

基于最小化点到平面距离的逆平面拟合的法线积分

摘要

本篇论文提出了一种表面法线积分方法，解决了局部平面拟合的逆问题。从法线图重建表面在光度形状重建中至关重要。为此，我们在相机坐标系中公式化了法线积分，并联合求解三维点位置和局部平面位移。与现有方法考虑三维点之间的垂直距离不同，我们最小化了点到平面距离的平方和。我们的方法可以处理具有任意边界的正射投影或透视法线图。与现有的法线积分方法相比，我们的方法避免了棋盘格伪影，并且在处理自然边界、锐利特征和离群点时表现得更加鲁棒。我们进一步基于我们的平面拟合公式，对先前方法中出现的伪影源进行了几何分析。通过在解析计算、合成和现实世界表面上的实验结果，我们的方法能够对正射投影和透视法线图进行准确且稳定的重建。

关键词：法线积分；三维重建；逆平面拟合

1 引言

法线积分 (Normal Integration) 指从一个表面的法线图重建表面 [8]，是三维重建中一个非常重要的课题，其在非常多的领域有所应用如人脸识别、自动驾驶、游戏开发等等。法线积分存在非常多的方法，例如阴影重建 [5]、光度立体重建 [10]、偏振重建等 [6]。对于所有的这些方法，一般主要可以分为两类：基于变分的方法 [3,4,8] 以及非变分方法 [11]，基于变分的方法已经被研究了许多年，如今已经非常成熟，其核心思想是基于对一个特定函数的优化，通过离散化函数本身或其最优条件以离散泊松方程的形式离散化，将目标转化为线性系统。非变分方法目前主要是基于 DGP (discrete geometry processing) [11] 提出，其同样通过解线性系统来估计表面。然而，对于这三种线性系统，它们的最小二乘近似解都面临着一些问题：

- (1) 对于离散函数，其存在棋盘伪影问题，如图 1 所示。
- (2) 离散泊松方程和 DGP 方法对锐利特征或离群点比较敏感，如图 1 所示。

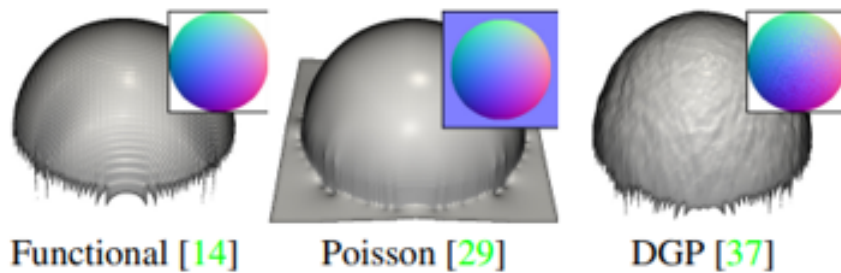


图 1. 各方法的效果

为了解决这类问题，人们提出了很多办法，尤其是在处理伪影时提出了很多不同策略：

(1) 正则化 [2,3] 或平滑项 [12]：这种方法通过在目标函数中加入惩罚项，限制解的变化，从而提高重建的平滑性和稳定性。然而，正则化需要选择一个合适的权重因子，且这个过程可能需要多次调优才能得到令人满意的结果，这增加了计算的复杂性。

(2) p-范数的残差向量 [1]：传统的 2-范数（欧几里得范数）可以替换为 p-范数（当 $p > 2$ 时），从而增加对异常值的鲁棒性。这种方法能够处理一些数据中的噪声和异常值，但计算量较大，尤其是当 p 值较大时，计算开销会显著增加。

(3) 小俯角的法线向量作为异常值处理 [11]：对于具有小俯角（小角度）的法线向量，可以将其视为异常值进行处理。这种方法使用启发式的阈值进行判断，但该阈值的选择并不能保证对所有异常值的鲁棒性，可能并不适用于所有情况。

本文介绍了一种新的法线积分方法，通过优化点的位置和局部平面位移来解决法线积分问题。本文的核心思想如下：

(1) 残差向量与线性系统：与传统方法不同，本文聚焦于线性系统的残差向量。通过有限差分离散化函数或泊松方程，作者表示该残差向量代表了 3D 点之间的垂直距离，即沿视图方向的距离。该方法通过局部平面拟合来优化点的位置和法线，从而改进法线积分。

(2) 点到平面的距离：本文引入了垂直的点到平面距离（沿平面法线方向的距离），这与传统的垂直距离有所不同，如图 2 所示。具体而言，法线积分被公式化为局部平面拟合的逆问题，并通过引入平面位移作为未知量，优化点的高度值。

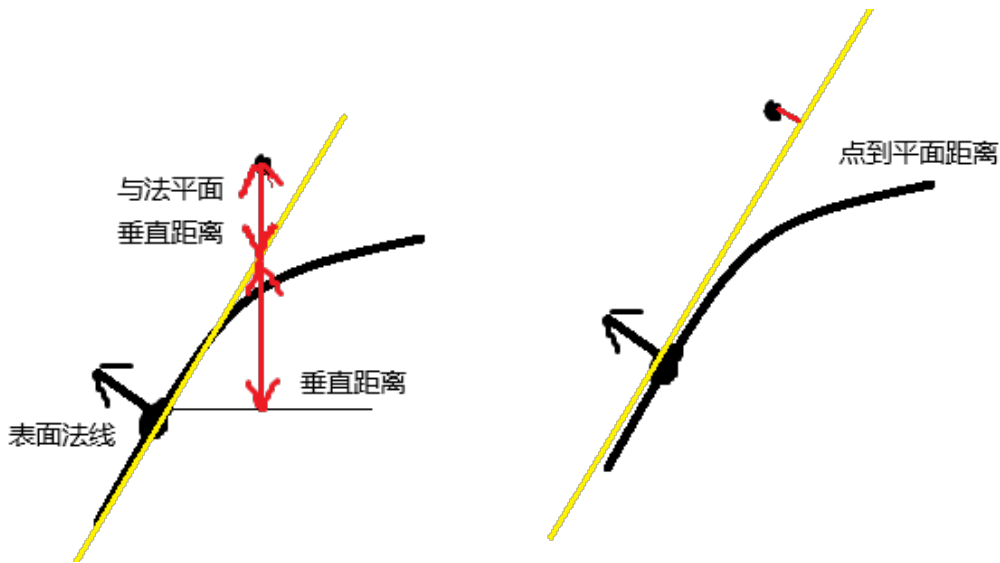


图 2. 垂直距离与点到平面距离的区别

(3) 几何视角：从几何角度来看，方法同时沿着相机光线和法线方向调整点和平面，以最小化点到平面距离的平方和。这个灵活的公式使得该方法能够轻松适应正射投影和透视投影的法线图，避免了棋盘伪影，并且对锐利特征或异常值具有较强的鲁棒性。

本文提出了一种基于点到平面距离的法线积分方法，通过联合优化点位置和局部平面位移来实现。这是首次基于点到平面距离进行法线积分的研究。

2 相关工作

2.1 非变分方法

比较早期的解决发现积分问题的非变分方法主要是路径积分 [7]，这些方法通过沿着路径计算法线或表面上的梯度信息，进而得到法线方向的积分。而对于某些具有缺陷而不可积的法线场（即梯度场，法线可通过对 SDF 求梯度得到），可将其梯度场投影到基函数的空间上强制表面可积性，进而进行法线的积分计算，这些基函数可以是：傅里叶基函数 [9]、余弦函数 [11]、正交向量 [8]、形状基函数 [10] 等。

近期出现了一种基于离散几何处理的发现积分方法（DGP） [11]，该方法直接从法线图恢复离散的多边形网格，这个方法不仅解决了法线图的离散化问题，还在后续工作中针对不连续性保持和非连接法线图问题进行了处理。该方法的创新在于通过离散几何技术，使法线数据的恢复更加精确，并能够处理一些不规则的输入数据。

2.2 变分方法

变分方法的核心思想是将法线积分问题转化为一个优化问题，并最小化一个函数 [1,2,4]，通常这个函数是与高度图（或表面）相关的某种形式的误差函数（如最小二乘误差）。在法线积分问题中，这个目标函数通常会包含法线场与恢复的表面之间的差异。一般来说，目标函数可以表示为 (1)：

$$J(z) = \int_{\Omega} \left[\left(\frac{\partial z}{\partial u} - \mathbf{p} \right)^2 + \left(\frac{\partial z}{\partial v} - \mathbf{q} \right)^2 \right] dudv \quad (1)$$

尽管变分法在法线积分问题中取得了显著的进展，但它们仍然存在一些局限性：对异常值较为敏感、计算复杂度高以及存在棋盘伪影。

3 本文方法

本文的主要思想是通过垂直于法线的局部平面来拟合曲面，我们将一个位于三维空间中的平面通过以下形式 (2) 表示：

$$\mathbf{p}^T \mathbf{n} + d = 0 \quad (2)$$

其中 \mathbf{p} 是表面上的点， \mathbf{n} 是曲面方向法向量， d 是曲面沿法线方向到原点的距离，我们现在有的是一系列的法向量，对于每一个法向量，我们需要求得它对应的 d ，使用最小二乘法进行优化，式子 (3) 如下：

$$\min \sum_{\mathbf{p} \in \mathcal{N}(\mathbf{p}(\mathbf{u}))} (\mathbf{p}^T \mathbf{n}(\mathbf{u}) + d(\mathbf{u}))^2 \quad (3)$$

其中 $\mathcal{N}(\mathbf{p}(\mathbf{u}))$ 表示位于像素坐标 \mathbf{u} 的点周围的点，如果联合考虑法线图中所有像素点，将得到以下式子，其与 planeSVD 中的式子 (4) 相同：

$$\min \sum_{\mathbf{u} \in \Omega_n} \sum_{\mathbf{p} \in \mathcal{N}(\mathbf{p}(\mathbf{u}))} (\mathbf{p}^\top \mathbf{n}(\mathbf{u}) + \mathbf{d}(\mathbf{u}))^2 \quad (4)$$

为了使用该式进行优化，需要提出两个假设：

(1) 假设对于三维到二维图像的投影已知，此时将 \mathbf{p} 记为 $\mathbf{p}(\mathbf{z}; \mathbf{u}_p)$ ，其中 \mathbf{u}_p 是像素空间中的点坐标， \mathbf{z} 则是点在三维空间中的坐标

(2) 假设在三维空间中邻近的点，被投影到二维后依然邻近

在上述两假设成立后邻近点集 $\mathcal{N}(\mathbf{p}(\mathbf{u}))$ 可记为 (5)：

$$\mathcal{N}(\mathbf{p}(\mathbf{u})) = \{\mathbf{p}(\mathbf{z}; \mathbf{u}_p) \mid \mathbf{u}_p \in \mathcal{N}(\mathbf{u})\} \quad (5)$$

优化公式也可更改为如 (6) 所示：

$$\min \sum_{\mathbf{u} \in \Omega_n} \sum_{\mathbf{p} \in \mathcal{N}(\mathbf{u})} (\mathbf{p}(\mathbf{z}; \mathbf{u}_p)^\top \mathbf{n}(\mathbf{u}) + \mathbf{d}(\mathbf{u}))^2 \quad (6)$$

接下来还存在一个问题，即如何确定周围的点 $\mathcal{N}(\mathbf{u})$ ，本文中给出了两种策略，如图-3所示：

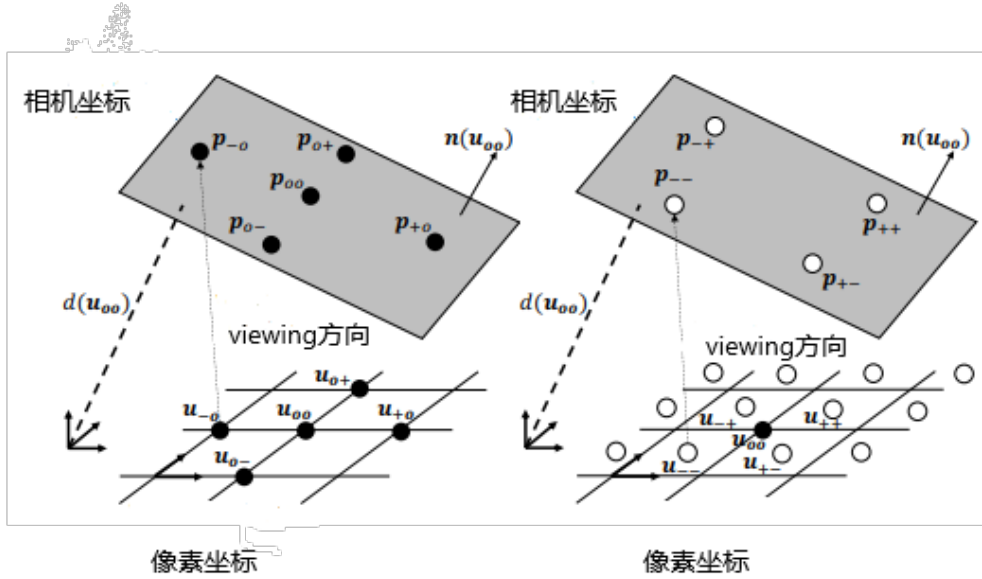


图 3. 五点平面拟合（左）与四点平面拟合（右）

(1) 让 $\mathbf{u}_p = \Omega_n$ ，这是传统变分方法的做法，意味着我们将在法线图的所有像素位置上进行求解。

(2) 将 \mathbf{u}_p 放置在图像中 \mathbf{u} 像素的半个像素步长处，如图右所示。这种做法与离散几何处理 (DGP) 方法中的做法相同，作者将其表示为 Ω_z 。

为了检索 \mathbf{u} 的最近邻点的投影，使用半径为 1 的球形查询 (ball query)。

对于 $\mathbf{u}_p = \Omega_n$ 的方法，本文中记为五点平面拟合，而对于 $\mathbf{u}_p = \Omega_z$ 的方法，本文中记为四点平面拟合。

对于五点平面拟合，优化公式可以简化为式 (7)：

$$\sum_{\mathbf{u}_p \in \mathcal{N}(\mathbf{u})} ((\mathbf{p}(\mathbf{z}; \mathbf{u}_p))^T \mathbf{n}(\mathbf{u}) + \mathbf{d}(\mathbf{u}))^2 = \left\| [\mathbf{p}_{oo} \mathbf{p}_{o+} \mathbf{p}_{o-} \mathbf{p}_{+o} \mathbf{p}_{-o}]^T \mathbf{n}(\mathbf{u}) + \mathbf{d}(\mathbf{u}) \mathbf{1} \right\|_2^2 \quad (7)$$

类似的，对于四点平面拟合，优化公式可简化为如式 (8) 所示：

$$\sum_{u_p \in \mathcal{N}(u)} ((\mathbf{p}(\mathbf{z}; u_p))^T \mathbf{n}(u) + \mathbf{d}(u))^2 = \left\| [\mathbf{p}_{++} \mathbf{p}_{+-} \mathbf{p}_{-+} \mathbf{p}_{--}]^T \mathbf{n}(u) + \mathbf{d}(u) \mathbf{1} \right\|_2^2 \quad (8)$$

接下来将对正交投影和透视投影分别阐述具体方法。

3.1 正交投影情况

在正射投影下，表面上的点 $\mathbf{p}(\mathbf{z}; \mathbf{u}_p)$ 被投影到图像平面上，并且在 x 和 y 轴方向上会有一个缩放因子 h ，可以得到式 (9)：

$$\mathbf{p}(\mathbf{z}; \mathbf{u}_p) = [\mathbf{h}\mathbf{u}_p, z]^T \quad (9)$$

其中 h 是将相机的 x - y 坐标转换为像素坐标的缩放因子。在有限差分的上下文中， h 被称为“步长” (step size)。在本论文中，为了简化符号，假设 $h=1$ 。

投影的关键是将 3D 表面点映射到 2D 图像平面。公式中的 $\mathbf{p}(\mathbf{z}; \mathbf{u}_p)$ 表示 3d 表面点， \mathbf{u}_p 是与该点相关的像素坐标。通过缩放， $\mathbf{p}(\mathbf{z}; \mathbf{u}_p)$ 被转换到像素坐标系。

将点的表示带入优化方程中，并将已知项移至方程右边，得到每个像素的线性方程系统如式 (10) 所示：

$$\begin{bmatrix} n_z & & & & 1 \\ & n_z & & & 1 \\ & & n_z & & 1 \\ & & & n_z & 1 \\ & & & & n_z & 1 \end{bmatrix} \begin{bmatrix} z_{oo} \\ z_{o+} \\ z_{o-} \\ z_{+o} \\ z_{-o} \\ z \end{bmatrix} = - \begin{bmatrix} u_{oo}^T \\ u_{o+}^T \\ u_{o-}^T \\ u_{+o}^T \\ u_{-o}^T \end{bmatrix} \begin{bmatrix} n_x \\ n_y \end{bmatrix} \quad (10)$$

将所有像素的方程堆叠起来，得到一个稀疏的过定方程系统如式 (11) 所示：

$$\mathbf{A} \begin{bmatrix} z \\ d \end{bmatrix} = \mathbf{b} \quad (11)$$

其中矩阵 \mathbf{A} 的行数约等于 $5|\Omega_n|$ ，即每个像素点附近的邻域大小的总和，列数为 $2|\Omega_n|$ ，对应 $|\Omega_n|$ 个深度值 z 和 $|\Omega_n|$ 个平面位移值 d 。

为了得到高度值和位移值的最优解，目标是通过最小化式 (12) 残差来找到解：

$$\begin{aligned} \min \|\mathbf{e}\|_2^2 \\ \text{s.t. } \mathbf{e} = \mathbf{A} \begin{bmatrix} \mathbf{z} \\ \mathbf{d} \end{bmatrix} - \mathbf{b} \end{aligned} \quad (12)$$

3.2 透视投影情况

在透视相机模型中，假设有一个针孔相机，且具有内参矩阵 K ，该相机将一个表面点 $p(z; u_p)$ 投影到图像平面上的像素 u_p 。对于每个像素 u_p ，相机射线（camera ray）会通过该像素点，并且与某个未知的深度 z 一起反向投影（back-projection）回到 3D 空间，如式 (13) 所示：

$$\mathbf{p}(z; \mathbf{u}_p) = z\mathbf{K}^{-1}\mathbf{u}_p \equiv z\mathbf{p} \quad (13)$$

其中是 \mathbf{u}_p 是像素坐标 u_p 的齐次坐标， \mathbf{p} 是经过反向投影后的点。

将点的表示带入优化方程中，得到每个像素的线性系统如式 (14) 所示：

$$\begin{bmatrix} \mathbf{n}^\top \tilde{\mathbf{p}}_{oo} & & & & & & 1 \\ & \mathbf{n}^\top \tilde{\mathbf{p}}_{o+} & & & & & 1 \\ & & \mathbf{n}^\top \tilde{\mathbf{p}}_{o-} & & & & 1 \\ & & & \mathbf{n}^\top \tilde{\mathbf{p}}_{+o} & & & 1 \\ & & & & \mathbf{n}^\top \tilde{\mathbf{p}}_{-o} & & 1 \\ & & & & & \mathbf{n}^\top \tilde{\mathbf{p}}_o & 1 \end{bmatrix} \begin{bmatrix} \mathbf{z}_{oo} \\ \mathbf{z}_{o+} \\ \mathbf{z}_{o-} \\ \mathbf{z}_{+o} \\ \mathbf{z}_{-o} \\ \mathbf{z} \end{bmatrix} = 0 \quad (14)$$

同样的，为了得到高度值和位移值的最优解，目标是通过最小化式 (15) 残差来找到解：

$$\begin{aligned} \min \|\mathbf{e}\|_2^2 \\ \text{s.t. } \mathbf{e} = \mathbf{A} \begin{bmatrix} \mathbf{z} \\ \mathbf{d} \end{bmatrix} \end{aligned} \quad (15)$$

3.3 关于矩阵 \mathbf{A} 的秩和解的歧义性

正交投影情况：在正交投影的情况下，系统矩阵 \mathbf{A} 是秩缺失的，即其秩小于列数。具体来说， $\text{nullity}(\mathbf{A}) = 1$ ，意味着解存在一个自由度或歧义性，这对应于一个偏移歧义。直观地讲，对于透视投影而言，将被投影的表面远离或靠近投影平面，其得到的结果是相同的，法线积分正是投影的逆过程，因此最终将得到无穷多组解。对于其解的高效计算，当系统矩阵 \mathbf{A} 变得非常大时，使用多重网格方法（如 Multigrid）来求解法线积分问题的正规方程会更加高效。多重网格方法能够在大规模稀疏矩阵中快速计算解。

透视投影情况：在透视投影的情况下，系统矩阵 \mathbf{A} 是满秩的，因为平面拟合的残差只有在表面本身就是平面时才可能为零。这种直觉是符合观察的：当表面为平面时，矩阵 \mathbf{A} 才会变为秩缺失。在透视投影情况下，优化方程是一个齐次方程，因此其解存在一个比例歧义，即解的唯一性只有在比例上，而非绝对值。对于其解的高效计算，本文采用了 ARPACK 库中的隐式重启 Lanczos 方法，这种方法能够高效地计算大规模稀疏矩阵的少数几个特征向量，特别适用于处理大规模矩阵。

4 复现细节

4.1 与已有开源代码对比

本篇论文已公开其源代码, 代码的 github 网址为: <https://github.com/xucao-42/NormalIntegration>。在本次复现过程中, 为了测试该方法的适用广泛性, 还使用了 wonder3d 扩散模型生成的数据进行测试, wonder3d 的 GitHub 网址为: <https://github.com/xxlong0/Wonder3D>, 该模型主要根据输入的单视图 RGB 图像生成多视图的法线图, 本次实验也会对这些生成的法线图数据进行测试。除此以外, 本次实验还会使用不同方法 (离散方函数、离散泊松方程、DGP 等) 对经典数据集 DiLiGenT 进行测试并对它们的结果进行对比, 并分析各方法之间的优劣。

4.2 实验环境搭建

本次实验在 windows11 上进行, 使用 CPU 为 Intel xeon w-2133, 使用 GPU 为 Nvidia GTX Titan X, 使用 Anaconda 构建虚拟环境, 首先需要在 GitHub 网站下载项目代码, 然后打开 cmd 运行如下代码以构建环境:

```
1 conda env create -f=environment.yml
2 conda activate ni
```

4.3 界面分析与使用说明

成功构建环境后, 本次实验分三部分进行, 分别是正交投影数据部分、透视投影数据部分、扩散模型生成数据部分。对于正交投影部分, 需运行如下代码:

```
1 python comparison_on_analytically_computed_orthographic_normal_maps
  .py
```

上述脚本比较了 3 种正交法线贴图上的 5 种方法并结果将保存在 results/TIME 中。若是需要在运行过程中添加离群点和噪点, 可以通过以下代码运行脚本以设置离群点和噪点强度用以研究不同方法对噪点和离群点的敏感程度:

```
1 python comparison_on_analytically_computed_orthographic_normal_maps
  .py --noise 0.1
2 python comparison_on_analytically_computed_orthographic_normal_maps
  .py --outlier 0.1
3 python comparison_on_analytically_computed_orthographic_normal_maps
  .py --outlier 0.1 --noise 0.1
```

若是需要对透视投影数据进行测试, 需要使用与正交投影不同的方法进行测试, 可以通过如下代码运行:

```
1 python comparison_on_perspective_diligent_normal_maps.py
```

上述代码运行脚本比较了 9 个 DiLiGenT 对象上的 6 种透视法积分方法。最后, 还可以将扩散模型生成的数据放入指点文件夹, 然后运行以下代码进行测试:

```
1 python methods/perspective_five_point_plane_fitting.py --path data/  
sample_data/sample.npy
```

5 实验结果分析

5.1 正交投影

本部分主要进行了不同方法在分别在各向异性高斯、球面以及花瓶面的法线图上的测试，测试结果如图-4所示。

图中从左到右分别是法线图、ground truth、离散函数、DGP、五点平面拟合、四点平面拟合、离散泊松方程方法重建得到的曲面，从图中我们可以较为直观地看到，离散函数方法的棋盘伪影十分明显，离散泊松方程和 DGP 对噪点较为敏感，表面存在较多坑洼，而本文的方法不仅没有棋盘伪影，生成的表面也足够平滑，且由于四点平面拟合相较于五点平面拟合多了一步插值，因此从图中能够观察到，四点平面拟合得到的平面光滑度更高，这点从花瓶数据的绝对误差图中也能够看出，图中从左到右分别为离散函数、DGP、五点平面拟合、四点平面拟合、离散泊松方程方法重建得到曲面的绝对误差图，其中颜色越接近蓝色则绝对误差越低，如图-5所示。

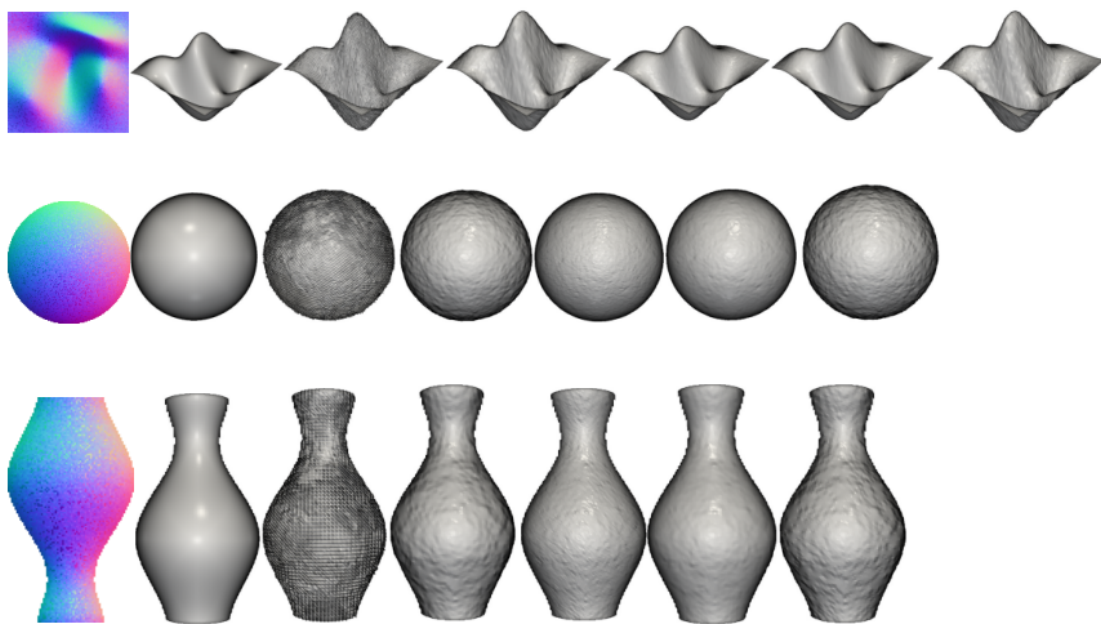


图 4. 各向异性高斯、球面、花瓶面的法线积分效果

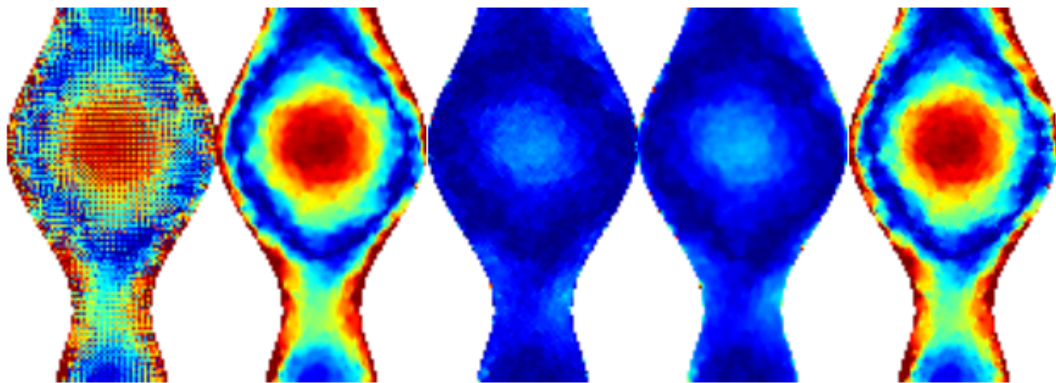


图 5. 花瓶面的法线积分绝对误差图

除此以外，还可以计算 MAE 和 RMSE 得到如下表所示的结果：

方法	五点拟合	四点拟合	离散泊松	离散函数	DGP
RMSE	0.0097166	0.00970851	0.019660	0.042828	0.01704
MAE	0.003859	0.00369	0.01223200	0.02737878	0.00920097
解决时间	0.1022448	0.1570978	0.05385	0.032913	0.0458777
总运行时间	0.1227042	0.165070	0.070319	0.037899	0.0678207

可以看到，本文提出的逆平面拟合方法相比于传统的离散泊松方程和 DGP 方法，在处理噪声、离群值和表面尖锐特征时更为鲁棒。

5.2 透视投影

本部分使用各类透视方法在数据集 DiLiGenT 上进行测试，其结果如图-6所示。从左到右分别是法线图、ground truth、离散函数、离散泊松、五点拟合、四点拟合，其中部分数据加入了噪点和离群点，可以看到离散函数方法受到离群点的影响较大（图中 cat 数据较为明显）且棋盘伪影非常明显，而离散泊松方法收到噪点的影响较大且其表面平滑度不够高，且 reading 数据集以及 harvest 数据集中也出现了较为明显的伪影，而本文提出的两种平面拟合方法则表现较好，然而查看绝对误差图，却会得到与上述完全不同的结果，如图-7所示。

方法	五点拟合	四点拟合	离散泊松	离散函数
RMSE	7.72552435	8.1020085	2.206921207	1.3416454
MAE	6.5596200	6.9080082	1.299540	0.86115073
解决时间	0.612776	1.047816	0.294477	0.3668243
总运行时间	0.6596491	1.1943991	0.3583872	0.462085

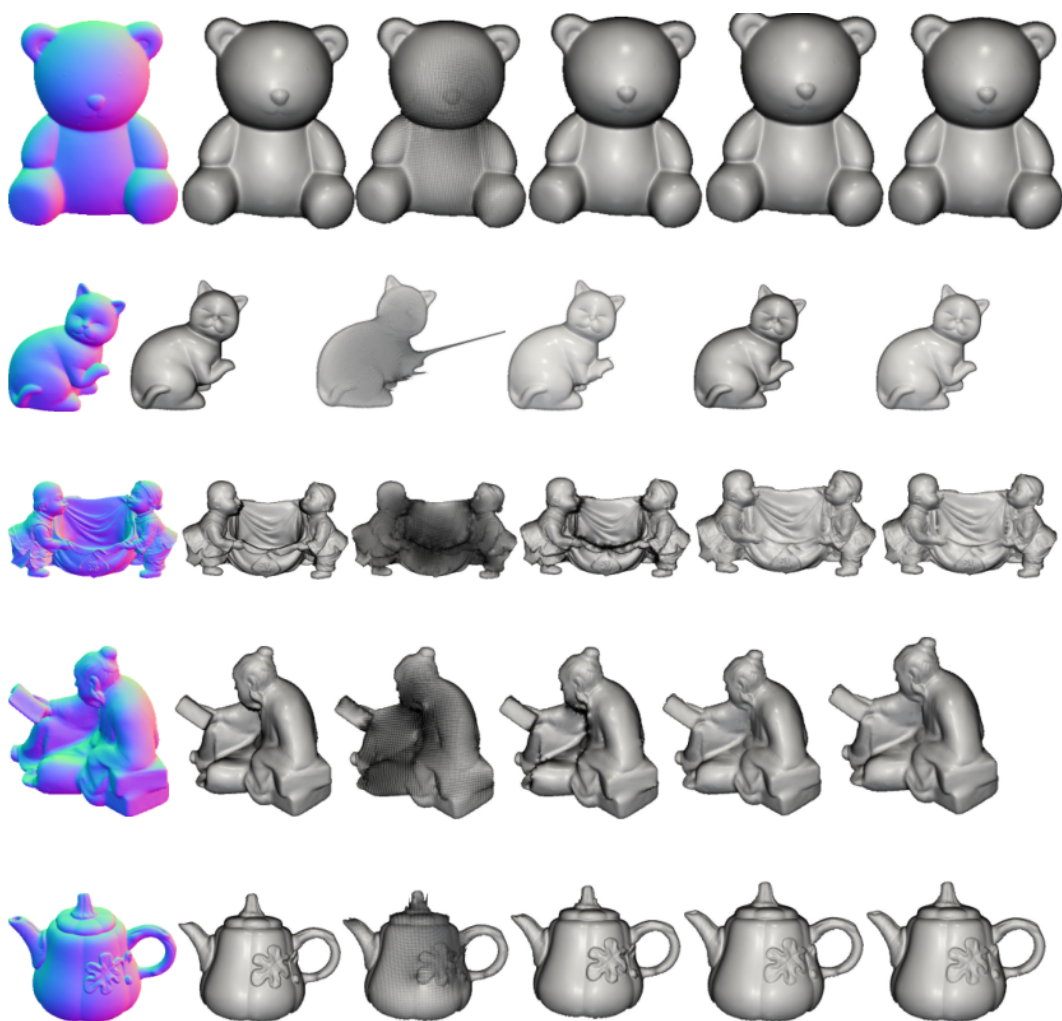


图 6. 各类透视方法在数据集 DiLiGenT 数据集测试结果

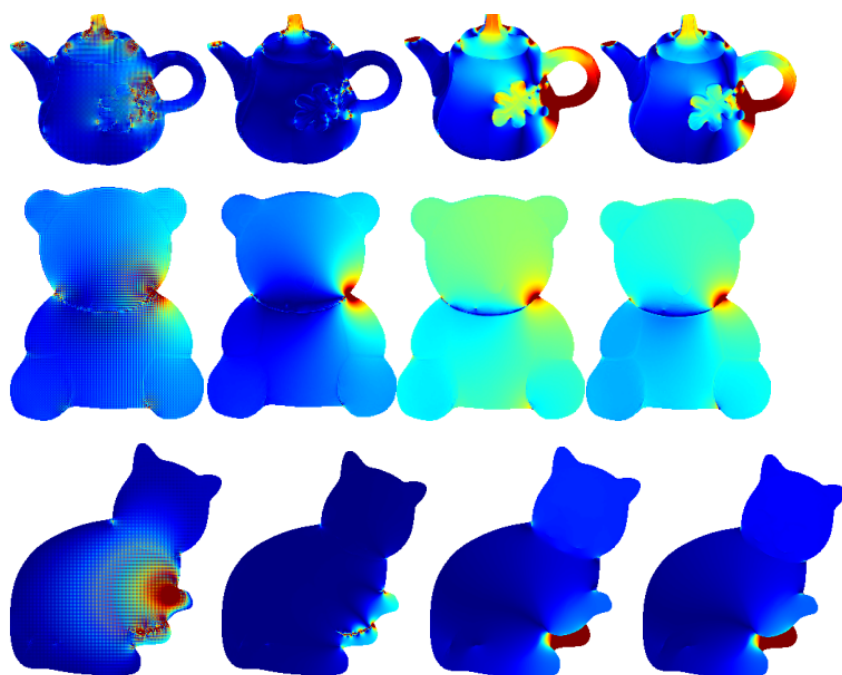


图 7. 透视投影法线积分绝对误差图

从左到右分别为离散函数、离散泊松、五点拟合、四点拟合的绝对误差图，出现了离散函数与离散泊松的绝对误差小于本文提出方法的情况，对于该现象，猜测有两种原因：

- (1) 逆平面拟合的插值平滑会将表面原有的凹凸也平滑掉，从而导致绝对误差增加
- (2) 在计算绝对误差时可能对于表面与 ground truth 未对齐或者缩放不正确导致计算得到的绝对误差完全不正确。

该问题将在今后的实验中继续进行探究。

5.3 扩散模型生成数据

本部分使用数据为通过 wonder3d 扩散模型生成数据，旨在通过该部分测试本文方法在扩散模型生成数据上能否正常运行。将扩散模型生成的法线图通过本文方法处理后得到的三维模型导入 blender 后得到如图-8所示结果：

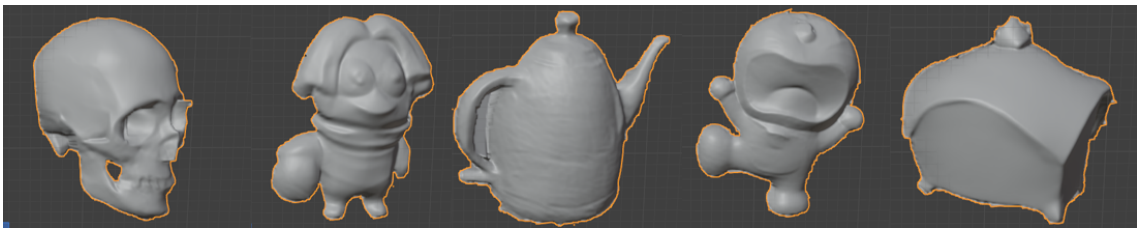


图 8. 扩散模型数据重建结果

从上述结果可以看到，扩散模型生成的法线图是能够被本文方法正常重建，但是这里也存在几个问题：

1. 由于生成的法线图的分辨率较低，因此生成的曲面高频细节几乎完全缺失，与原图相比差异较大。
2. wonder3d 生成的是多视图法线图，本文的重建方法无法完全利用所有的法线图

6 总结与展望

在本文中，作者提出、分析并评估了一种逆平面拟合方法，用于解决法线积分问题。与传统方法通过最小化垂直距离不同，本文的方法通过最小化点到平面的距离的平方和来构建线性系统。本文中提出的方法可以有效地降低棋盘伪影，且对离群点和噪点的敏感度较低，最终能够得到较为平滑的曲面。

对于文章的改进思路：尽管该方法在理论上具有较好的表现，但在某些情况下，离群值或噪声的处理可能仍然存在问题，尤其是在极端情况下（如强噪声、动态场景等），可以将深度学习引入到优化过程中，使用卷积神经网络（CNN）或者图神经网络（GNN）来辅助法线图的处理，进行噪声去除、特征提取和重建优化。例如，先通过深度网络对法线图进行初步修复，再使用逆平面拟合优化重建结果，或者使用生成对抗网络（GAN）生成更准确的表面。

参考文献

- [1] Zhouyu Du, Antonio Robles-Kelly, and Fangfang Lu. Robust surface reconstruction from gradient field using the l1 norm. In *9th Biennial Conference of the Australian Pattern*

- Recognition Society on Digital Image Computing Techniques and Applications (DICTA 2007)*, pages 203–209, 2007.
- [2] Matthew Harker and Paul O’Leary. Least squares surface reconstruction from gradients: Direct algebraic methods with spectral, tikhonov, and constrained regularization. In *CVPR 2011*, pages 2529–2536, 2011.
 - [3] Matthew Harker and Paul O’ leary. Regularized reconstruction of a surface from its measured gradient field: algorithms for spectral, tikhonov, constrained, and weighted regularization. *Journal of Mathematical Imaging and Vision*, 51:46–70, 2015.
 - [4] Berthold K.P Horn and Michael J Brooks. The variational approach to shape from shading. *Computer Vision, Graphics, and Image Processing*, 33(2):174–208, 1986.
 - [5] Katsushi Ikeuchi and Berthold KP Horn. Numerical shape from shading and occluding boundaries. *Artificial intelligence*, 17(1-3):141–184, 1981.
 - [6] Achuta Kadambi, Vage Taamazyan, Boxin Shi, and Ramesh Raskar. Polarized 3d: High-quality depth sensing with polarization cues. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.
 - [7] Richard B Lehoucq, Danny C Sorensen, and Chao Yang. *ARPACK users’ guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*. SIAM, 1998.
 - [8] Yvain Quéau, Jean-Denis Durou, and Jean-François Aujol. Normal integration: a survey. *Journal of Mathematical Imaging and Vision*, 60:576–593, 2018.
 - [9] D. C. Sorensen. Implicit application of polynomial filters in a k-step arnoldi method. *SIAM Journal on Matrix Analysis and Applications*, 13(1):357–385, 1992.
 - [10] Robert J Woodham. Photometric stereo: A reflectance map technique for determining surface orientation from image intensity. In *Image understanding systems and industrial applications I*, volume 155, pages 136–143. SPIE, 1979.
 - [11] Wuyuan Xie, Yunbo Zhang, Charlie CL Wang, and Ronald C-K Chung. Surface-from-gradients: An approach based on discrete geometry processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2195–2202, 2014.
 - [12] Dizhong Zhu and William AP Smith. Least squares surface reconstruction on arbitrary domains. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 530–545. Springer, 2020.