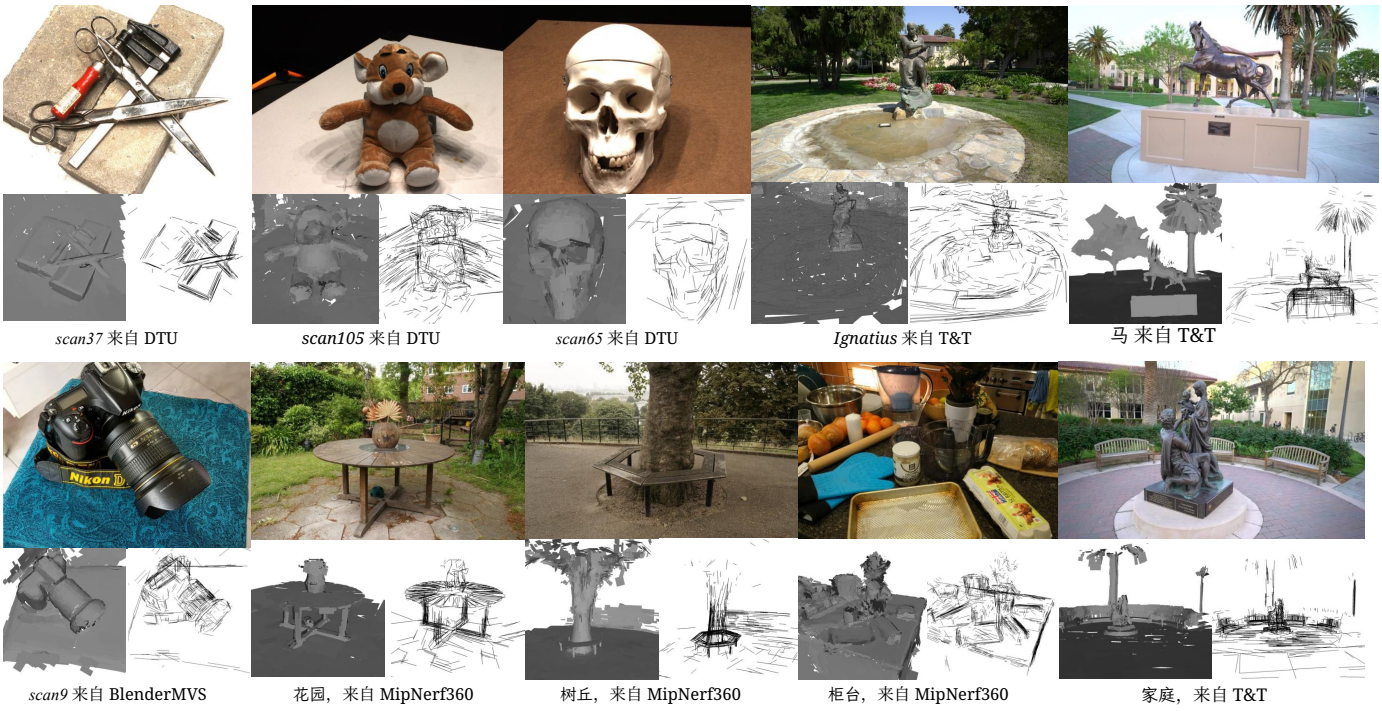


图 16：在非平面结构场景中重建的平面网格和三维线图的定性结果。

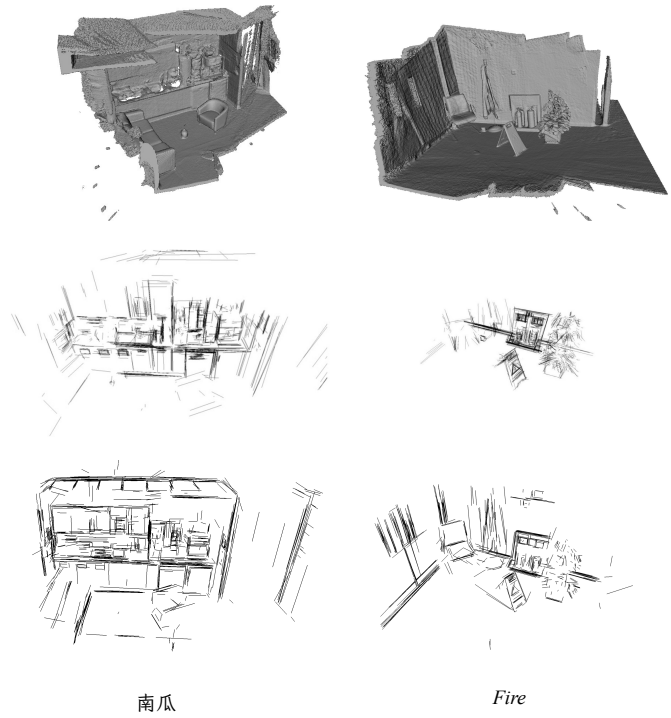


图 17：与 Pl2Map [61] 在 7scenes [18]上的定性比较。上：真实网格。中：使用真实值的 Pl2Map。下：我们使用 VGGT 得到的结果。

5 结论

在本文中，我们提出了*LiP-Map*，一种用于三维线图构建的线-平面联合优化框架。我们的方法利用线段的二维原始草图和 2.5D 从位姿多视图图像中获取深度和法线表示的草图，以优化三维平面与三维线段之间的几何协同。通过在线优化过程，我们同时实现了高性能的三维线映射，以及具有显著拓扑结构的三维平面重建。我们提出的*LiP-Map*能够生成准确、完整且细致的三维线映射，并可无缝适配任何二维线段检测器，包括经典方法（如 LSD）和基于学习的方法。通过绕过具有挑战性的二维线段匹配步骤，我们的方法在真实世界和合成室内场景基准测试上均显著优于当前最先进的方法。我们希望并相信，我们的结构化三维重建方法将有助于构建紧凑的空间表示，塑造不久的将来空间智能的新时代。

参考文献

[1] S. Liu、Y. Yu、R. Pautrat、M. Pollefeys 和 V. Larsson，《三维线映射再探》，发表于*IEEE*计算机视觉与模式识别会议 IEEE，2023，21 445 21 455 1, 2, 6, 7, 8, 9, 11, 20 页码 – 。L. Li、S. Peng、[2] Z. Yu、S. Liu、R. Pautrat、X. Yin 和 M. Pollefeys，《三维神经边缘重建》，发表于*IEEE*计算机视觉与模式识别会议 IEEE，2024，21 219 21 229 1, 2 页码 – 。M. Hofer、M. Maurer 和 [3] H. Bischof，《使用线段进行高效三维场景抽象》，发表于《计算机视觉与图像理解》，第 157 卷，页码 167 178, 2017 1, 2 – 。N. Xue、B. Tan、Y. Xiao、L. [4] Dong、G. Xia、T. Wu 和 Y. Shen，《NEAT：从神经吸引场中提炼三维线框》，发表于*IEEE*计算机视觉与模式识别会议 IEEE，2024 年，页码 19 968–19 977 1, 2, 7 。Z. Yu、J. Zheng、D. Lian、Z. Zhou 和 S. Gao，《通过关联嵌 [5] 入实现单幅图像逐片平面三维重建》，发表于*IEEE*计算机视觉与模式识别会议，2019，1029 1037 1, 2 页码 – 。B. Tan、[6] N. Xue、S. Bai、T. Wu 和 G. Xia，《Planetr：用于三维平面恢复的结构引导变换器》，发表于国际计算机视觉会议，2021 年，页码 4166–4175。1、2

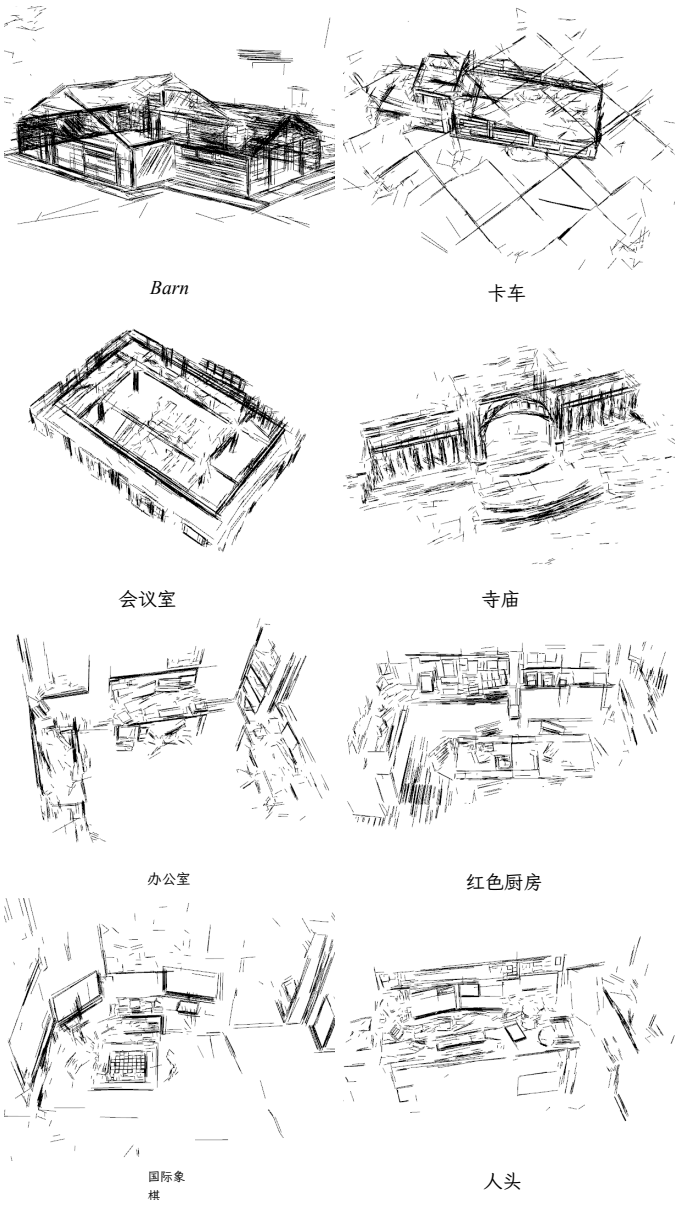


图 18：我们在 Tanks and Temples [19] 和 7scene [18]上的三维线图定性结果。“谷仓”、“卡车”、“会议室”和“寺庙”来自 Tanks and Temples 数据集。“办公室”、“红色厨房”、“国际象棋”和“人头”来自 7scenes 数据集。

[7] W. Ma、B. Tan、N. Xue、T. Wu、X. Zheng 和 G. Xia，《How-3d：从单幅图像进行整体三维线框感知》，发表于国际会议 3 2022, 596 605 1 *D Vis.*，页码 – 。B. Tan、R. Yu、Y. Shen 和 N. Xue，《Planarsplatting：3 分钟内实现精准的平面表面重建》，发表于*IEEE*计算机视觉与模式识别会议，2025 1, 2, 3, 5, 6, 7, 10, 21, 22 。L. Zhang 和 R. Koch，《一种基于 LBD描述子 和成对几何一致性的高效稳健线段匹配方法》，发表于《视觉通信与图像表示杂志》，第 24 卷，第 7 期，页码 794 805, 2013 1 – 。R. Pautrat、I. Suarez、Y. Yu、M. Pollefeys 和 V. Larsson，《Glue-Stick：通过将点和线结合实现稳健的图像匹配》，发表于国际计算机视觉会议，2023, 9706 9716 1, 2 页码 – 。R. Pautrat、D. Barath、V. Larsson、M. R. Oswald 和 M. Pollefeys，《DeepLSD：借助深度图像梯度进行线段检测与优化》，发表于*IEEE*计算机视觉与模式识别会议 IEEE，2023 年，页码 17 327–17 336。 [在线]。链接：https://doi.org/10.1109/CVPR52729.2023.01662 1、2、3、4、7、8、12

[12] R. G. von Gioi、J. Jakubowicz、J. Morel 和 G. Randall，《LSD：一种带误检控制的快速线段检测器》，IEEE模式分析与机器智能汇刊，第32卷，第4期，第722–732页，2010年。[13] 1, 3, 6, 8, 11 N. Xue、T. Wu、S. Bai、F. Wang、G. Xia、L. Zhang 和 P. H. S. Torr，《整体吸引式线框解析：从监督学习到自监督学习》，IEEE模式分析与机器智能汇刊，第 45, 12, 14 727 14 744, 2023 1, 2, 3, 4, 8 卷，第期，第–页。Z. Ke、[14] B. Tan、X. Zheng、Y. Shen、T. Wu 和 N. Xue，《ScaleLSD：简化流程的可扩展深度线段检测》，载于IEEE计算机视觉与模式识别会议，6月 2025, 6327 6336 1, 2, 3, 7, 8 ，第–页。A. [15] Dai、A. X. Chang、M. Savva、M. Halber、T. A. Funkhouser 和 M. Nießner，《ScanNet：带有丰富标注的室内场景三维重建》，载于IEEE计算机视觉与模式识别会议，2017年，第 2432 2443 2, 3, 6, 7, 8, 9, 21, 22, 23, 24, 30, 31 –页。C. [16] Yeshwanth、Y. Liu、M. Nießner 和 A. Dai，《ScanNet++：高保真的室内场景三维数据集》，载于国际计算机视觉会议， 2023, 12 22 2, 6, 7, 8, 22, 25 第–页。[17] M. Roberts、J. Ramapuram、A. Ranjan、A. Kumar、M. A. Bautista、N. Paczan、R. Webb 和 J. M. Susskind，《Hypersim：用于整体室内场景理解的照片级真实感合成数据集》，载于国际计算机视觉会议，2021年。2、4、7、8、9、10、12、20、21、22、23、26、[18] 32, 33, 34, 35 J. Shotton、B. Glocker、C. Zach、S. Izadi、A. Criminisi 和 A. Fitzgibbon，《用于RGB-D图像中相机重定位的场景坐标回归森林》，载于IEEE计算机视觉与模式识别会议，2013, 2930 2937 2, 11, 12, 13, 14, 15 第–页。A. Knapitsch、J. [19] Park、Q.-Y. Zhou 和 V. Koltun，《Tanks and Temples：大规模场景重建基准测试》，ACM图形学汇刊，第 36, 4, 2017 2, 12, 13, 15 卷，第期，7月。J. Wang、M. Chen、N. Karaev、A. Vedaldi、C. Rupprecht 和 D. [20] Novotny，《VGGT：视觉几何基础变换器》，载于IEEE计算机视觉与模式识别会议，2025 2, 13, 23 。J. L. Schonberger 和 J. [21] Frahm，《运动恢复结构再探》，载于IEEE计算机视觉与模式识别会议。IEEE计算机学会， 2016, 4104 4113 2, 11 第–页。D. Chekhlov、A. P. Gee、A. Calway 和 W. W. Mayol-Cuevas，《Ninja on a plane：利用视觉同步定位与建图自动发现物理平面以实现增强现实》，载于IEEE/ACM混合与增强现实国际研讨会，[22] 2007, 153 156 2 第–页。S. [23] Wang、V. Leroy、Y. Cabon、B. Chidlovskii 和 J. Revaud，《Dust3r：让几何三维视觉变得简单》，载于IEEE计算机视觉与模式识别会议，2024, 20 697 20 709 2 第–页。Z. Xue、Z. Yuan、J. [24] Wang、X. Wang、Y. Gao 和 H. Xu，《USEEK：用于泛化操作的无监督SE(3)等变三维关键点》，载于机器人与自动化国际会议，2023年，第1715–1722页。2[25] D. G. Lowe，《从尺度不变关键点中提取显著图像特征》，《国际计算机视觉期刊》，第 60, 2, 91 110, 2004 2 卷，第期，第–页。J. Morel 和 G. Yu，《ASIFT：一种用于完全仿射不变图像比较的新框架》，[26] SIAM《成像科学期刊》，第2卷，第2期，第 438 469, 2009 2 –页。P. Sarlin、D. DeTone、T. Malisiewicz 和 A. Rabinovich，《SuperGlue：利用图神经网络学习特征匹配》，载于IEEE计算机视觉与模式识别会议，[27] 2020, 4937 4946 2 第–页。H. Bay、A. Ess、T. Tuytelaars 和 L. V. Gool，《加速稳健特征（SURF）》，《计算机视觉与图像理解》，第110卷，第3期，第 346 359, 2008 2 –页。N. Xue、T. Wu、S. Bai、F. Wang、G. Xia、L. Zhang 和 P. H. S. Torr，《整体吸引式线框解析》，[28] [29] 载于IEEE计算机视觉与模式识别会议，2020, 2785 2794 2 第–页。R. Pautrat、J. Lin、V. Larsson、M. R. Oswald 和 M. Pollefeys，《SOLD2：自监督、具备遮挡感知能力的线描述与检测》，载于IEEE计算机视觉与模式识别会议，2021年，第11 368–11 378页。2[31] S. Huang、F. Qin、P. Xiong、N. Ding、Y. He 和 X. Liu，《TP-LSD：基于三点的线段检测器》，载于欧洲计算机视觉会议，第 12372, 2020, 770 785 2 卷，第–页。H. Li、K. Chen、J. Zhao、J. [30] [32] Wang、P. Kim、Z. Liu 和 Y. Liu，《学习识别球面上的正确二维-二维线对应关系》，载于IEEE计算机视觉与模式识别会议，2021, 11 743 11 752 2 第–页。D. [33] Wei、Y. Wan、Y. Zhang、X. Liu、B. Zhang 和 X. Wang，《ELSR：利用平面和点引导的高效线段重建》，载于IEEE计算机视觉与模式识别会议，2022年，第15 786–15 794页。2[34] X. Bai、H. Cui 和 S. Shen，《一致的三维线映射》，载于欧洲计算机视觉会议（ECCV），2024年。2、6、7、8

[35] H. Pfister、M. Zwicker、J. van Baar 和 M. H. Gross，《表面元素：作为渲染基元的表面元素》，载于ACM SIGGRAPH，2000，335 342 2 第–页。Y. Wang、F. Serena、S. Wu、C. ¨Oztireli 和 O. Sorkine-Hornung，《用于基于点的几何处理的可微表面泼溅》，ACM图形学汇刊，第38卷，第6期，第230:1–230:14页，2019年。2[37] B. Huang、Z. Yu、A. Chen、A. Geiger 和 S. Gao，《用于几何精确辐射场的二维高斯泼溅》，载于ACM SIGGRAPH会议论文集，A. Burbano、D. Zorin 和 W. Jarosz 编，2024，32 2 第页。[38] B. Tan、N. Xue、T. Wu 和 G. Xia，《NOPE-SAC：用于稀疏视图平面三维重建的神经单平面RANSAC》，IEEE模式分析与机器智能汇刊，第45，12，15233 15248, 2023 2 卷，第[39] 3期，第–页。C. Liu、K. Kim、J. Gu、Y. Furukawa 和 J. Kautz，《Planercnn：从单幅图像进行d平面检测与重建》，载于IEEE计算机视觉与模式识别会议，2019，4450 4459 2 第–页。J. Shi、S.[40] Zhi 和 K. Xu，《Planerectr：通过统一查询学习从单视图恢复三维平面》，载于国际计算机视觉会议，2023年，第9343 9352 2 –页。L. Jin、S. Qian、A. Owens 和 D. F. Fouhey，[41]《从稀疏视图进行平面表面重建》，载于国际计算机视觉会议，2021年，第12 971 12 980 2 –页。S. Agarwala、L. Jin、C. Rockwell 和 D. F. Fouhey，[42]《Planeformers：从稀疏视图平面到三维重建》，载于欧洲计算机视觉会议，第13663, 2022，192 209 2 卷，第–页。J. Liu、P. Ji、[43] N. Bansal、C. Cai、Q. Yan、X. Huang 和 Y. Xu，《Planemvs：从多视图立体视觉进行三维平面重建》，载于IEEE计算机视觉与模式识别会议，2022，8655 8665 2 第–页。T.[44] Shen、J. Gao、K. Yin、M. Liu 和 S. Fidler，《深度行进四面体：用于高分辨率三维形状合成的混合表示》，载于神经信息处理系统进展，2021年，第6087–6101页。2[45] Z. Chen、A. Tagliasacchi 和 H. Zhang，《Bsp-net：通过二叉空间划分生成紧凑网格》，载于IEEE计算机视觉与模式识别会议，2020，42 51 2 第–页。Z. Chen、A.[46] Tagliasacchi、T. A. Funkhouser 和 H. Zhang，《神经对偶轮廓》，ACM图形学汇刊，第41卷，第4期，第104:1–104:13页，2022年。2[47] T. Monnier、J. Austin、A. Kanazawa、A. A. Efros 和 M. Aubry，《可微积木世界：通过渲染基元进行定性三维分解》，载于神经信息处理系统进展，2023年。2[48] C. Gao、X. Wang、Q. Yu、L. Sheng、J. Zhang、X. Han、Y.-Z. Song 和 D. Xu，《通过视图相关深度采样从单幅草图进行三维重建》，IEEE模式分析与机器智能汇刊，第46卷，第12，9661 9676, 2024 2 期，第–页。L. Yariv、J. Gu、[49] Y. Kasten 和 Y. Lipman，《神经隐式表面的体渲染》，载于神经信息处理系统进展，2021年，第4805 4815 2 –页。Z. Yu、S. Peng、M. Niemeyer、[50] T. Sattler 和 A. Geiger，《Monosdf：探索用于神经隐式表面重建的单目几何线索》，载于神经信息处理系统进展，2022 2 。H. Guo、[51] S. Peng、H. Lin、Q. Wang、G. Zhang、H. Bao 和 X. Zhou，《基于曼哈顿世界假设的神经三维场景重建》，载于IEEE计算机视觉与模式识别会议。IEEE，2022年，第5501 5510 2 –页。A. Eftekhar、A. Sax、J. Malik 和 A.[52] Zamir，《Omnidata：从三维扫描构建多任务中层视觉数据集的可扩展流程》，载于国际计算机视觉会议，2021年，第10 766–10 776页。3、7、11、19、23、30[53] M. Hu、W. Yin、C. Zhang、Z. Cai、X. Long、H. Chen、K. Wang、G. Yu、C. Shen 和 S. Shen，《Metric3d v2：用于零样本度量深度和表面法线估计的通用单目几何基础模型》，CoRR，第abs/2404.15506卷，2024年。3、7、8、19、23[54] A. Paszke、S. Gross、F. Massa、A. Lerer、J. Bradbury、G. Chanan、T. Killeen、Z. Lin、N. Gimelshein、L. Antiga、A. Des maison、A. K¨opf、E. Z. Yang、Z. DeVito、M. Raison、A. Tejani、S. Chilamkurthy、B. Steiner、L. Fang、J. Bai 和 S. Chintala，《PyTorch：一种命令式风格、高性能的深度学习库》，载于神经信息处理系统进展，2019，8024 8035 6 第–页。D. P.[55] Kingma 和 J. Ba，《Adam：一种随机优化方法》，载于国际学习表征会议，2015 6 。J. Watson、F. Aleotti、M.[56] Sayed、Z. Qureshi、O. M. Aodha、G. J. Brostow、M. Firman 和 S. Vicente，《Airplanes：精确平面基于三维一致嵌入的估计，”载于*IEEE*计算机视觉与模式识别会议，2024，5270 5280 7 页码 – 。M. Sayed、J. Gibson、J.[57] Watson、V. Prisacariu、M. Firman 和 C. Godard，“SimpleRecon：无需三维卷积的三维重建”，载于欧洲计算机视觉会议，第13693, 2022，1 19 7 卷，页码 – 。A. M.[58] Araujo 和 M. M. Oliveira，“一种针对无组织点云进行平面检测的稳健统计方法”，《模式识别》，第100，107115, 2020 8 卷，第 页。P.-E. Sarlin、C.[59] Cadena、R. Siegwart 和 M. Dymczyk，“由粗到细：大规模稳健分层定位”，载于计算机视觉与模式识别会议，2019 11 。S. Gao、J. Wan、Y. Ping、X.[60] Zhang、S. Dong、Y. Yang、H. Ning、J. Li 和 Y. Guo，“通过联合优化视觉点与线进行位姿优化”，载于智能机器人与系统国际会议，2022，2888 2894 11 页码 – 。B.-T. Bui、H.-H.[61] Bui、D.-T. Tran 和 J.-H. Lee，“用于相机重定位的三维稀疏地图点与线表示”，载于智能机器人与系统国际会议，2024 11, 13, 14 。D. DeTone、T. Malisiewicz 和[62] A. Rabinovich，“SuperPoint：自监督兴趣点检测与描述”，载于*IEEE*计算机视觉与模式识别会议，2018，224 236 11 页码 – 。V.[63] Larsson 等人，“PoseLib——相机位姿估计的最小求解器”，2020年。 [在线]。可访问：https: //github.com/vlarsson/PoseLib
[64] H. Aanæs、R. R. Jensen、G. Vogiatzis、E. Tola 和 A. B. Dahl，“用于多视图立体视觉的大规模数据”，《国际计算机视觉期刊》，页码 1 16, 2016 12 – 。Y. Yao、Z. Luo、S. Li、J. Zhang、Y. Ren、L.[65] Zhou、T. Fang 和 L. Quan，“BlendedMVS：面向广义多视图立体网络的大规模数据集”，*IEEE*计算机视觉与模式识别会议，页码1787 1796, 2020 12 —- 。J. T. Barron、B. Mildenhall、D.[66] Verbin、P. P. Srinivasan 和 P. Hedman，“Mip-NeRF 360：无界抗锯齿神经辐射场”，载于*IEEE*计算机视觉与模式识别会议，2022年，页码 5470–5479。12[67] F. Pedregosa、G. Varoquaux、A. Gramfort、V. Michel、B. Thirion、O. Grisel、M. Blondel、P. Prettenhofer、R. Weiss、V. Dubourg、J. Vanderplas、A. Passos、D. Cournapeau、M. Brucher、M. Perrot 和 E. Duchesnay，“Scikit-learn：Python中的机器学习”，《机器学习研究杂志》，第12卷，页码2825–2830，2011年。20[68] R. Wang、S. Xu、Y. Dong、Y. Deng、J. Xiang、Z. Lv、G. Sun、X. Tong 和 J. Yang，“MoGe-2：具有度量尺度和清晰细节的精确单目几何”，载于神经信息处理系统进展，2025年。23

参考文献

[1] S. Liu、Y. Yu、R. Pautrat、M. Pollefeys 和 V. Larsson，《三维线映射再探》，载于 *IEEE*计算机视觉与模式识别会议 IEEE，2023，21 445 21 455 1, 2, 6, 7, 8, 9, 11, 20 第 – 页。L. Li、S. Peng、[2] Z. Yu、S. Liu、R. Pautrat、X. Yin 和 M. Pollefeys，《三维神经边缘重建》，载于 *IEEE*计算机视觉与模式识别会议 IEEE，2024，21 219 21 229 1, 2 第 – 页。M. Hofer、M. Maurer 和[3] H. Bischof，《使用线段进行高效的三维场景抽象》，刊于《计算机视觉与图像理解》，第157卷，第167 178, 2017 1, 2 –页。N. Xue、B. Tan、Y. Xiao、L. Dong、[4] G. Xia、T. Wu 和 Y. Shen，《NEAT：从神经吸引场中提炼三维线框》，载于*IEEE* 计算机视觉与模式识别会议 IEEE，2024年，第19 968– 19 977 1, 2, 7 页。Z.[5] Yu、J. Zheng、D. Lian、Z. Zhou 和 S. Gao，《通过关联嵌入实现单幅图像的分段平面三维重建》，载于*IEEE*计算机视觉与模式识别会议，2019，1029 1037 1, 2 第 – 页。B. Tan、[6] N. Xue、S. Bai、T. Wu 和 G. Xia，《Planetr：用于三维平面恢复的结构引导变换器》，载于国际计算机视觉会议，2021年，第4166 4175 1, 2 –页。W.[7] Ma、B. Tan、N. Xue、T. Wu、X. Zheng 和 G. Xia，《How-3d：从单幅图像进行整体三维线框感知》，载于国际 3 2022，596 605 1 视觉会议，第 – 页。[8] B. Tan、R. Yu、Y. Shen 和 N. Xue，《Planarsplatting：3分钟内实现精准的平面表面重建》，载于*IEEE*计算机视觉与模式识别会议，2025 1, 2, 3, 5, 6, 7, 10, 21, 22 。L. Zhang 和 R. Koch，《一[9] 种基于LBD描述子和成对几何一致性的高效稳健线段匹配方法》，刊于《视觉通信与图像表示杂志》，第24卷，第7期，第794–805页，2013年。1

[10] R. Pautrat、I. Su´arez、Y. Yu、M. Pollefeys 和 V. Larsson，《Glue-Stick：通过将点和线结合实现稳健图像匹配》，载于《国际计算机视觉会议》，2023，9706 9716 1, 2 第 – 页。R.
[11] Pautrat、D. Barath、V. Larsson、M. R. Oswald 和 M. Pollefeys，《DeepLSD：利用深度图像梯度进行线段检测与优化》，载于IEEE计算机视觉与模式识别会议，IEEE，2023年，第17 327–17 336页。 [在线]。链接：https://doi.org/10 1109 52729 2023 01662 1, 2, 3, 4, 7, 8, 12 。/CVPR。 .
[12] R. G. von Gioi、J. Jakubowicz、J. Morel 和 G. Randall，《LSD：一种带误检控制的快速线段检测器》，《IEEE模式分析与机器智能汇刊》，第32卷，第4期，第722–732页，2010年。1、3、6、8、9、11[13] N. Xue、T. Wu、S. Bai、F. Wang、G. Xia、L. Zhang 和 P. H. S. Torr，《整体吸引式线框解析：从监督学习到自监督学习》，《IEEE模式分析与机器智能汇刊》，第45，12，14727 14744, 2023 1, 2, 3, 4, 8 卷，第期，第 – 页。Z. Ke、
[14] B. Tan、X. Zheng、Y. Shen、T. Wu 和 N. Xue，《ScaleLSD：精简化的可扩展深度线段检测》，载于IEEE计算机视觉与模式识别会议，6月2025，6327 6336 1, 2, 3, 7, 8 ，第 – 页。
[15] A. Dai、A. X. Chang、M. Savva、M. Halber、T. A. Funkhouser 和 M. Nießner，《ScanNet：带丰富标注的室内场景三维重建》，载于IEEE计算机视觉与模式识别会议，2017年，第2432–2443页。2、3、6、7、8、9、21、22、23、24、30、31[16] C. Yeshwanth、Y. Liu、M. Nießner 和 A. Dai，《ScanNet++：高保真室内场景三维数据集》，载于《国际计算机视觉会议》，2023，12 22 2, 6, 7, 8, 22, 25 第 – 页。M. Roberts、J. Ramapuram、A. Ranjan、A. Kumar、M. A. Bautista、N. Paczan、R. Webb 和 J. M. Susskind，《Hypersim：用于整体室内场景理解的照片级真实感合成数据集》，载于《国际计算机视觉会议》，2021年。2、4、7、8、9、10、12、20、21、22、23、26、32、33、34、35[18] J. Shotton、B. Glocker、C. Zach、S. Izadi、A. Criminisi 和 A. Fitzgibbon，《利用场景坐标回归森林在RGB-D图像中进行相机重定位》，载于IEEE计算机视觉与模式识别会议，2013，2930 2937 2, 11, 12, 13, 14, 15 第 – 页。A. Knapitsch、J.
[19] Park、Q.-Y. Zhou 和 V. Koltun，《Tanks and Temples：大规模场景重建基准测试》，《ACM图形学汇刊》，第36，4，2017 2, 12, 13, 15 卷，7月。J. Wang、M. Chen、N. Karaev、A. Vedaldi、C. Rupprecht 和 D. Novotny，《VGGT：视觉几何基础变换器》，载于IEEE计算机视觉与模式识别会议，2025 2, 13, 23 。J. L. Schonberger 和 J.
[21] ¨Frahm，《运动恢复结构再探》，载于IEEE计算机视觉与模式识别会议，IEEE计算机学会，2016，4104 4113 2, 11 第 – 页。D. Chekhlov、A. P. Gee、A.
[22] Calway 和 W. W. Mayol-Cuevas，《Ninja on a plane：利用视觉同步定位与建图自动发现用于增强现实的物理平面》，载于IEEE/ACM混合与增强现实国际研讨会，2007，153 156 2 第 – 页。S.
[23] Wang、V. Leroy、Y. Cabon、B. Chidlovskii 和 J. Revaud，《Dust3r：让几何三维视觉变得简单》，载于IEEE计算机视觉与模式识别会议，2024，20 697 20 709 2 第 – 页。Z. Xue、Z. Yuan、J.
[24] Wang、X. Wang、Y. Gao 和 H. Xu，《USEEK：用于泛化操作的无监督SE(3)等变三维关键点》，载于机器人与自动化国际会议，2023年，第1715–1722页。2[25] D. G. Lowe，《尺度不变关键点的显著图像特征》，《国际计算机视觉期刊》，第60，2，91 110, 2004 2 卷，第期，第 – 页。
[26] J. Morel 和 G. Yu，《ASIFT：一种用于完全仿射不变图像比较的新框架》，《SIAM成像科学杂志》，第2卷，第2期，第438 469, 2009 2 –页。P. Sarlin、
[27] D. DeTone、T. Malisiewicz 和 A. Rabinovich，《SuperGlue：利用图神经网络学习特征匹配》，载于IEEE计算机视觉与模式识别会议，2020，4937 4946 2 第 – 页。H. Bay、
[28] A. Ess、T. Tuytelaars 和 L. V. Gool，《加速稳健特征（SURF）》，《计算机视觉与图像理解》，第110卷，第3期，第346–359页，2008年。2[29] N. Xue、T. Wu、S. Bai、F. Wang、G. Xia、L. Zhang 和 P. H. S. Torr，《整体吸引式线框解析》，载于IEEE计算机视觉与模式识别会议，2020，2785 2794 2 第 – 页。R. Pautrat、J. Lin、V.
[30] Larsson、M. R. Oswald 和 M. Pollefeys，《SOLD2：自监督的遮挡感知线描述与检测》，载于IEEE计算机视觉与模式识别会议，2021年，第11 368–11 378页。2[31] S. Huang、F. Qin、P. Xiong、N. Ding、Y. He 和 X. Liu，《TP-LSD：基于三点的线段检测器》，载于欧洲计算机视觉会议，第12372卷，2020年，第770–785页。2
[32] H. Li、K. Chen、J. Zhao、J. Wang、P. Kim、Z. Liu 和 Y. Liu，“学习在球面上识别正确的二维-二维线对应关系”，载于 IEEE计算机视觉与模式识别会议，2021，11 743 11 752 2 页码 –
[33] 。D. Wei 、 Y. Wan 、 Y. Zhang 、 X. Liu 、 B. Zhang 和 X. Wang，“ELSR：利用平面和点引导的高效线段重建”，载于 IEEE计算机视觉与模式识别会议，2022年，页码 15 786–15 794。2[34] X. Bai、H. Cui 和 S. Shen，“一致的三维线映射”，载于 欧洲计算机视觉会议（ECCV），2024 2, 6, 7, 8 。H. Pfister、
[35] M. Zwicker、J. van Baar 和 M. H. Gross，“表面元素：作为渲染基元的表面元素”，载于 ACM SIGGRAPH，2000，335 342 2 页码 – 。Y. Wang、F. Serena、
[36] S. Wu、C. ¨Oztireli 和 O. Sorkine-Hornung，“用于基于点的几何处理的可微表面泼溅”，《ACM图形学汇刊》，第 38 卷，第 6 期，页码 230:1–230:14，2019年。2[37] B. Huang、Z. Yu、A. Chen、A. Geiger 和 S. Gao，“用于几何精确辐射场的二维高斯泼溅”，载于 ACM SIGGRAPH会议论文集，A. Burbano、D. Zorin 和 W. Jarosz 编，2024，32 2 页码 。B. Tan、N. Xue、T. Wu 和 G. Xia，“
[38] NOPE-SAC：用于稀疏视图平面三维重建的神经单平面RANSAC”，《IEEE模式分析与机器智能汇刊》，第45，12，15233 15248, 2023 2 卷，第 期，
[39] 3页码 – 。C. Liu、K. Kim、J. Gu、Y. Furukawa 和 J. Kautz，“Planercnn：从单幅图像进行 d 平面检测与重建”，载于 IEEE计算机视觉与模式识别会议，2019，4450 4459 2 页码 – 。J. Shi、S.
[40] Zhi 和 K. Xu，“Planerectr：通过统一查询学习从单视图恢复三维平面”，载于 国际计算机视觉会议，2023年，页码 9343 9352 2 – 。L. Jin、S. Qian、A.
[41] Owens 和 D. F. Fouhey，“稀疏视图平面表面重建”，载于 国际计算机视觉会议，2021年，页码 12 971 12 980 2 – 。S. Agarwala、L. Jin、C. Rockwell 和
[42] D. F. Fouhey，“Planeformers：从稀疏视图平面到三维重建”，载于 欧洲计算机视觉会议，第13663, 2022，192 209 2 卷，页码 – 。J. Liu、
[43] P. Ji、N. Bansal、C. Cai、Q. Yan、X. Huang 和 Y. Xu，“Planemvs：基于多视图立体视觉的三维平面重建”，载于 IEEE计算机视觉与模式识别会议，2022，8655 8665 2 页码 – 。T.
[44] Shen、J. Gao、K. Yin、M. Liu 和 S. Fidler，“深度行进四面体：用于高分辨率三维形状合成的混合表示”，载于 神经信息处理系统进展，2021年，页码 6087–6101。2[45] Z. Chen、A. Tagliasacchi 和 H. Zhang，“Bsp-net：通过二叉空间划分生成紧凑网格”，载于 IEEE计算机视觉与模式识别会议，2020，42 51 2 页码 – 。Z. Chen、A.
[46] Tagliasacchi、T. A. Funkhouser 和 H. Zhang，“神经对偶轮廓”，《ACM图形学汇刊》，第 41 卷，第 4 期，页码 104:1–104 13, 2022 2 : 。T. Monnier、J.
[47] Austin、A. Kanazawa、A. A. Efros 和 M. Aubry，“可微积木世界：通过渲染基元进行定性三维分解”，载于 神经信息处理系统进展，2023年。2[48] C. Gao、X. Wang、Q. Yu、L. Sheng、J. Zhang、X. Han、Y.-Z. Song 和 D. Xu，“通过视图相关深度采样从单幅草图进行三维重建”，《IEEE模式分析与机器智能汇刊》，第 46 卷，第12，9661 9676, 2024 2 期，页码 – 。L. Yariv、J. Gu、Y. Kasten 和
[49] Y. Lipman，“神经隐式表面的体渲染”，载于 神经信息处理系统进展，2021年，页码 4805 4815 2 – 。Z. Yu 、 S. Peng 、 M. Niemeyer 、 T. Sattler 和 A.
[50] Geiger，“Monosdf：探索用于神经隐式表面重建的单目几何线索”，载于 神经信息处理系统进展，2022 2 。
[51] H. Guo、S. Peng、H. Lin、Q. Wang、G. Zhang、H. Bao 和 X. Zhou，“基于曼哈顿世界假设的神经三维场景重建”，载于 IEEE计算机视觉与模式识别会议。IEEE，2022年，页码 5501 5510 2 – 。A. Eftekhar、A. Sax、J. Malik 和 A.
[52] Zamir，“Omnidata：从三维扫描构建多任务中层视觉数据集的可扩展流程”，载于国际计算机视觉会议，2021年，页码 10 766–10 776。3, 7, 11,[53] 19, 29, 30 M.
Hu、W. Yin、C. Zhang、Z. Cai、X. Long、H. Chen、K. Wang、G. Yu、C. Shen 和 S. Shen，“Metric3d v2：用于零样本度量深度和表面法线估计的通用单目几何基础模型”，CoRR，第 abs/2404.15506 卷，2024年。3, 7, 8,[54] 19, 29 A. Paszke、
S. Gross、F. Massa、A. Lerer、J. Bradbury、G. Chanan、T. Killeen、Z. Lin、N. Gimelshein、L. Antiga、A. Desmaison，

A. K¨opf、E. Z. Yang、Z. DeVito、M. Raison、A. Tejani、S. Chil-amkurthy、B. Steiner、L. Fang、J. Bai 和 S. Chintala，“PyTorch：一种命令式风格的高性能深度学习库”，载于 神经信息处理系统进展，2019， 8024 8035 6 页码 – 。D. P. [55] Kingma 和 J. Ba，“Adam：一种随机优化方法”，载于 国际学习表征会议，2015 6 。J. Watson、F. Aleotti、M. [56] Sayed、Z. Qureshi、O. M. Aodha、G. J. Brostow、M. Firman 和 S. Vicente，“Airplanes：通过三维一致嵌入实现精确平面估计”，载于 *IEEE*计算机视觉与模式识别会议，2024， 5270 5280 7 页码 – 。M. Sayed、J. [57] Gibson、J. Watson、V. Prisacariu、M. Firman 和 C. Godard，“SimpleRecon：无需三维卷积的三维重建”，载于 欧洲计算机视觉会议，第 13693, 2022， 1 19 7 卷，页码 – 。A. M. [58] Araujo 和 M. M. Oliveira，“一种针对无组织点云平面检测的稳健统计方法”，《模式识别》，第 100， 107115, 2020 8 卷，页码 。P.-E. Sarlin、C. Cadena、[59] R. Siegwart 和 M. Dymczyk，“由粗到细：大规模稳健分层定位”，载于 计算机视觉与模式识别会议，2019 11 。S. Gao、J. Wan、Y. Ping、X. Zhang、S. [60] Dong、Y. Yang、H. Ning、J. Li 和 Y. Guo，“通过视觉点与线的联合优化进行位姿细化”，载于 智能机器人与系统国际会议，2022， 2888 2894 11 页码 – 。B.-T. Bui、H.-H. [61] Bui、D.-T. Tran 和 J.-H. Lee，“用于相机重定位的三维稀疏地图点与线表示”，载于 智能机器人与系统国际会议，2024 11, 13, 14 。D. DeTone、T. Malisiewicz 和 A. Rabinovich，“SuperPoint：自监督兴趣点检测与描述”，载于 *IEEE*计算机视觉与模式识别会议，2018， 224 236 11 页码 – 。V. [62] Larsson 等，“PoseLib - 相机位姿估计的最小求解器”，2020年。 [在线]。可通过以下网址获取：https: //github.com/vlarsson/PoseLib [63] H. Aanæs、R. R. Jensen、G. Vogiatzis、E. Tola 和 A. B. Dahl，“用于多视图立体视觉的大规模数据”，《国际计算机视觉期刊》，页码 1 16, 2016 12 – 。Y. Yao、Z. Luo、S. Li、J. Zhang、Y. Ren、L. [64] Zhou、T. Fang 和 L. Quan，“Blendedmvs：用于广义多视图立体网络的大型数据集”，《*IEEE*计算机视觉与模式识别会议》，页码 1787 1796, 2020 12 —- 。J. T. Barron 、 B. Mildenhall 、 D. Verbin 、 P. P. Srinivasan 和 P. Hedman，“Mip-Nerf 360：无界抗锯齿神经辐射场”，载于 *IEEE*计算机视觉与模式识别会议，2022年，页码 5470–5479。12[67] F. Pedregosa、G. Varoquaux、A. Gramfort、V. Michel、B. Thirion、O. Grisel、M. Blondel、P. Prettenhofer、R. Weiss、V. Dubourg、J. Vanderplas、A. Passos、D. Cournapeau、M. Brucher、M. Perrot [65] [66] 和 E. Duchesnay，“Scikit-learn：Python中的机器学习”，《机器学习研究杂志》，第 12 卷，页码 2825–2830，2011年。20[68] R. Wang、S. Xu、Y. Dong、Y. Deng、J. Xiang、Z. Lv、G. Sun、X. Tong 和 J. Yang，“MoGe-2：具有度量尺度和清晰细节的精确单目几何”，载于 神经信息处理系统进展，2025年。23

附录 A 更多实现细节

射线与平面求交。我们首先计算平面基元与从图像像素发出的射线之间的交点。具体而言，给定一条从相机中心$\mathbf{r} = \{\mathbf{o},\mathbf{d}\}$出发、方向为$\mathbf{o} \in \mathbb{R}^3$ $\mathbf{d} \in \mathbb{R}^3$的射线，其与一个平面基元$\pi$ 的交点$\mathbf{x}_\pi^\mathbf{r} \in \mathbb{R}^3$ 可计算如下:

$$\mathbf{x}_\pi^\mathbf{r} = \mathbf{o} + \frac{(\mathbf{p}_\pi - \mathbf{o}\cdot\mathbf{n}_\pi)}{\mathbf{d}\cdot\mathbf{n}_\pi}\mathbf{d}, \tag{18}$$

其中，$\mathbf{p}_\pi$ 和$\mathbf{n}_\pi$ 分别是平面基元 π的中心和法向量。

平面泼溅函数。获得射线与平面求交$\mathbf{x}_\pi^\mathbf{r}$后，我们使用平面泼溅函数计算其泼溅权重，该权重将用于渲染。

对于射线$\mathbf{r}$ 与平面基元 π的给定交点$\mathbf{x}_\pi^\mathbf{r}$ ，我们首先计算该交点到平面基元的 X轴和 Y轴的投影距离$\mathcal{P}_X, \mathcal{P}_Y \in \mathbb{R}$:

$$\mathcal{P}_X = (\mathbf{x}_\pi^\mathbf{r} - \mathbf{p}_\pi)\cdot\mathbf{v}_\pi^x, \quad \mathcal{P}_Y = (\mathbf{x}_\pi^\mathbf{r} - \mathbf{p}_\pi)\cdot\mathbf{v}_\pi^y. \tag{19}$$

然后，我们计算沿平面 X轴方向的泼溅权重 π :

$$w_X(\mathbf{x}_\pi^\mathbf{r}) = \begin{cases} 2\sigma(5\lambda(r_\pi^{x+} - |\mathcal{P}_X|)), & \text{if } \mathcal{P}_X > 0 \\ 2\sigma(5\lambda(r_\pi^{x-} - |\mathcal{P}_X|)), & \text{otherwise} \end{cases}, \tag{20}$$

其中， $\sigma(\cdot)$ 是Sigmoid 函数， λ 是用于控制泼溅权重的超参数。同理，我们继续计算沿平面 Y轴方向的泼溅权重π :

$$w_Y(\mathbf{x}_\pi^\mathbf{r}) = \begin{cases} 2\sigma(5\lambda(r_\pi^{y+} - |\mathcal{P}_Y|)), & \text{if } \mathcal{P}_Y > 0 \\ 2\sigma(5\lambda(r_\pi^{y-} - |\mathcal{P}_Y|)), & \text{otherwise} \end{cases}, \tag{21}$$

其中， $r_\pi^{x+}, r_\pi^{x-}, r_\pi^{y+}, r_\pi^{y-}$是平面 π的半径参数。优化过程中，λ的值通过指数函数逐渐增大，直至最大值 300，如下所示:

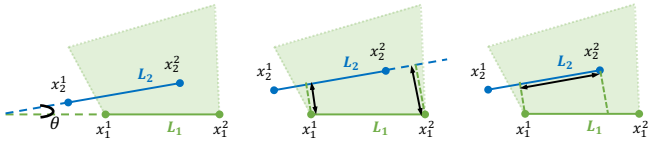
$$\lambda = min(20e^{(-(1-0.001*ite))}, 300), \tag{22}$$

其中 ite 表示优化过程中的迭代次数。最后，可计算最终的泼溅权重:

$$w(\mathbf{x}_\pi^\mathbf{r}) = \begin{cases} w_X, & \text{if } w_X < w_Y \\ w_Y, & \text{otherwise} \end{cases}. \tag{23}$$

平面混合合成。对于所有射线与平面的交点，我们滤除泼溅权重低于0.0001的交点，然后按深度由近及远对剩余交点排序。接着，选取每条射线最近的 M 个交点用于渲染（本文中为$M = 5$ ）。将射线$\mathbf{r}$ 的所选交点记为 $P_\mathbf{r} = \{\mathbf{x}_{\pi_{\tau(j)}}^\mathbf{r}\}_{j=1}^M$。其中， $\tau(j)$ 表示该平面在所有平面基元中的索引。最后，我们将某一图像$\mathbf{I}$的深度和法线图渲染为:

$$\mathbf{D}_{\text{render}}^\Pi(\mathbf{r}) = \sum_{j=1}^M T_j w(\mathbf{x}_{\pi_{\tau(j)}}^\mathbf{r}) t_j, \tag{24}$$



(a) 直线间的夹角 (b) 垂直距离 (c) 重叠率

图 19：几何一致性度量。我们提出了三种尺度不变度量，以确保三维平面边缘与二维检测线之间的几何一致性。

$$\mathbf{N}_{\text{render}}^\Pi(\mathbf{r}) = \sum_{j=1}^M T_j w(\mathbf{x}_{\pi_{\tau(j)}}^\mathbf{r}) \mathbf{n}_{\pi_{\tau(j)}}, \tag{25}$$

其中

$$T_j = \prod_{i=1}^{j-1}(1 - w(\mathbf{x}_{\pi_{\tau(i)}}^\mathbf{r})). \tag{26}$$

其中， t_j 是交点的深度，$\mathbf{n}^{\pi_{\tau(j)}}$ 是平面基元 $\pi_{\tau(j)}$的法向量。为了监督渲染后的深度和法线图，我们使用给定数据集的真实深度/法线图，或 Metric3Dv2 [53] 和 Omnidata [52] 预训练模型预测的深度/法线图作为伪标签。用于监督的深度图和法线图分别记为 $\mathbf{D}^{\text{pre}}(\mathbf{r})$ 和$\mathbf{N}^{\text{pre}}(\mathbf{r})$。最后，可计算渲染损失:

$$\begin{aligned} \mathcal{L}_{\text{render}}^\Pi = &\alpha_1 \sum_{\mathbf{r}\in\mathbf{I}} \|1 - \mathbf{N}_{\text{render}}^\Pi(\mathbf{r})^\top \mathbf{N}_{\text{pre}}(\mathbf{r}))\|_1 + \\ &\alpha_1 \sum_{\mathbf{r}\in\mathbf{I}} \|\mathbf{N}_{\text{render}}^\Pi(\mathbf{r}) - \mathbf{N}_{\text{pre}}(\mathbf{r}))\|_1 + \\ &\alpha_2 \sum_{\mathbf{r}\in\mathbf{I}} \|\mathbf{D}_{\text{render}}^\Pi(\mathbf{r}) - \mathbf{D}_{\text{pre}}(\mathbf{r})\|_1, \end{aligned} \tag{27}$$

其中， $\alpha_1 = 5.0$、 $\alpha_2 = 1.0$、 $\alpha_3 = 2.0$， $\mathbf{r}$ 是从图像 $\mathbf{I}$的像素发出的射线。

附录 B 几何一致性度量。

我们在论文中定义了一些几何约束，并应用这些约束来提取最终的三维线图并构建线轨迹。在此，我们介绍这些几何约束的计算方法，并在图 19 中给出一个示例以作说明。假设绿色区域由三维平面基元投影而来，(x_1^1, x_1^2) 是来自三维平面边缘的投影二维直线 L_1 的端点，(x_2^1, x_2^2) 是检测二维直线 L_2的端点。我们按如下方式计算三种几何度量:

- 投影二维线与每条检测二维直线之间的角距离 d_{ang}（记为 θ）。如图 19(a) 所示， θ 的计算方式为:

$$\theta = \arccos\left(\text{clamp}\left(\frac{\|(x_1^2 - x_1^1)\cdot(x_2^2 - x_2^1)\|}{\|(x_1^2 - x_1^1)\|}, -1, 1\right)\right). \tag{28}$$

第3.2节和第3.4节中使用的角距离也按此方式计算。

- 最大正交距离 d_{dist} 为投影二维直线端点到检测二维直线的距离。如图19(b)所示， d_{dist} 的计算方式为:

$$d_{\text{dist}} = \max\left(d_1, d_2\right), d_i = \frac{\left\|\left(x_1^i - x_2^1\right) \times \left(x_1^i - x_2^2\right)\right\|}{\left\|\left(x_2^2 - x_2^1\right)\right\|}. \tag{29}$$

第 3.4 节中定义的正交距离 d_{dist} 也以这种方式计算。

- 投影后的二维直线与检测二维直线之间的重叠率为 $d_{overlap}$。如图 19(C) 所示，$d_{overlap}$ 的计算方式如下:

$$d_{overlap} = \frac{\|L\|}{\|L_2\|}, \tag{30}$$

其中， L 是 L_1 在 L_2上的内部投影部分。第 3.4 节中使用的重叠率也以这种方式计算。

附录 C 评估指标的计算

给定真值模型（网格或点云）和重建三维直线，我们介绍如何计算广泛使用的标准指标 M1 和基于线轨迹的指标 M2。基本步骤是计算三维线段与模型之间的最近距离。为此，我们首先沿每条三维线段均匀采样点，然后在真值模型或这些采样点上构建 KD树，以高效计算采样点与真值模型之间的距离。

指标 M1。 为构建输入点 P_{pre}，对于线级指标，我们沿每条三维线段均匀采样 100 个点；对于交汇点级指标，我们收集每条三维线段的两个端点。给定真实值点云 P_{gt}，我们构建 KD树:

$$\begin{aligned} \text{KDT}_{pre} &= \text{KDTree}(P_{pre}), \\ \text{KDT}_{gt} &= \text{KDTree}(P_{gt}), \end{aligned} \tag{31}$$

其中，算法KDTree采用自 Scikit-learn 库 [67]。接着，我们使用点 P_{pre} 查询KDT^{gt}，并使用点云 P_{gt} 查询KDT^{pre}，以获得距离:

$$\begin{aligned} D_{pre} &= \text{KDT}_{gt}(P_{pre}), \\ D_{gt} &= \text{KDT}_{pre}(P_{gt}). \end{aligned} \tag{32}$$

准确度和完整性按如下方式计算:

$$\begin{aligned} \text{ACC} &= \frac{1}{n_1}\sum_{i=1}^{n_1} D_{gt}^i, \\ \text{COMP} &= \frac{1}{n_2}\sum_{i=1}^{n_2} D_{pre}^i, \end{aligned} \tag{33}$$

其中， n_1 是点组 P_{pre}中的点数， n_2 是点云 P_{gt}中的点数。给定距离阈值 τ_d（本文中为 0.05 m）后，精确率、召回率和 F1分数的计算方式如下:

$$\begin{aligned} \text{PREC} &= \frac{1}{n_1}\sum_{i=1}^{n_1} \mathbf{1}(D_{pre}^i <= \tau_d), \\ \text{RECALL} &= \frac{1}{n_2}\sum_{i=1}^{n_2} \mathbf{1}(D_{gt}^i <= \tau_d), \\ \text{F1} &= 2 \cdot \frac{\text{PREC} \cdot \text{RECALL}}{\text{PREC} + \text{RECALL}} \end{aligned} \tag{34}$$

其中，$\mathbf{1}(\cdot)$ 是指示函数。

指标 M2。 给定重建的三维直线，按照 LIMAP [1]，我们沿每条三维线段均匀采样 1000 个点，并构建 KD树以获得最近距离 D_{pre}。给定距离阈值 τ_d （本文中为 1/5/10 毫米或 5/10/50 毫米），计算每条三维线段上距离小于 τ_d 的点所占比例:

$$\text{ratios}_\text{i} = \frac{1}{n}\sum_{j=1}^{n} \mathbf{1}(D_{pre}^{(i,j)} <= \tau_d) \tag{35}$$

其中， $n = 1000$是每条线的采样点数量。计算距真实值模型 $\tau_d$mm 以内的线段长度总和，如下所示:

$$R_{\tau_d} = \sum_{i=1}^{m} \text{Len}_i \cdot \text{ratios}_\text{i}, \tag{36}$$

其中，Len_i 是第 i 条三维直线的长度。计算距真实值模型 τ_d mm 以内的轨迹百分比，如下所示:

$$P_{\tau_d} = 100 \cdot \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}(\text{ratios}_i > 0). \tag{37}$$

附录 D
全局线合并

我们的流程可能会继承二维线检测器带来的误差，这些误差通常表现为跨视图的碎片化检测结果、重复线假设或轻微的端点错位。为此，我们提出了一种轻量级的全局合并策略，在保持几何一致性的同时整合冗余的三维线段。

基线与局部合并： 以不进行合并的原始方法为基线，我们的局部合并策略将对应于同一条检测线的三维直线分组，并将它们合并为一个线段。

全局合并：我们的全局合并策略使用 DB-SCAN 对所有视图中与每条检测到的二维直线相关联的三维直线进行聚类，并在这些簇中传播唯一标识符以建立全局对应关系，再使用基于PCA的对齐合并具有相同标识符的直线，从而确定最终合并后的线段。该算法如算法 1 所示。我们在代码仓库中提供了相应的实现，并在实现中将阈值设为 $\tau_{dbscan} = 0.01$。

图 20 展示了 Hypersim数据集中的 ai 001_001和 ai 001 010 场景的结果定性比较 [17]。与未进行合并的原始结果相比，局部合并策略能够清理错误的线，并

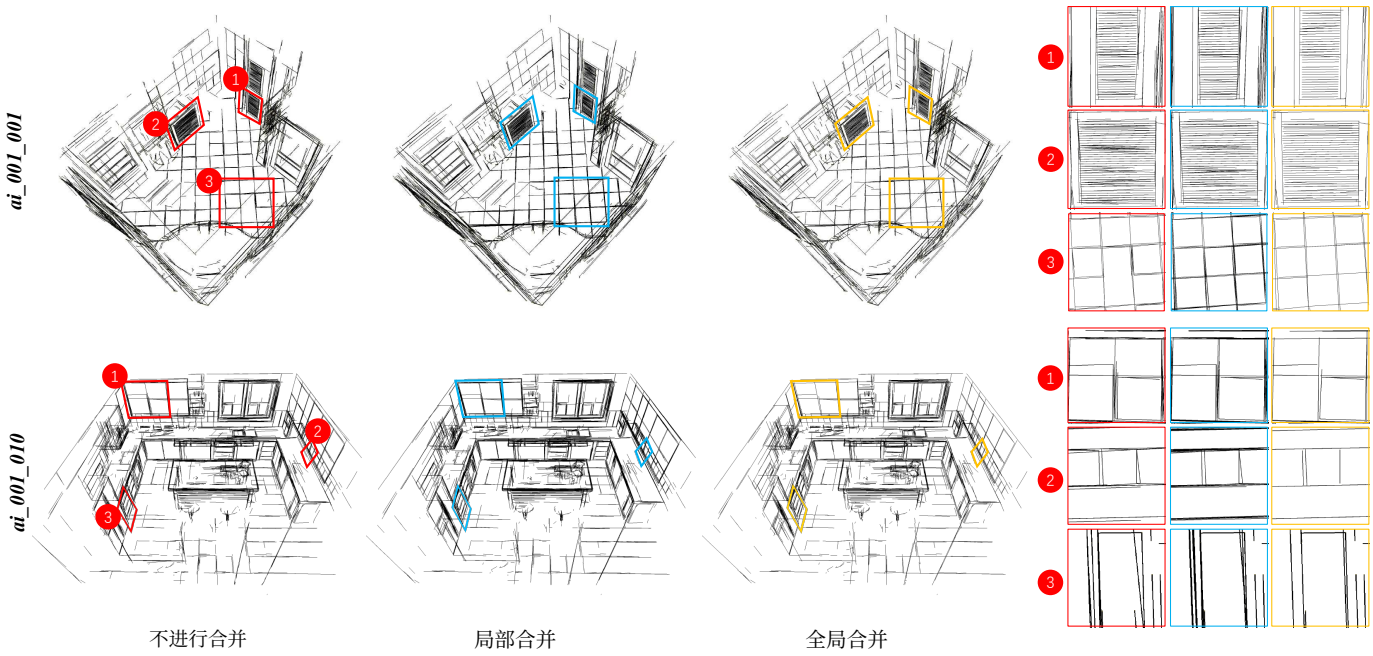


图 20：Hypersim数据集 [17]中两个场景的线合并策略定性结果。我们裁剪了一些图像块以便详细展示。**红色**：未进行合并的原始 LiP-Map。**蓝色**：采用局部合并的 LiP-Map。**橙色**：采用全局合并的LiP-Map。

合并对应的线，但也会引入更多冗余线。全局合并策略进一步整合对应的线，移除冗余线，最终提升线映射的质量和完整性。由于复杂度较高，全局合并策略的计算开销大于局部合并策略，但根据最终结果，这是可以接受的。

算法 1： 全局线合并策略

输入： 优化后的三维平面 $\mathcal{P}$、检测到的二维线 $\{\mathbf{L}_i\}_{i=1}^{N}$ 来自所有视图，垂直距离阈值 τ_{dbscan}
输出： 合并后的三维线地图 $\mathcal{L}_{merged}$
// 步骤 1：多视图线关联
对于每个视图 $i \in \{1, \ldots, N\}$ **执行**
  对于每个检测二维直线$\mathbf{l}_{i,\ j} \in \mathbf{L}_i$ **执行**
    通过从1像素区域投射射线，找出所有三维直线$\mathcal{L}_{i,\ j} \subset \mathcal{L}$ 并收集相交三维平面中的边缘，位于$\mathbf{l}_{i,\ j}$ 区域平面；
    // 步骤 2：局部 DBSCAN聚类
    计算中所有线对的两两垂直距离 $\mathcal{L}_{i,\ j}$ 每条线采样 10 个点；
    应用 DBSCAN，并使用阈值 τ_{dbscan} 获取簇 $\{\mathcal{C}_{i,j,k}\}$；
    // 步骤 3：全局标识符传播
    对每个 簇$\mathcal{C}_{i,\ j,\ k}$ **执行**
      如果任意线在$\mathcal{C}_{i,\ j,\ k}$ 具有全局标识符 g_{id} **则**
        将 g_{id} 分配给 $\mathcal{C}_{i,\ j,\ k}$中的所有线；
      else
        创建新的全局标识符 g_{id}^{new} 并将其分配给所有位于$\mathcal{C}_{i,\ j,\ k}$中的线；
// 步骤 4：基于主成分分析的直线合并
按全局标识符对所有三维直线进行分组 $\{\mathcal{G}_g\}$；
对每个全局组 $\mathcal{G}_g$ **执行**
  应用主成分分析，得到主方向$\mathbf{d}_g$ 和均值点$\mathbf{p}_g$；
  将 $\mathcal{G}_g$ 中的所有端点投影到主直线上，以找到 t_{min} 和 t_{max}；
  构造合并直线：$\mathbf{l}_g = (\mathbf{p}_g + t_{min} \cdot \mathbf{d}_g, \mathbf{p}_g + t_{max} \cdot \mathbf{d}_g)$；
  将$\mathbf{l}_g$ 添加到 $\mathcal{L}_{merged}$；
返回 $\mathcal{L}_{merged}$

表16：ScanNetV2数据集上的平面重建结果定量比较 [15]。

	VOI↓	RI↑	SC↑
PlanarSplatting [8]	2.489	0.951	0.530
Ours	**2.469**	**0.952**	**0.532**

附录 E 三维平面重建的增强

由于我们的 *LiP-Map* 增强了平面边缘的重建，从而能够更准确地感知物理边界，因此还能进一步提升三维平面重建的性能。遵循 PlanarSplatting [8]，，我们采用信息变差（VOI）、兰德指数（RI）和分割覆盖（SC）等指标。如表16所示，与 PlanarSplatting [8]相比，我们的 *LiP-Map*表现更优。我们还在图21中展示了重建结果的比较。从比较结果可以看出，我们的方法能有效地将平面基元的边缘与场景中的线结构对齐，从而生成几何拓扑更加鲜明的重建网格。这使得重建结果更加准确、合理，并能够从场景中提取完整且精确的三维线地图。

附录 F 局限性

尽管我们的方法具有诸多优势，但仍存在一些有待进一步研究的局限。由于二维直线检测存在误差，可能会出现重复的三维线段，进而

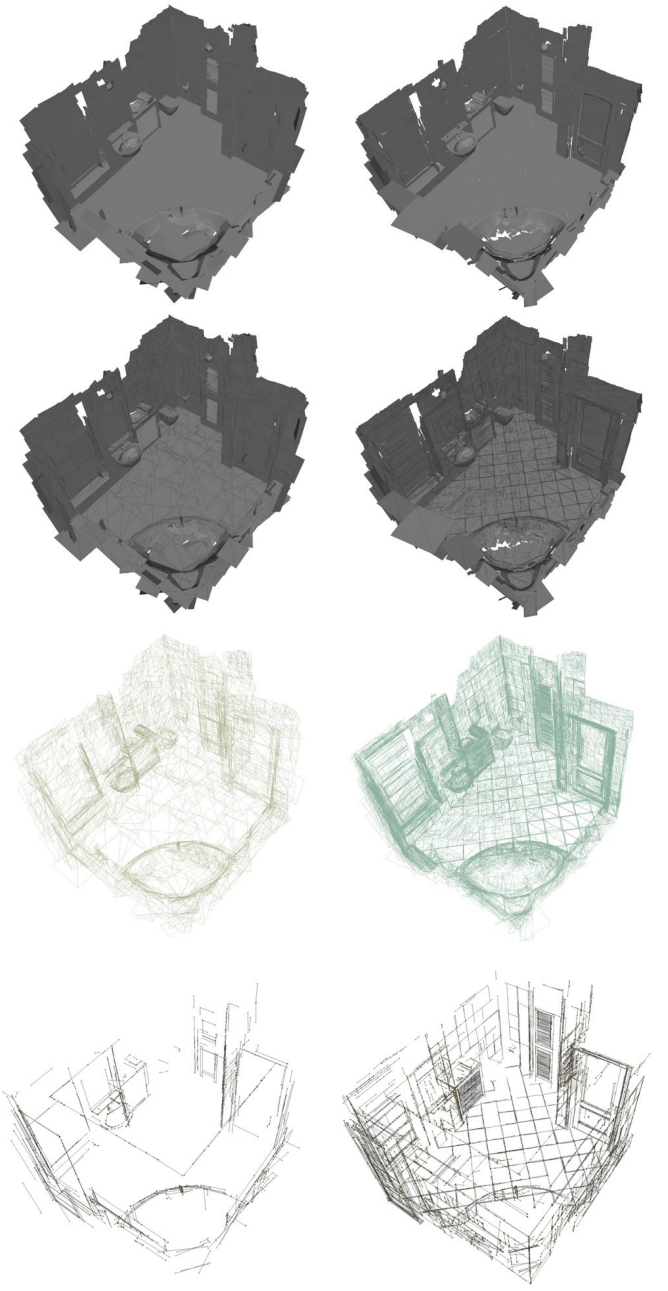


图21：在 Hypersim 的“ai 001 001”上，将平面网格/边缘重建与基线 PlanarSplatting 进行比较的示意图 [8] 。左列：PlanarSplatting。右列：我们的方法。第一行：平面表面。第二行：平面网格。第三行：平面边。最后一行：三维线图。

违反三维线映射中的简约原则。此外，三维线段的基于平面的表示特别适用于重建人工环境中的建筑结构，但在重建非平面非结构化物体方面的能力有限。尽管如此，我们认为，三维线段重建的主要目标是感知环境中的几何特征，并高效捕捉建筑物的轮廓和拓扑结构，从而生成轻量且语义丰富的矢量化模型——而这些正是我们的方法

表17：优化时间和平面数量随分裂阈值的变化。

阈值	0.02	0.1	0.2	1.0	2.0	10
时间（秒）	948.5209	388.9839	333.0532	312.5148	305.0371	301.3305
初始平面数	1873	1873	1873	1873	1873	1873
最终平面数	115615	19876	10064	2071	1369	1072

该方法展现出明显优势。

附录G 我们线图的更多定性结果

我们展示了在ScanNetV2数据集 [15]（图22）、theScan-Net++ 数据集 [16] （图23）以及Hypersim数据集 [17]（图24）中的场景上，我们的三维线图的更多定性结果。

附录H 关于对启发式方法和超参数敏感性的定性结果

我们已在第4.4.3节报告了关于对启发式方法和超参数敏感性的定量结果。本节将进一步展开讨论，并提供相应的定性结果。

平面分裂阈值（基于半径梯度）。在所有实验中，该参数均固定为 0.2。一方面，我们将其设定与 Planar- Splatting [8] 保持一致。另一方面，阈值过小会导致平面分裂过于激进，使平面数量激增；而阈值过大则会减慢分裂速度，导致平面数量不足，无法支持精细重建。我们通过报告表17中的优化时间和平面数量统计数据，并在图25中展示一些重建细节，验证了这一观点。

随机平面基元的初始数量。我们已在表15中详细报告了优化时间随初始平面数量变化的结果，并在图15中提供了相应的定性比较。值得注意的是，在处理大规模场景时，初始化时使用更多平面确实有助于获得更好的重建结果（如图36(b)所示）。因此，最直接的选择是对所有场景都采用较大的初始化值（例如 50,000）；唯一的权衡是优化时间略有增加（如表15所示）。对于一般的室内场景，2000 个初始平面已足够。

三维线映射的分配阈值。如第 3.4 节所述，分配阈值仅用于推理阶段，在优化后完成最终的线映射。本文采用一对非常严格的阈值，$\tau_d = 1$ 像素和 $\tau_\alpha = 0.01$ 弧度（约 0.57 度），以确保三维线映射的准确性。放宽阈值会产生更多三维直线，并输出更完整的三维线图，如图 26 所示，但也会引入大量噪声线。

联合优化中的平衡权重 $\alpha_{\mathbf{\Pi}}$、α_L 。我们选择的平衡权重，其中 $\alpha_{\mathbf{\Pi}} = 10$、$\alpha_L = 0.1$，为

旨在使平面表面优化损失与三维线优化损失达到相近的数值尺度。这种平衡也使优化过程非常稳定。我们还很高兴在图27中提供了一些结果的定性可视化。可以看出，用于三维线优化的较大权重 α_L 会导致平面数量快速增加，从而延长优化时间。相比之下，用于平面表面优化的较小权重 $\alpha_{\mathbf{\Pi}}$会减少平面数量，这会影响完整且细粒度的场景重建。

附录I 初始化质量的影响：

我们同意，了解性能提升主要源于所提出的联合优化框架，还是仅仅继承自高质量先验，至关重要。

ScanNetV2。 为此，我们在 ScanNetV2 [15] 上使用不同质量的深度图进行了额外实验，例如原始传感器深度图、Metric3D [53]，预测的单目深度图，以及近期 SOTA 方法 MoGe-2 [68] 生成的深度图。图28展示了一些定性结果，以便进行详细比较。结果表明，尽管初始深度图的质量确实会影响最终结果，我们的联合优化框架仍能显著改善几何结构，与初始网格相比，生成更完整、更平滑的网格。具体而言，即使以噪声较大或稀疏的深度估计作为起点，经过细化过程，线图精度和表面一致性也得到了显著提升。

此外，我们在 Scan- NetV2 [15] 上进行了实验，以研究法线图对我们方法的影响。我们使用传感器深度图，并分别使用来自 Metric3D [53]，、MoGe-2 [68],和 Omnidata [52]，的法线图进行优化。结果如图29所示。对于每个场景，我们展示一个视图的法线图以及重建网格中对应的部分，作为简洁的对比，以详细展示法线图对优化结果的影响。可以看出，法线图可能会对最终优化后的网格产生一定影响，但所有设置最终仍能生成质量相近且较高的三维线图。

Hypersim。 同样，针对深度图对优化结果的影响，我们在Hypersim数据集 [17] 上进行了大量实验，分别使用真实深度图、Metric3D生成的深度图以及近期SOTA方法MoGe-2生成的深度图，同时保持其他设置完全一致。图30对比了Metric3D [53] 和MoGe-2 [68] 生成的初始网格与我们优化后的平面网格，表明即使从低质量的初始化开始，我们的方法也能恢复出高质量的平面网格。图31显示，即使初始网格质量较差，我们的优化框架仍能生成高质量的线图，其效果可与使用真实网格初始化时相媲美。图32和图33展示了Hypersim中其他场景的更多结果。

以上大量结果表明，我们的优化框架能够应对输入深度图和法向量图中一定程度的噪声和稀疏性，也证明了所提出的优化对最终输出有实质性贡献，而不仅仅是拟合输入先验。

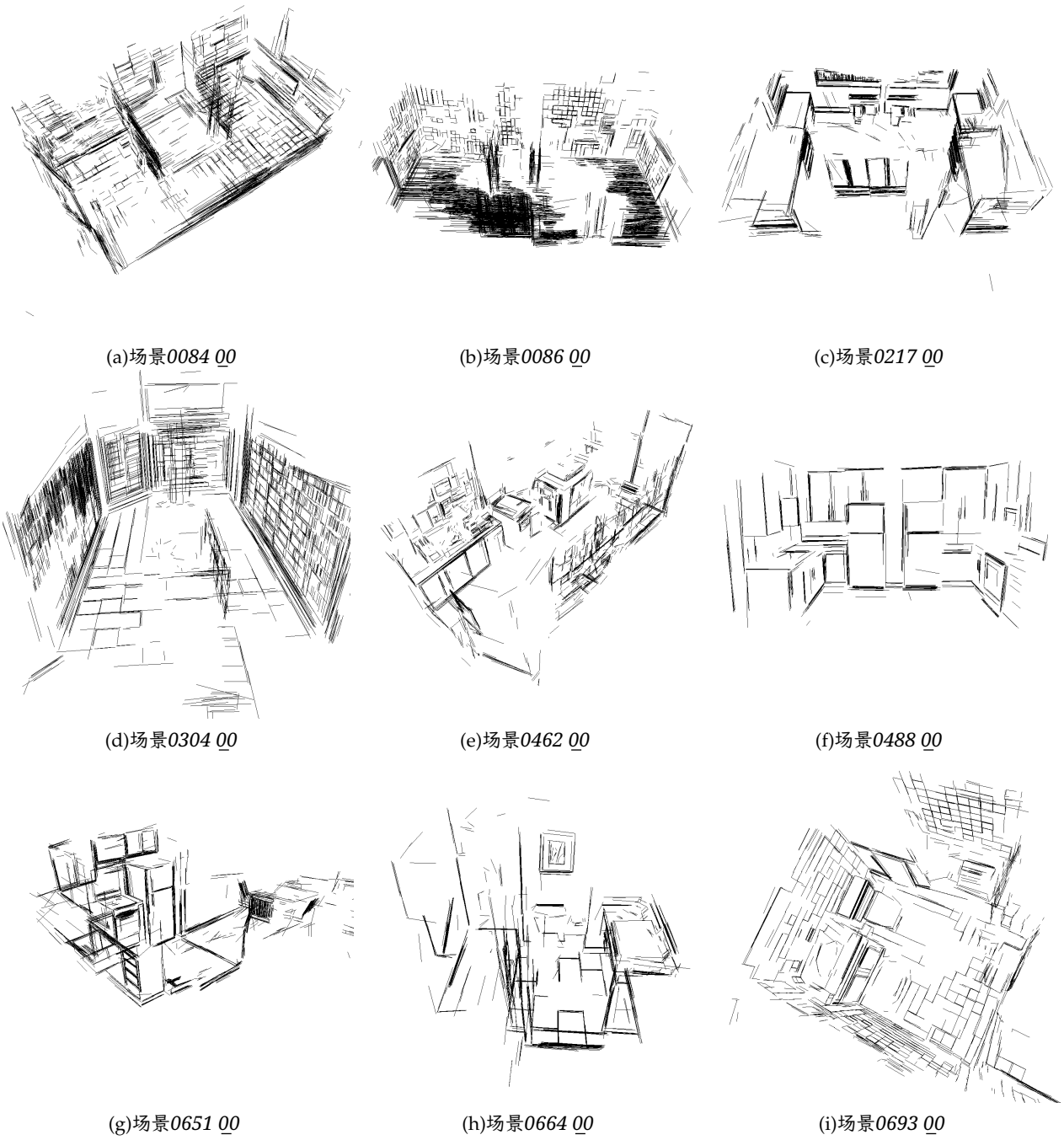
附录 J 关于泛化到非平面结构的更多结果

我们研究了我们的方法在通用非平面结构上的泛化能力，并仅使用来自一些常见数据集场景的多视图图像进行了实验。在本节中，我们从包含此类非平面几何结构的 Hypersim [17] 场景中选取若干视图，并在图34中可视化二维线检测结果、网格以及由我们的方法重建的线图。可以看出，我们的方法能够通过使用密集的平面基元集合拟合这些非平面结构，近似表示曲面，这表明该方法能够满足重建许多常见人造室内场景中非平面结构的需求。

此外，我们在图 35 中展示了一些非平面场景的局部失败案例。红框标出了不准确或幻觉平面，这些平面往往是为了满足来自噪声深度图或法线图的监督而产生的。然而，我们的分配策略采用了严格的阈值（如第 3.4 节所述），确保 2D 与 3D 线之间具有较强的几何一致性，从而能够过滤掉这些不准确的平面边缘。

附录 K 失败案例

我们总结了四种不同条件下的一些失败案例，并在图 36 中展示了定性结果，其中包括完全失败和部分不理想的案例。为了尽可能全面地探索我们方法的适用边界，所有结果均使用 VGGT 的 [20]不完美输出进行重建。(a) 纯旋转：这是一个典型的病态问题。我们在 Blender 中对两个场景分别渲染了 100 个视图，过程中仅旋转相机而不进行平移。纯旋转序列会导致 VGGT 的输出不可靠，使初始网格出现严重的几何漂移，最终导致重建失败。(b) 初始平面数量：给定相同的初始网格，如果用于初始化平面表面的平面基元数量不足，大规模场景的重建也可能失败。(c) 非平面场景：对非平面结构化场景进行重建对我们的方法而言仍然具有挑战性，其结果不如室内场景的重建准确、简洁。(d) 初始网格/深度：在初始化时使用相同数量的平面基元，如果初始网格或深度图存在噪声或不准确，也会降低重建质量。

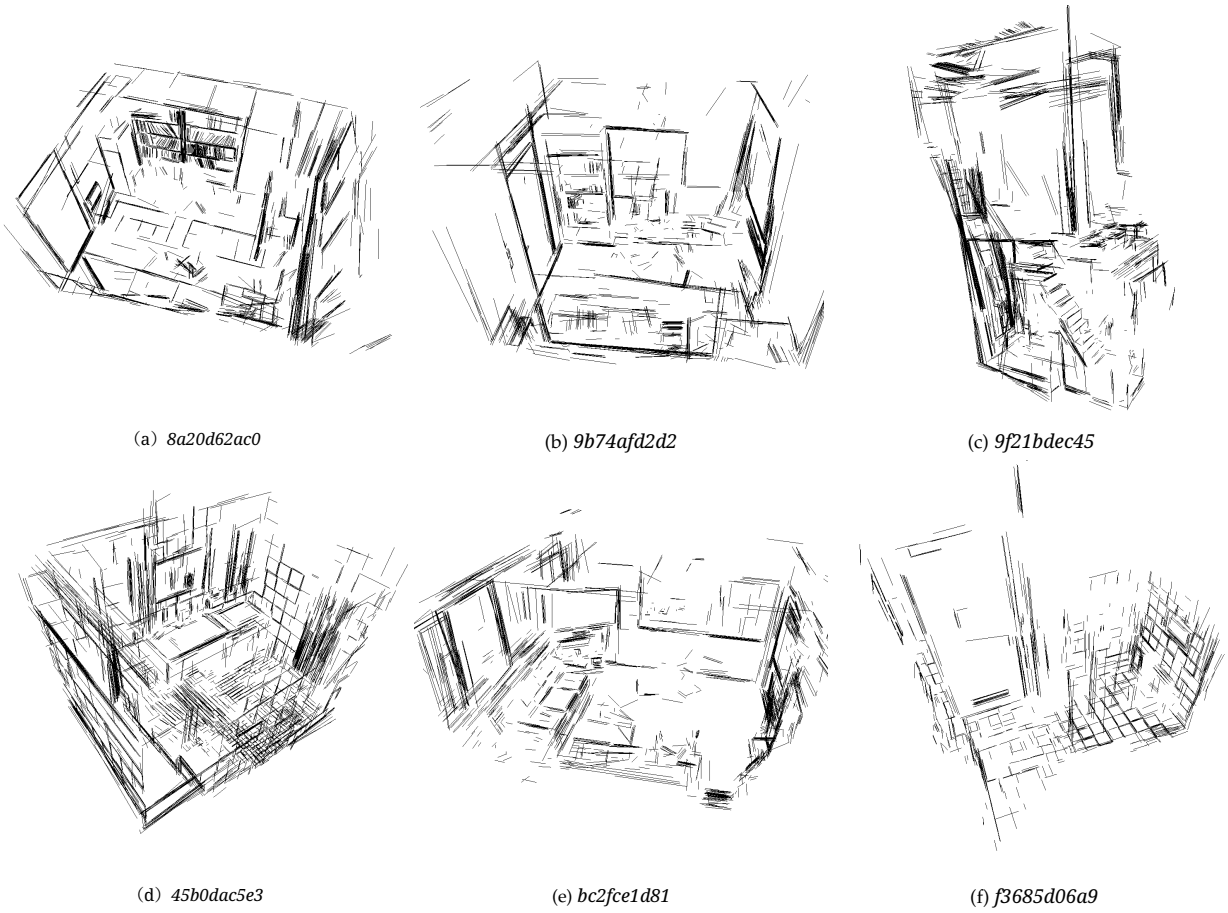


(a)场景*0084 00* (b)场景*0086 00* (c)场景*0217 00*

(d)场景*0304 00* (e)场景*0462 00* (f)场景*0488 00*

(g)场景*0651 00* (h)场景*0664 00* (i)场景*0693 00*

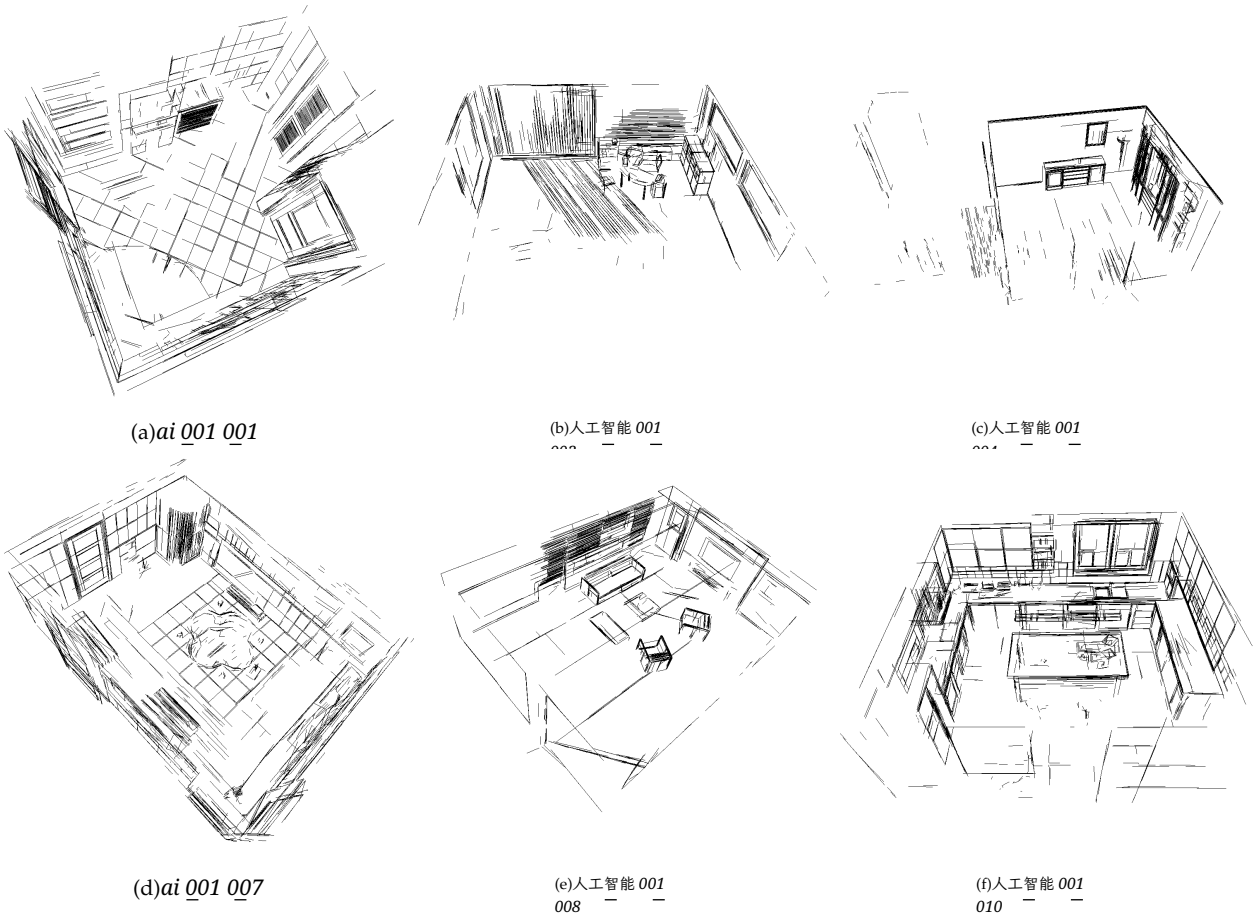
图 22：我们的方法在 ScanNetV2数据集 [15] 上恢复的三维线图的更多定性结果。



（a）*8a20d62ac0* (b) *9b74afd2d2* (c) *9f21bdec45*

（d）*45b0dac5e3* (e) *bc2fce1d81* (f) *f3685d06a9*

图 23：更多定性结果 我们的方法在 ScanNet++ 数据集 [16]上恢复的三维线图的更多定性结果。



(a)*ai 001 001* (b)人工智能 *001* (c)人工智能 *001*

(d)*ai 001 007* (e)人工智能 *001 008* (f)人工智能 *001 010*

图 24：更多定 性结果：我们的方法在 Hypersim数据集上恢复的三维线图 [1 7].

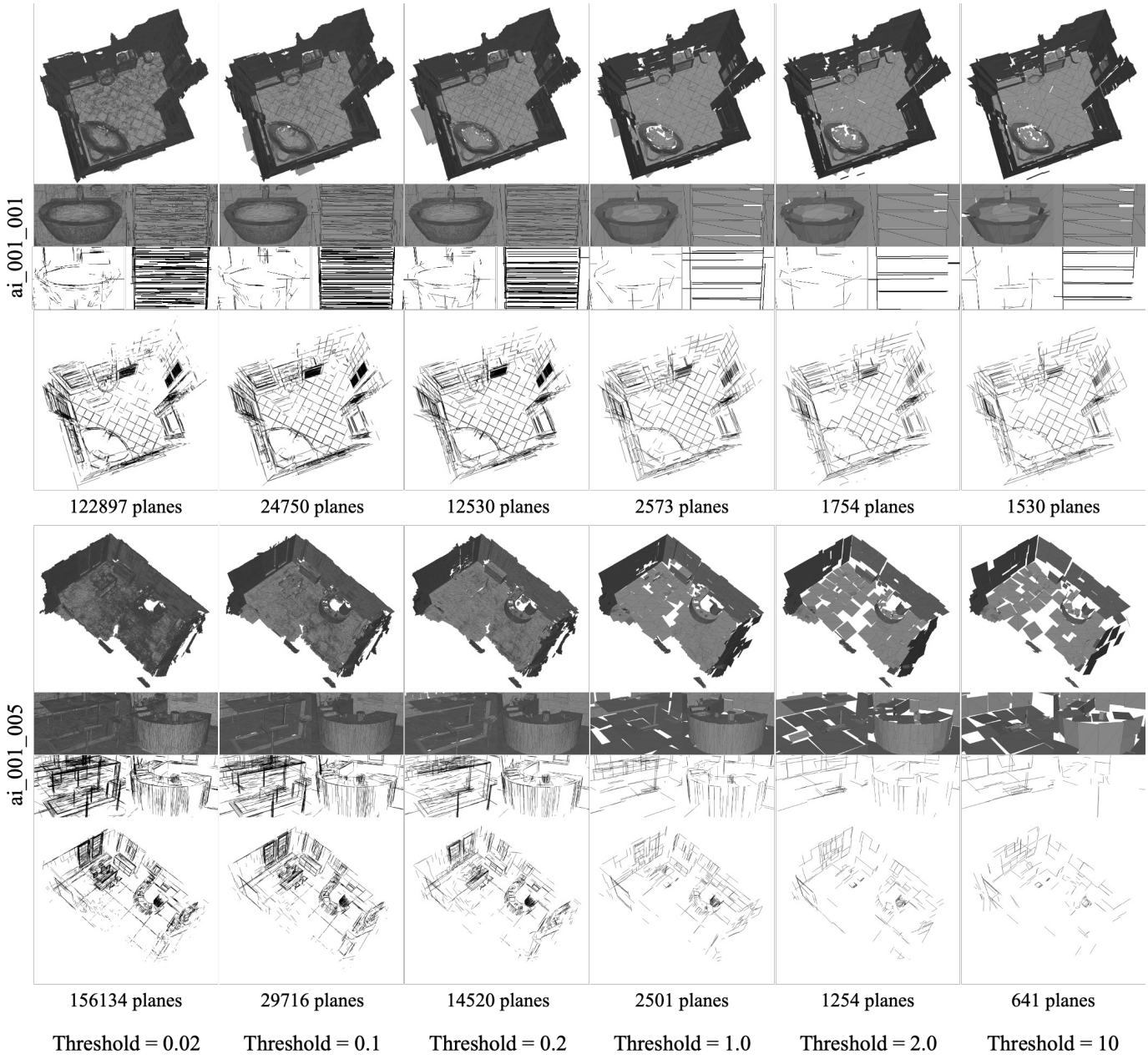


图25：不同平面分割阈值下的重建定性结果。部分图像块被裁剪以便进行详细比较，最终的平面数量附于下方。

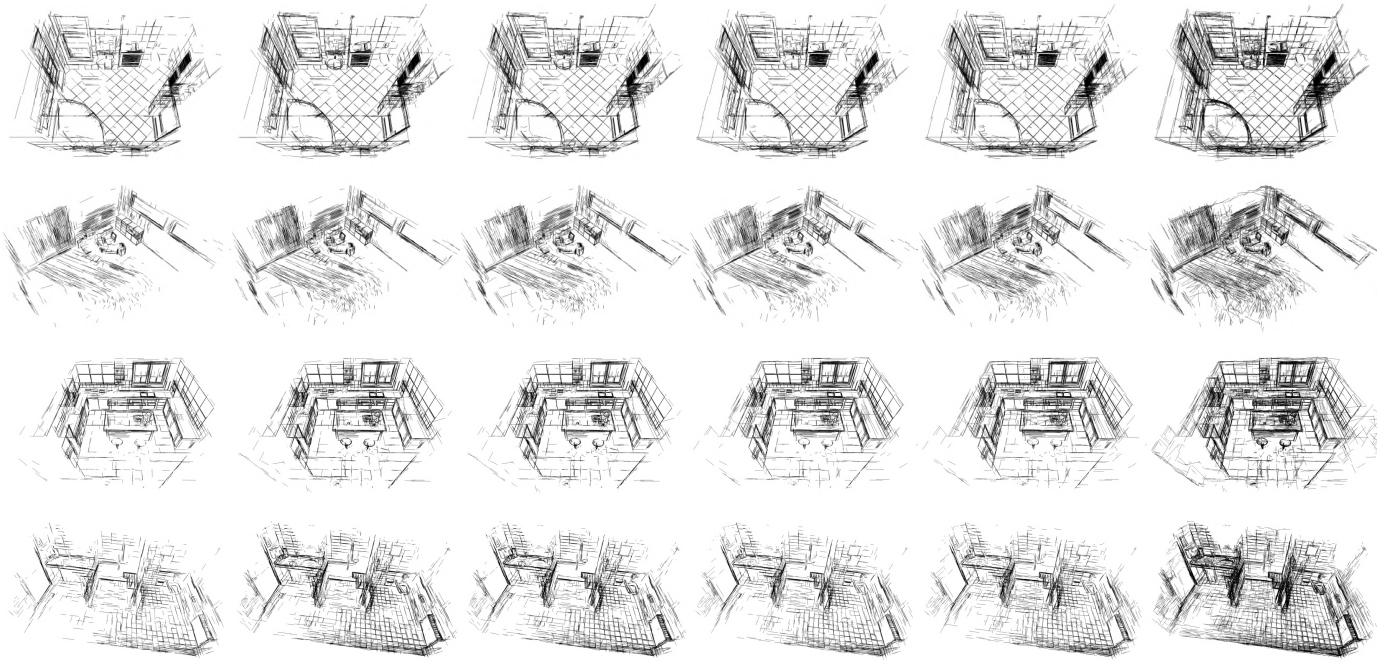


图 26：采用不同分配阈值组合进行三维线映射的重建定性结果。

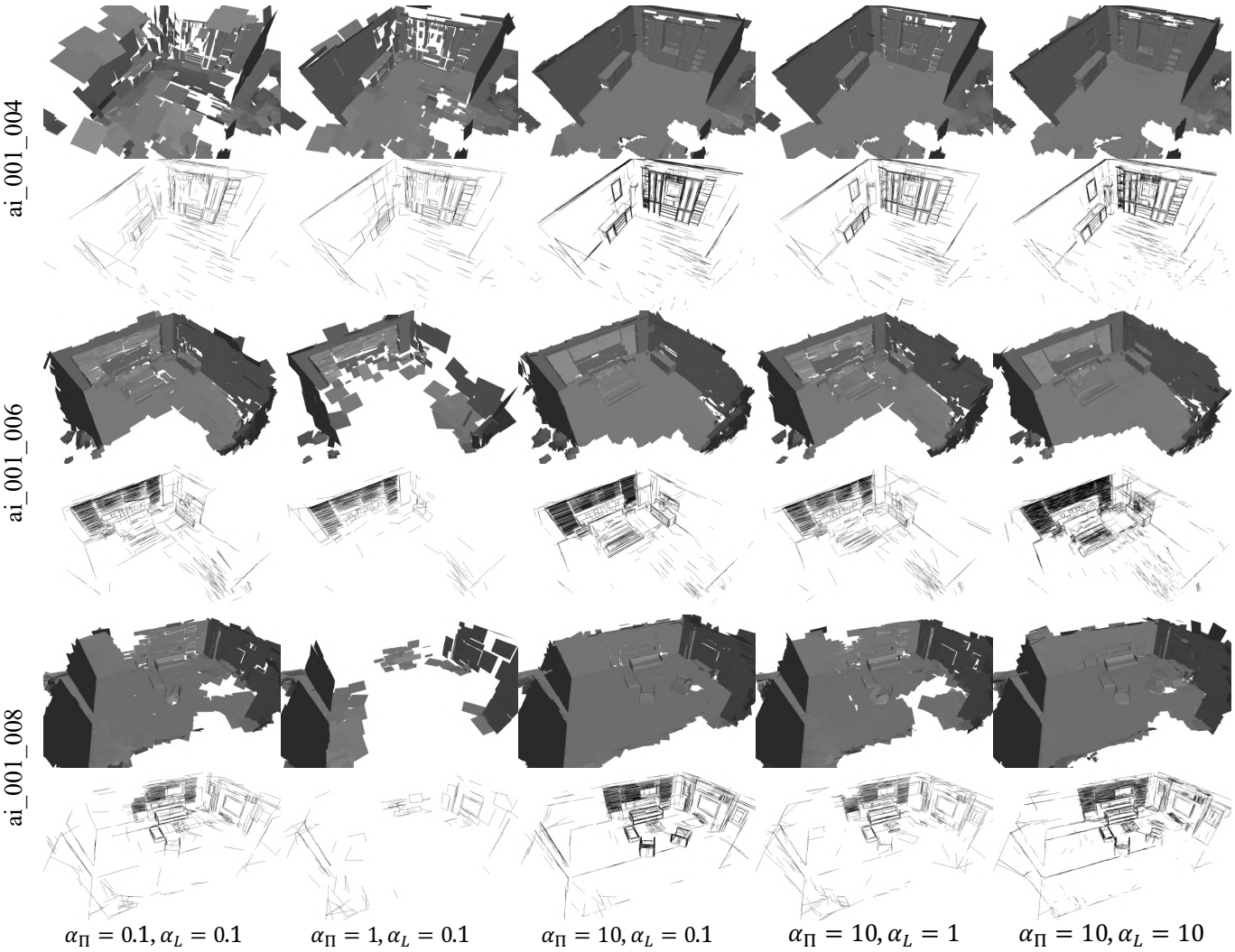


图27：Q 采用不同损失权重组合进行三维线地图重建的定性结果

g.

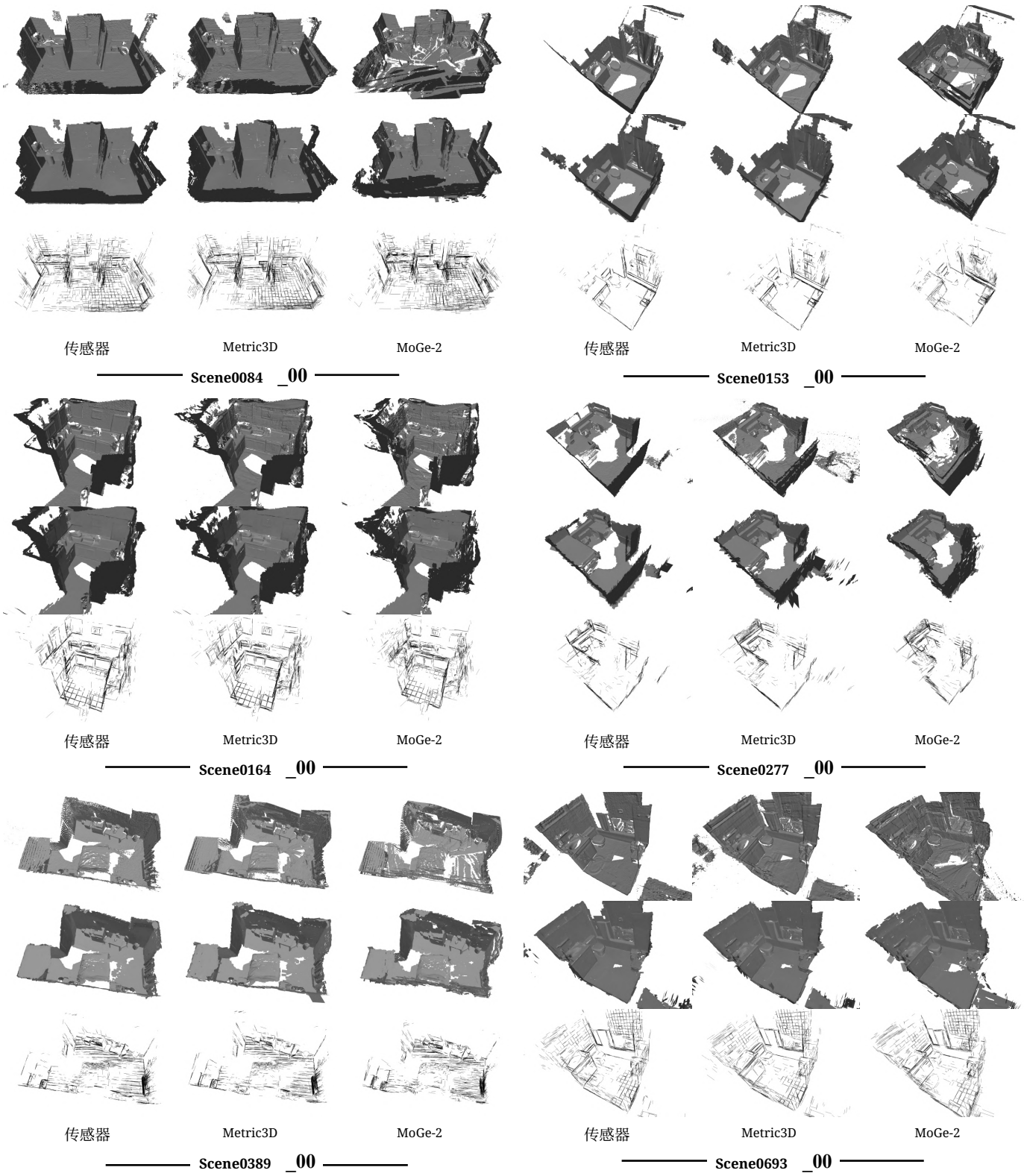


图28：在 ScanNetV2 [15] 上对初始网格（第一行）、我们的平面网格（第二行）和我们的三维线图（最后一行）进行定性比较。本实验采用 Omnidata [52] 预测的法向量图。

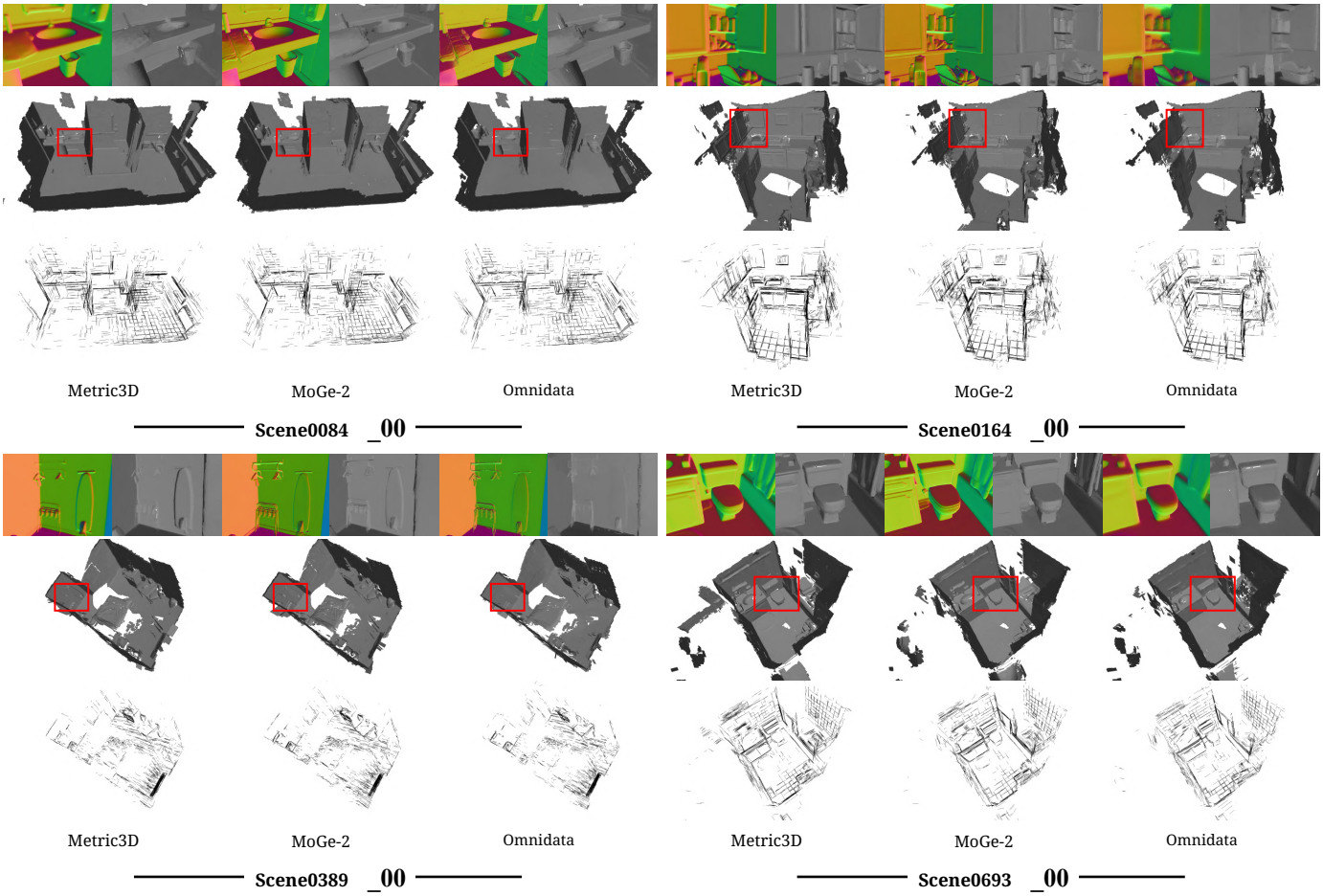
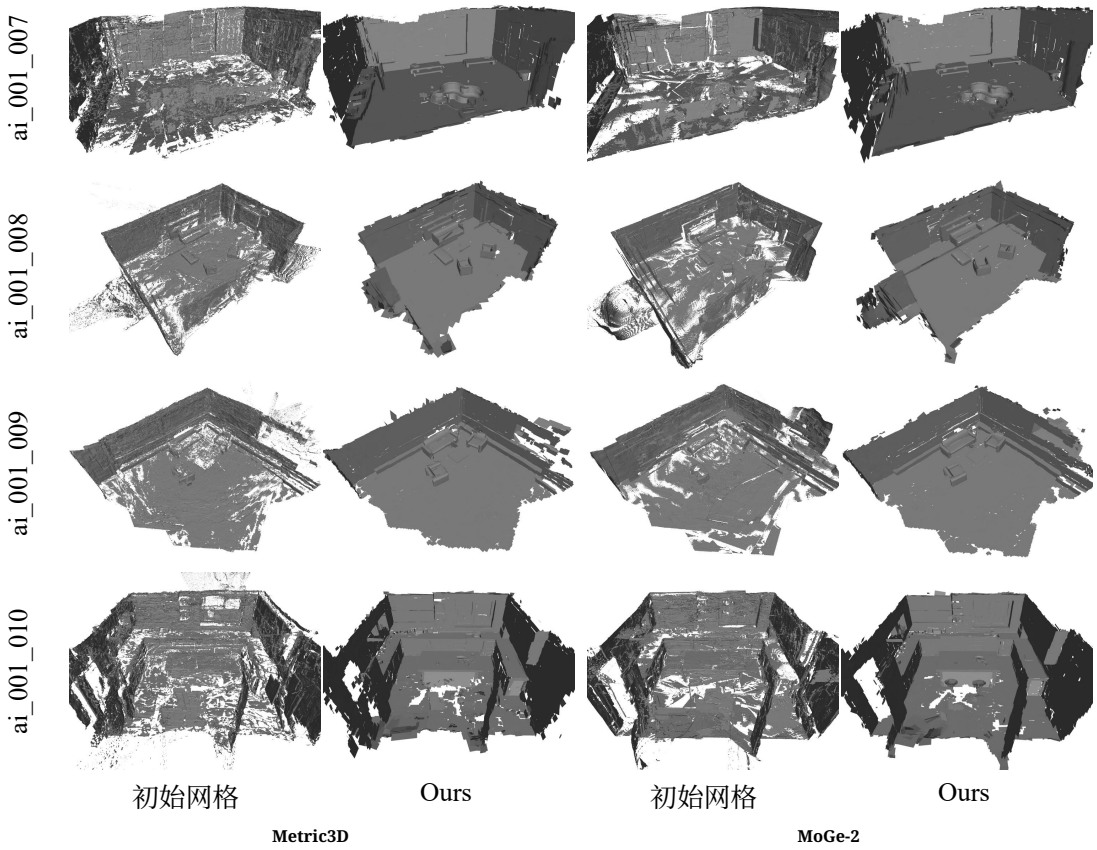


图29：使用不同模型在 ScanNetV2 [15] 上获得的法线图重建的平面网格和三维线图的定性比较。本实验采用原始传感器深度图。



Fi图30：Hypersim 上初始网格与我们的平面网格的定性比较 [17]。

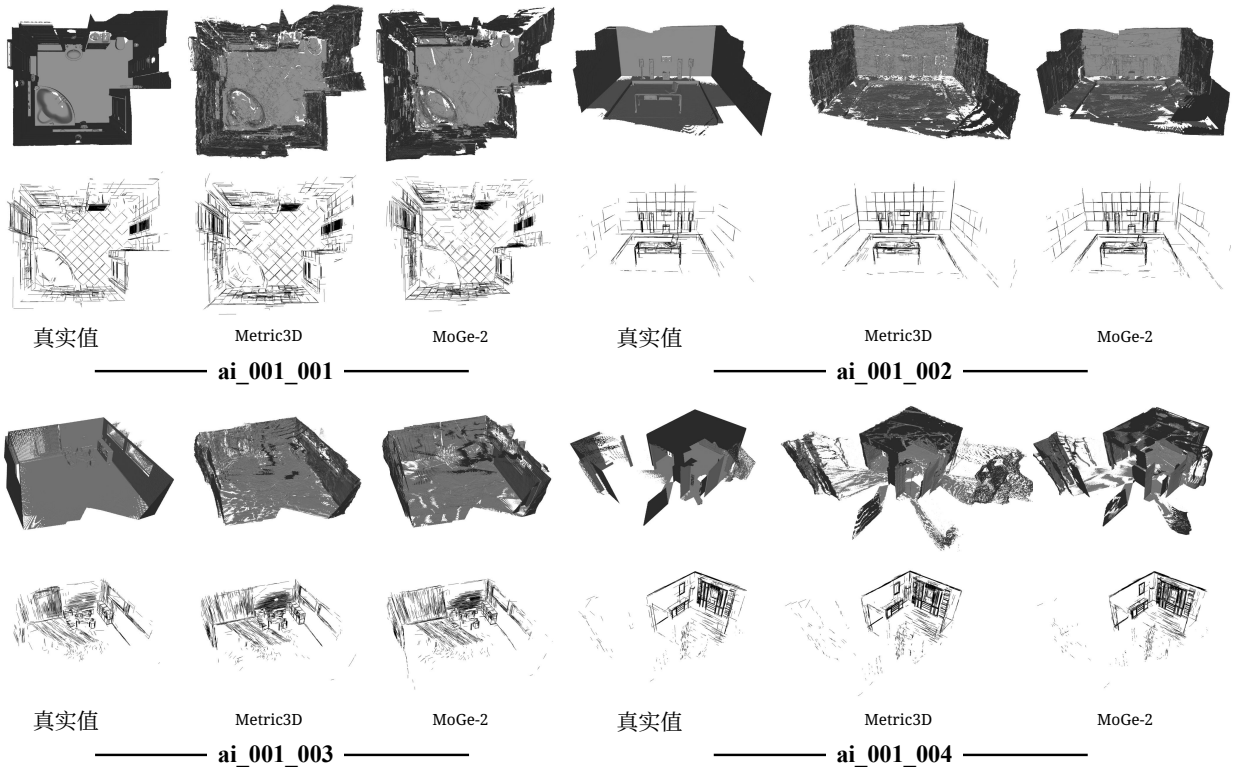


图31：Hypersim 上不同质量初始网格的三维线映射定性比较 [17]。

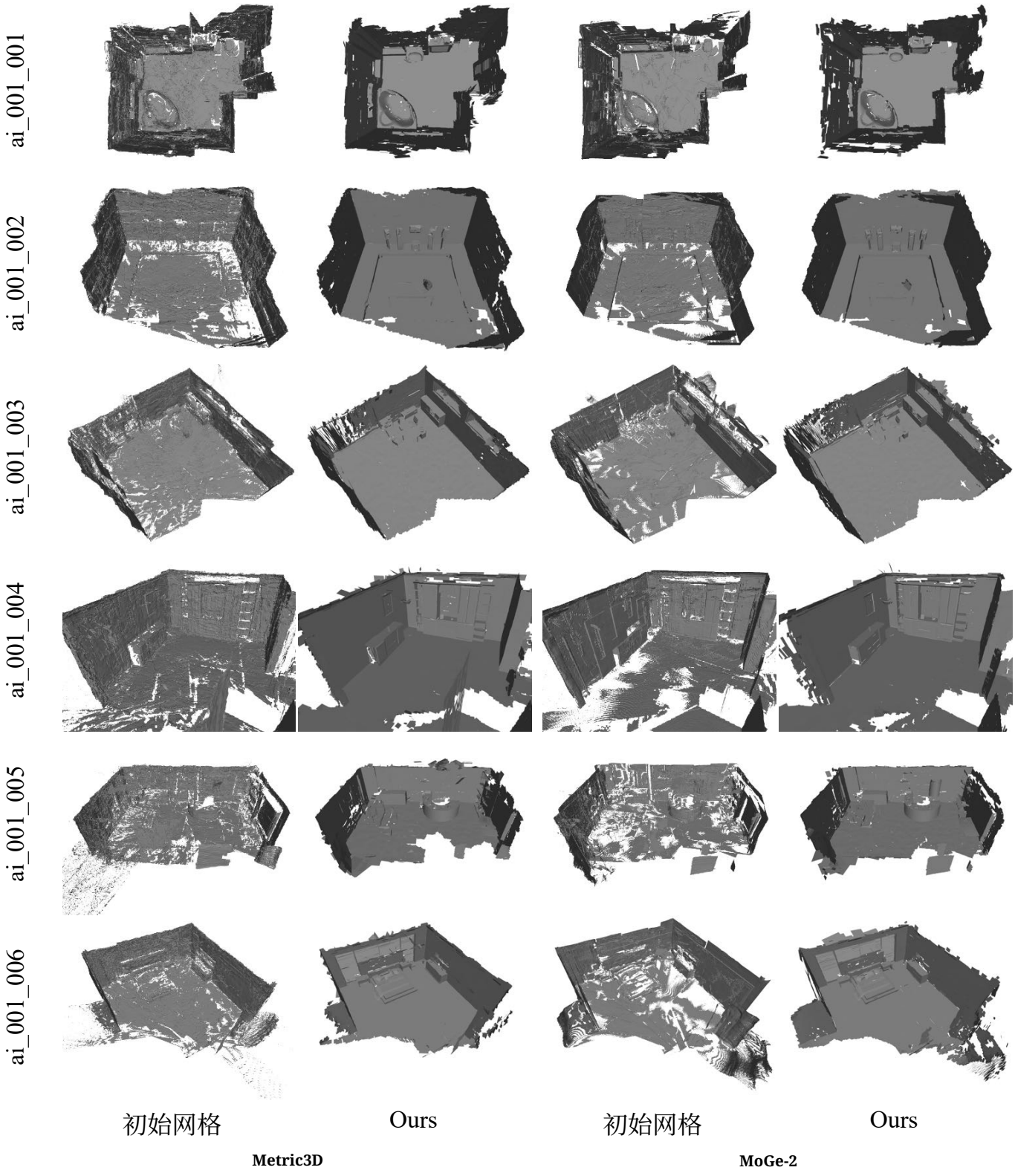


图32：在 Hypersim [17] 上对初始网格和我们的平面网格进行的更多定性比较。

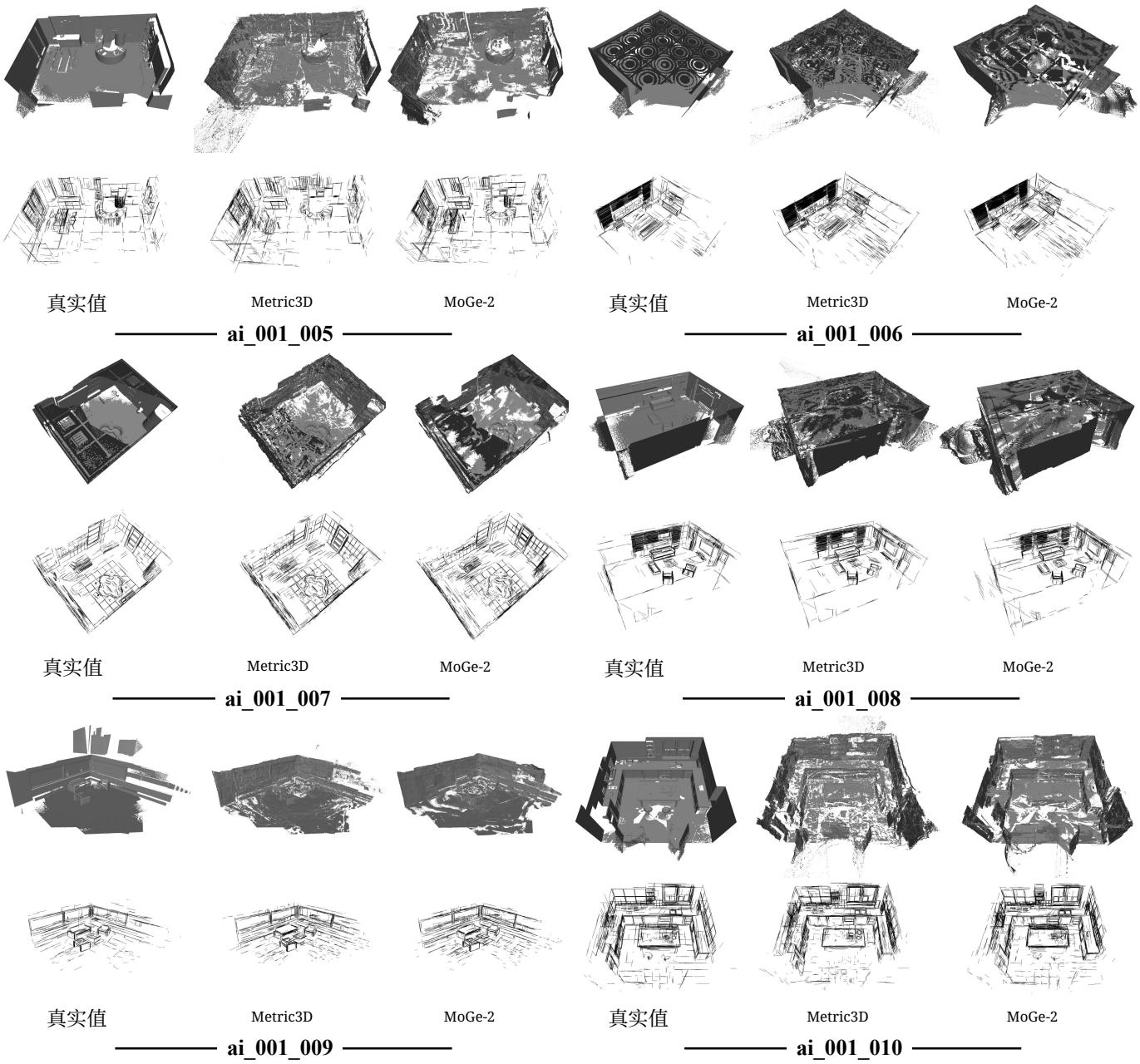


图33：在 Hyper- sim [17] 上，使用不同质量的初始网格/深度进行三维线映射的额外定性比较。

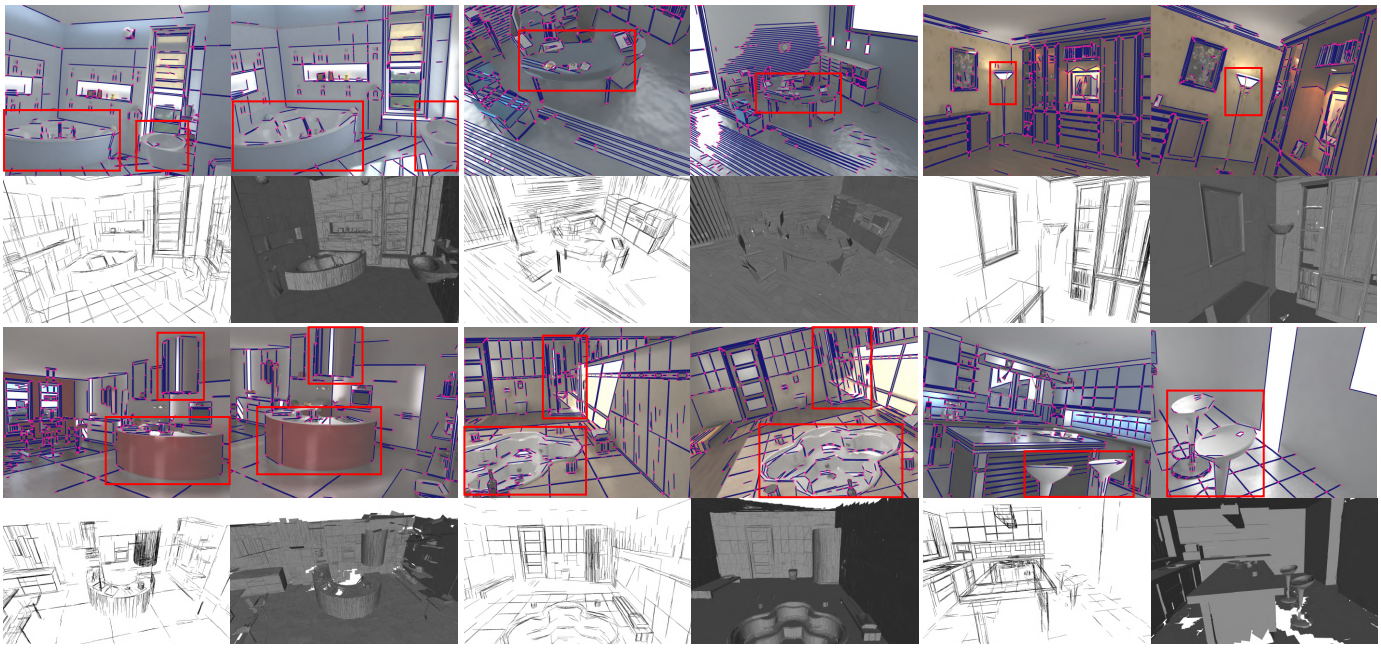


图 34：Hypersim [17] 场景中非平面结构重建的定性结果。我们展示了椅子和圆桌等曲面结构的二维线检测结果、重建网格和三维线图。

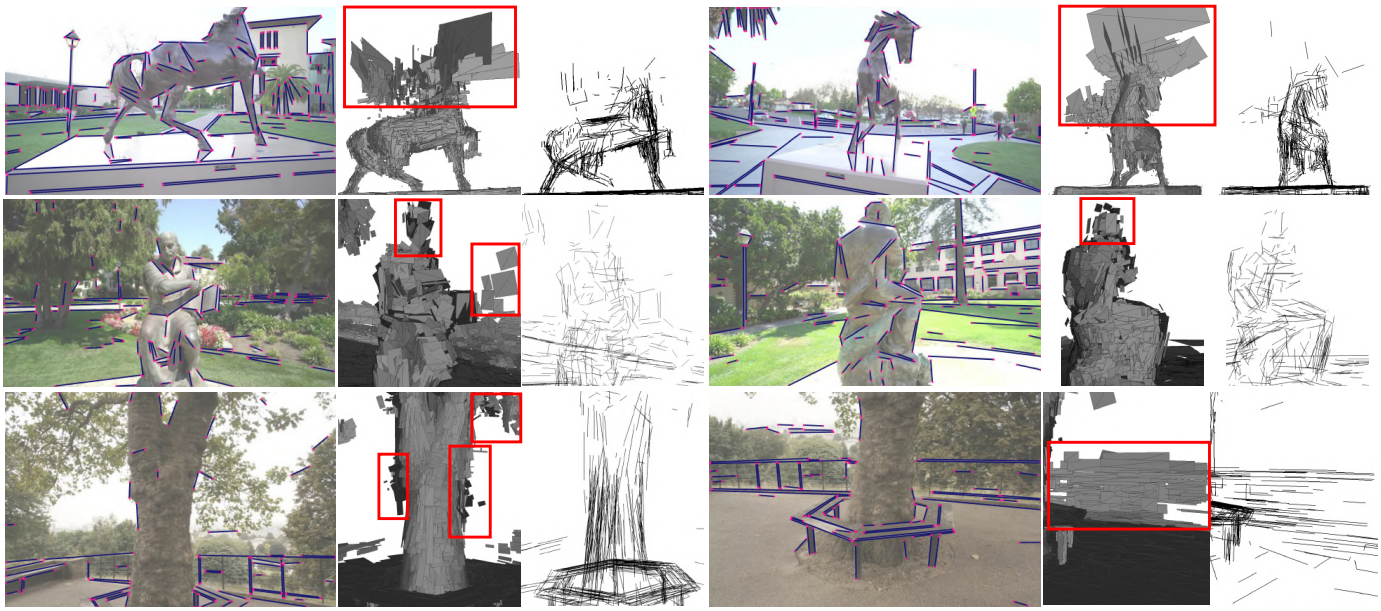


图 35：局部非平面结构的定性结果。图中展示了二维线检测结果，以及重建的三维平面表面和三维线段，供参考。

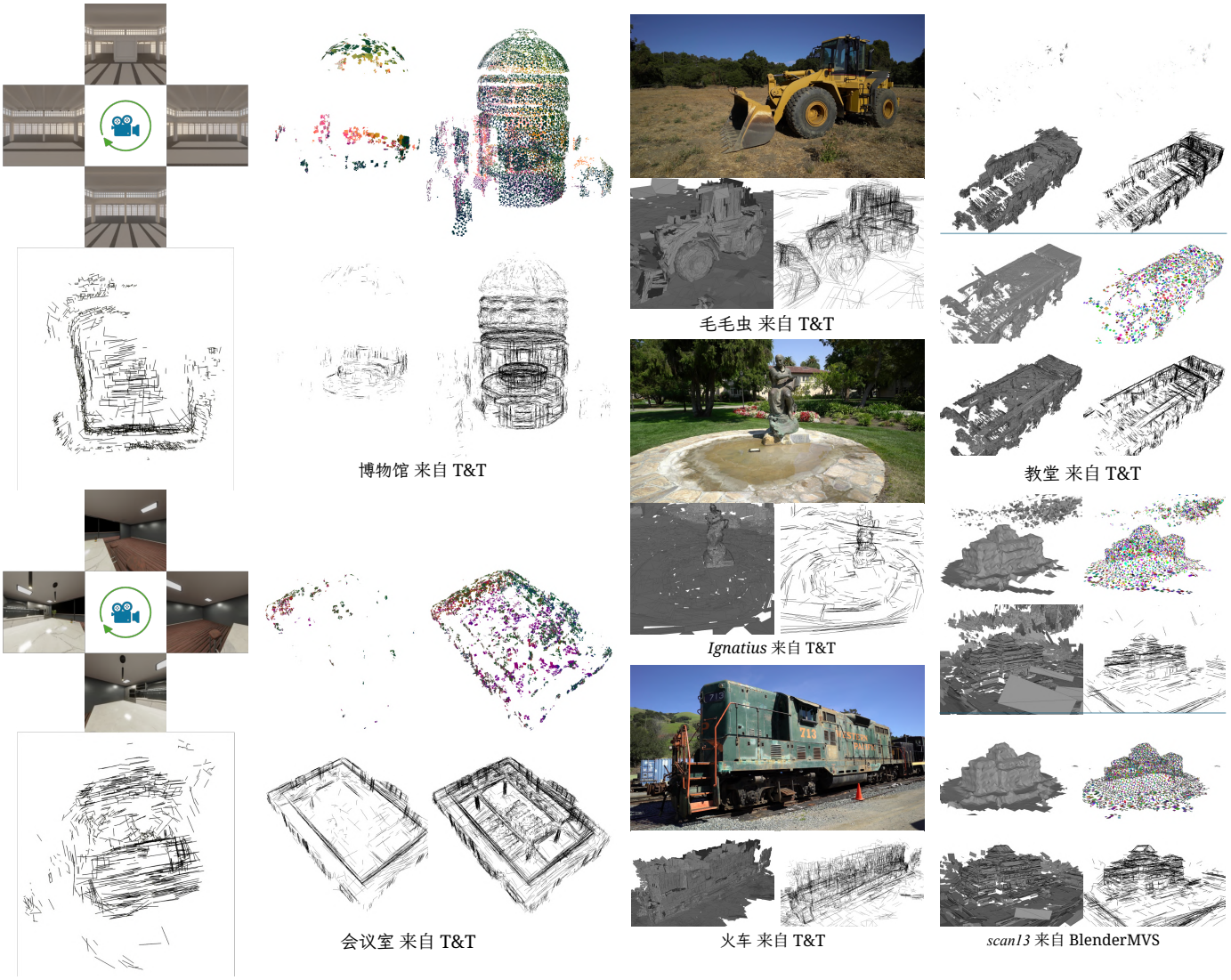


图 36：我们完全／部分失败案例的可视化。