

# 泊松表面重建

## 摘要：

我们展示了对有向点集的表面重建可以转化为一个空间泊松问题。这种泊松公式化表达的方法同时考虑了所有的点，而不借助于启发式的空间分割或合并，于是对数据的噪声有很大的抵抗性。不像径向基函数的方案，我们的泊松方法允许对局部基函数划分层次结构，从而使问题的解缩减为一个良态的稀疏线性系统。我们描述了一个空间自适应的多尺度算法，其时间和空间复杂度正比于重建模型的大小。使用公共提供的扫描数据进行实验，在重建的表面，我们的方法比先前的方法显示出更详细的细节。

## 1、引言：

由点样本重建三维表面在计算机图形学中是一个热门研究问题。它允许对扫描数据的拟合，对表面空洞的填充，和对现有模型的重新构网。我们提出了一种重要的方法，把表面重建问题表示为泊松方程的解。

跟许多先前的工作一样(参见第2部分)，我们使用隐式函数框架来处理表面重建问题。特别地，像[Kaz05]我们计算了一个三维指示函数(在模型内部的点定义为1，外部的点定义为0)，然后可以通过提取合适的等值面获得重建的表面。

我们的核心观点是从模型表面采样的有向点集和模型的指示函数之间有一个内在关系。特别地，指示函数的梯度是一个几乎在任何地方都是零的向量场(由于指示函数在几乎任何地方都是恒定不变的)，除了模型表面附近的点，在这些地方指示函数的梯度等于模型表面的内法线。这样，有向点样本可视为模型的指示函数梯度的样本(如图1)。

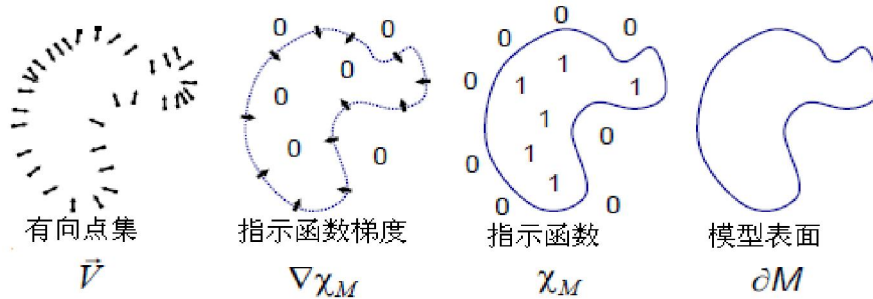


图1 二维泊松重建的直观图例

计算指示函数的问题因此简化为梯度算子的反算，即找到标量函数  $\tilde{\chi}$ ，使其梯度最佳逼近样本定义的向量场  $\vec{V}$ ，即， $\min_{\chi} \|\nabla \chi - \vec{V}\|$ ，如果我们使用散度算子，那么这个变化的问题转化为标准的泊松问题：计算标量函数  $\tilde{\chi}$ ，它的拉普拉斯算子(梯度的散度)等于向量场  $\vec{V}$  的散度，

$$\Delta \chi \equiv \nabla \cdot \nabla \chi = \nabla \cdot \vec{V}.$$

在第3、4部分我们将会对上式作精确的定义。

把表面重建问题表达成泊松问题拥有许多优点。很多对隐式表面进行拟合的方法把数据分割到不同区域以进行局部拟合，然后使用合成函数进一步合并这些局部拟合结果。相比之下，泊松重建是一个全局的方法，同时考虑了所有数据，而不借助于启发式的分割或合并。这样，类似于径向基函数(RBF)的方法，泊松重建创建了一个非常平滑的表面，它稳健地近似了含有噪声的数据。但是，由于理想的RBF都是全局支持和无退化的，而泊松问题允许局部支持的函数的层次结构，于是问题的解简化

为一个良态的稀疏线性系统。

此外，在许多隐式拟合方案中，隐式函数的值仅仅在样本点附近受到约束，造成重建物在远离样本点的地方可能包含伪平面片。通过引入辅助的外表面点，这种问题可以削弱（如：[CBC 01, CBA 03]）。在泊松表面重建中，这种表面片很少出现，因为隐式函数的梯度被全部的空间点所约束。特别地，在远离样本的地方，它被约束为 0。

泊松系统以它对不完整数据的抵抗性闻名。比如，“梯度域”处理算法（如 [FLW02]）有意地更改了梯度数据使得它不再跟任何势场相关，而仅仅依赖于泊松系统来恢复全局最佳拟合模型。

在解决泊松问题方面已经有广泛的跨学科的研究，提出了许多有效的、稳健的方法。在我们的问题实例中，一个特有的方面是我们只需要在重建表面的附近得到泊松方程的精确解。这就允许我们采取自适应的泊松算子来开发一个重建算法，使得算法的空间和时间复杂度正比于重建表面的大小。

## 2、相关的工作：

**表面重建** 由有向点集来重建表面在实践中有许多困难。采样点往往不均匀。由于采样的不精确和扫描的错误配准，点的位置和点的法线是有噪声的。在扫描过程中可达性约束使得一些表面区域缺乏数据。鉴于这些问题，重建方法试图推断未知表面的拓扑结构，精确的拟合（而不过度拟合）含有噪声的数据，并且合理的填充空洞。

许多方法都是基于组合结构的，比如 Delaunay 三角剖分（如 [Boi84, KSO04]），alpha 形状（alpha shapes）（[BM94, BBX95, BMR 99]）或者 Voronoi 图 [ABK98, ACK01]。这些方案一般通过对全部或者大部分点进行内插然后进行三角构网。在有噪声数据的场合，重建的表面往往是参差不齐的，因此需要在后处理中对点集进行平滑和修正。

另外的一些方案直接重建近似的表面，一般在隐式形式中表示。我们可以把它们大体分为全局方法和局部方法。

全局拟合法通常定义隐式函数为以样本点为中心的径向基函数（RBF）的和（如 [Mur91, CBC 01, TO02]）。然而，理想的 RBF 都是全局支持和无退化的，所以解矩阵是稠密且病态的。对大数据集的实用解法包括自适应的 RBF 简化和快速多极化方法 [CBC 01]。

局部拟合方法每次考虑一些邻近点的子集。一个简单的方案是估计切面并定义隐式函数为到最近点的切面的有符号距离 [HDD 92]。有符号距离可以累加为一个体积网格 [CL96]。为了保持函数的连续性，许多邻近点的影响可以相互调整，比如使用移动最小二乘法 [ABCO 01, SCS04]。另一种方法是通过自适应的对空间分块来形成点的邻近区，比如通过适应性的八叉树。通过对个体的多级分割，对八叉树的调整是可能的，而且在八叉树的每个节点中局部隐式面片的类型可以启发式的选择 [CBA 03]。

我们的泊松重建方法结合了全局和局部拟合方法的优点。它是全局的，于是在形成邻近区域、选择面片类型和调整时的权重时不涉及启发式的决策。但是，基函数是和周围空间而不是数据点相关的，是局部支持的，并且有一个层次结构，从而产生一个稀疏的良态系统。

我们解指示函数的方法跟基于傅里叶的重建方法类似 [Kaz05]。实际上，我们在附录 A 中展示了我们的泊松公式和它是等价的。虽然快速傅里叶变换（FFT）是解密集的周期性的泊松系统的常用方法，

但是，FFT 需要  $O(r^3 \log r)$  的时间复杂度和  $O(r^3)$  的空间复杂度，其中  $r$  是三维网格的分辨率，提高分辨率时，算法速度得到了抑制。相比之下，泊松系统支持自适应的离散化，因此产生一个可扩展的解决方案。

**泊松问题** 泊松方程出现在很多应用领域。比如，在计算机图形学中高动态范围图像的色调映射 [FLW02]，图像区域的无缝编辑 [RGB03]，流体力学 [LGF04]，和网格编辑 [YZX 04]。多重网格的泊松解法已经被改造应用于高效的 GPU 计算 [BFGS03, GVL 03]。

泊松方程也被用于热传导和扩散问题。有趣的是，Davis 等人[DMGL02]使用扩散来填充重建表面的空洞。以一个固定的有符号距离函数  $d$  的形式给定一个边界约束，他们的扩散方法实质上通过解均匀的泊松方程  $\Delta d = 0$  创建一个隐式面来生成边界。他们使用迭代法而不是全局、多尺度的泊松系统。

Nehab 等人[NRDR05]使用一个泊松系统对采样位置和采样法线拟合 2.5 维的高度场。他们的方法是拟合一个给定参数的表面，适合于个别的扫描数据的表面重建。然而，在样本来自于对模型的多次扫描的情况下，他们的方法不能直接应用于获得连贯的无缝表面。

### 3、我们的泊松重建方法：

输入数据  $S$  是一个样本集  $s \in S$ ，每个样本包含一个点  $s \cdot P$  和一个内法线  $s \cdot \vec{N}$ ，假设位于或者邻近一个未知模型  $M$  的表面为  $\partial M$ 。我们的目标是通过估计模型的指示函数和提取等值面，对表面重建一个无缝的三角逼近，如图 2。

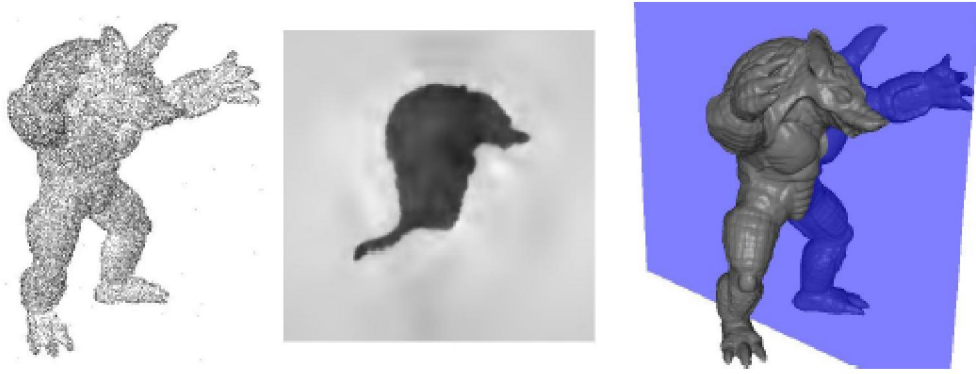


图 2 来自于对 Armadillo Man 模型(左)的扫描点，我们的泊松表面重建(右)，以及指示函数在一个三维空间的切面上的可视化表示(中)

我们的关键问题是根据样本精确地计算指示函数。在这一部分，我们推导了指示函数的梯度和曲面法线场的积分之间的一个关系。然后通过对给定有向点样本求和来近似计算这个曲面积分。最后，我们通过将这个梯度场作为一个泊松问题重建了指示函数。

**定义梯度场** 由于梯度函数是一个分段常函数，对梯度场的显式计算会导致在表面边界该向量场出现无限值。为了避免这种情况，我们用一个平滑滤波器和指示函数进行卷积，然后考虑平滑后的函数的梯度场。下面的引理形式化表示了平滑后的指示函数的梯度和曲面法线场的关系。

**引理：**给定一个三维实体  $M$ ，它的边界为  $\partial M$ ，用  $\chi_M$  表示  $M$  的指示函数， $\vec{N}_{\partial M}(p)$  表示  $P$  点 ( $p \in \partial M$ ) 的向内曲面法线， $\tilde{F}(q)$  为一个平滑滤波器， $\tilde{F}_p(q) = \tilde{F}(q - p)$  表示它到  $P$  点的平移。平滑后的指示函数的梯度和通过曲面法线场获得的向量场相等：

$$\nabla(\chi_M * \tilde{F})(q_0) = \int_{\partial M} \tilde{F}_p(q_0) \vec{N}_{\partial M}(p) dp. \quad (1)$$

**证明：**为了证明这个结论，我们展示了向量场的每个分量的相等性。计算平滑后的指示函数对  $x$  的偏导数，我们得到：

$$\begin{aligned}
\left. \frac{\partial}{\partial x} \right|_{q_0} (\chi_M * \tilde{F}) &= \left. \frac{\partial}{\partial x} \right|_{q=q_0} \int_M \tilde{F}(q-p) dp \\
&= \int_M \left( -\frac{\partial}{\partial x} \tilde{F}(q_0-p) \right) dp \\
&= - \int_M \nabla \cdot (\tilde{F}(q_0-p), 0, 0) dp \\
&= \int_{\partial M} \langle (\tilde{F}_p(q_0), 0, 0), \tilde{N}_{\partial M}(p) \rangle dp.
\end{aligned}$$

(第一个等式基于  $\chi_M$  在  $M$  外部为 0 而内部为 1。第二个等式基于  $(\partial/\partial q)\tilde{F}(q-p) = -(\partial/\partial p)\tilde{F}(q-p)$ 。最后一个等式基于散度定理。) 类似地，可以证明等式两边的  $y$  和  $z$  分量也对应相等，于是完成了证明。

**估计向量场** 毫无疑问，由于不知道模型表面的几何属性，我们不能近似计算曲面积分。然而，输入的有向点集提供了精确的、足够的信息，用来通过离散求和近似计算曲面积分。具体地，使用点集  $S$  把  $\partial M$  分割为不同的小面片  $\mathcal{P}_s \subset \partial M$ ，我们可以根据点样本  $s.p$  的值和小面片的面积的乘积近似计算小面片  $\mathcal{P}_s$  上的积分：

$$\begin{aligned}
\nabla(\chi_M * \tilde{F})(q) &= \sum_{s \in S} \int_{\mathcal{P}_s} \tilde{F}_p(q) \tilde{N}_{\partial M}(p) dp \\
&\approx \sum_{s \in S} |\mathcal{P}_s| \tilde{F}_{s.p}(q) s.\tilde{N} \equiv \tilde{V}(q).
\end{aligned} \tag{2}$$

应该指出的是，虽然等式 1 对任何平滑滤波器  $\tilde{F}$  都成立，但在实践中必须注意选择滤波器。具体来说，我们希望所选的滤波器满足两个条件。一方面，它必须充分的窄以便我们不会过度平滑数据；另一方面，它必须足够的宽使得在  $\mathcal{P}_s$  上的积分非常逼近于  $s.p$  的值与小面片面积的乘积。一个平衡了这两种条件的滤波器的好的选择是高斯滤波器，它的方差约等于采样分辨率。

**解泊松问题** 在形成向量场  $\tilde{V}$  之后，我们希望解得函数  $\tilde{\chi}$  从而使得  $\nabla \tilde{\chi} = \tilde{V}$ 。然而， $\tilde{V}$  通常是不可积的，换言之，它不是与路径无关(curl-free)的。这样，精确解一般不存在。为了找到最佳的最小二乘估计解，我们应用了散度算子来组成泊松方程。

在下一部分，我们更详细地描述了我们对这些步骤的具体实现过程。

#### 4、实现过程

我们首先基于点样本在模型表面均匀分布的假设表示我们的重建算法。我们定义了一个函数空间，它

在模型表面附近具有高分辨率，而在远离表面的地方具有低分辨率，把向量场  $\tilde{V}$  表达为在这个空间中的函数的线性总和，构造并解泊松方程，提取求取的指示函数的等值面。然后，扩展我们的算法使它能够处理样本点非均匀分布的情况。

#### 4.1 离散化问题

首先，我们选择函数空间，在函数空间中对问题进行离散化。最直接的方法是从一个规则的三维网格开始 [Kaz05]，但是这个均匀的结构在对细节的重建时变得不切实际，因为在该分辨率上空间的维度是三次的，而三角面的数量的增长是二次的。

幸运的是，我们只需要在重建表面的附近精确的表示隐式函数。这启发我们使用一个自适应的八叉树来表示隐式函数，然后解泊松系统(如 [GKS02,LGF04])。特别地，我们使用样本点的位置来定义八叉树 $\mathcal{O}$ ，然后附加一个函数 $F_0$ 给八叉树的每一个节点 $o \in \mathcal{O}$ ，选择八叉树和附加函数，使之满足下列条件：

- 1、 向量场 $\vec{V}$ 可以被精确有效的表示为 $F_0$ 的线性求和。
- 2、 泊松方程的矩阵表示按照 $F_0$ 能够有效求解的方式来表达。
- 3、 指示函数的表示，作为 $F_0$ 的和，能够在模型表面附近精确有效的估计。

**定义函数空间** 给定一个样本点集 $S$ 和最大树深 $D$ ，我们定义八叉树 $\mathcal{O}$ 为每个样本点都落在深度为 $D$ 的叶子节点上的最小八叉树。

然后，我们定义一个函数空间为可进行距离平移和尺度缩放、单位积分的基函数 $F: \mathbb{R}^3 \rightarrow \mathbb{R}$ 。

对每个节点 $o \in \mathcal{O}$ ，我们设定 $F_o$ 为单位积分的“节点函数”，它以节点 $o$ 为中心，以节点 $o$ 的大小展开：

$$F_o(q) \equiv F\left(\frac{q - o.c}{o.w}\right) \frac{1}{o.w^3}.$$

其中 $o.c$ 和 $o.w$ 分别是节点 $o$ 的中心和宽度。

这个函数空间 $\mathcal{F}_{\mathcal{O}, F} \equiv \text{Span}\{F_o\}$ 有一个多尺度的结构，类似于传统的小波表示。细节点跟高频函数相关，并且，当接近模型表面时，函数表示变得更加精确。

**选择基函数** 在选择基函数 $F$ 时，我们的目标是选择一个函数使得定义在等式 2 中的向量场 $\vec{V}$ 可以精确有效地表示为节点函数 $\{F_o\}$ 的线性求和。

如果我们要用包含样本点的叶子节点的中心来代替样本点的位置，那么向量场 $\vec{V}$ 能够有效的表示为 $\{F_o\}$ 的线性求和，通过设定：

$$F(q) = \tilde{F}\left(\frac{q}{2^D}\right).$$

这样的话，每个样本点都会贡献一项(法线向量)给与叶子节点函数相应的系数。由于采样间隔是 $2^{-D}$ 并且所有的样本点都落在深度为 $D$ 的叶子节点上，由固定(clamping)产生的误差不能太大(最多相当



于采样宽度的一半)。在下一部分，我们展示了通过三次线性插值法可以进一步减小误差以获得子节点的精度。

最后，由于最大树深  $D$  跟采样宽度  $2^{-D}$  相关，平滑滤波器应该近似于高斯滤波器，方差应该近似于  $2^{-D}$ 。这样， $F$  应该近似为一个单位方差的高斯滤波器。

为了提高效率，我们用一个简化的函数来近似单位方差滤波器使得 (1)计算的散度和拉普拉斯算子都是稀疏的，以及 (2)评价函数表达为  $F_0$  的线性求和，在有些点  $q$  处只需要对接近  $q$  的节点  $o \in \mathcal{O}$  求和。因此，我们设  $F$  为一个盒滤波器的  $n$  阶卷积，而它本身在基函数  $F$  中：

$$F(x,y,z) \equiv (B(x)B(y)B(z))^{*n} \quad \text{其中} \quad B(t) = \begin{cases} 1 & |t| < 0.5 \\ 0 & \text{其它} \end{cases}$$

注意到随着  $n$  的增大， $F$  更加逼近高斯滤波器，而且它的支持范围变得更大；在我们的实现过程中，我们使用了  $n=3$  的分段二次逼近。因此，函数  $F$  定义在作用域  $[-1.5, 1.5]^3$  中，并且对于八叉树的任意节点的基函数  $F$ ，最多有  $5^3 - 1 = 124$  个在相同深度上的其它节点的函数与之重叠。

## 4.2 定义向量场

为了达到子节点的精度，我们避免固定一个采样点的位置到包含采样点的叶子节点的中心，取而代之的是使用三次线性插值法分配样本点到八个最邻近的节点。这样，我们定义对指示函数的梯度场的近似值为：

$$\vec{V}(q) \equiv \sum_{s \in \mathcal{S}} \sum_{o \in \text{Ngbr}_D(s)} \alpha_{o,s} F_o(q) s \cdot \vec{N} \quad (3)$$

其中  $\text{Ngbr}_D(s)$  为最邻近  $s \cdot P$  的八个深度为  $D$  的节点， $\{\alpha_{o,s}\}$  为三次线性插值的权。(如果最邻近点不在八叉树中，我们对八叉树进行修改从而纳入最邻近点。)

由于样本是均匀的，所以我们可以假设面片  $\mathcal{P}_s$  的面积恒定不变，而  $\vec{V}$  是平滑后的指示函数梯度的好的近似值，最多差一个乘常数。我们将会展示乘常数的选择不会影响重建。

## 4.3 泊松解法

定义了向量场  $\vec{V}$  之后，我们希望求解函数  $\tilde{\chi} \in \mathcal{F}_{\mathcal{O},F}$  使得  $\tilde{\chi}$  的梯度最接近  $\vec{V}$ ，即泊松方程  $\Delta \tilde{\chi} = \nabla \cdot \vec{V}$  的一个解。

解  $\tilde{\chi}$  的一个问题是，虽然  $\tilde{\chi}$  和  $\vec{V}$  的坐标函数都在空间  $\mathcal{F}_{\mathcal{O},F}$  中，但是函数  $\Delta \tilde{\chi}$  和  $\nabla \cdot \vec{V}$  却不一定在该空间中。

为了处理这个问题，我们需要求解函数  $\tilde{\chi}$ ，使得  $\Delta \tilde{\chi}$  在空间  $\mathcal{F}_{\mathcal{O}, F}$  上的投影最接近  $\nabla \cdot \tilde{V}$  的投影。

由于，一般来说，函数  $F_0$  不形成正交基向量，直接解决这个问题代价较大。然而，我们可以通过求解

函数  $\tilde{\chi}$  的最小化来简化问题：

$$\sum_{o \in \mathcal{O}} \left\| \langle \Delta \tilde{\chi} - \nabla \cdot \tilde{V}, F_o \rangle \right\|^2 = \sum_{o \in \mathcal{O}} \left\| \langle \Delta \tilde{\chi}, F_o \rangle - \langle \nabla \cdot \tilde{V}, F_o \rangle \right\|^2.$$

这样，当给定  $|\mathcal{O}|$  维向量  $v$ ，它的第  $o$  个坐标为  $v_o = \langle \nabla \cdot \tilde{V}, F_o \rangle$ ，我们的目标是求解  $\tilde{\chi}$  使得投影  $\tilde{\chi}$  的

拉普拉斯算子到每个  $F_0$  得到的向量尽可能接近于  $v$ 。

为了在矩阵形式中表达上述关系，令  $\tilde{\chi} = \sum_o x_o F_o$  以便我们求解向量  $x \in \mathbb{R}^{|\mathcal{O}|}$ 。然后，我们

定义  $|\mathcal{O}| \times |\mathcal{O}|$  上的矩阵  $L$  使得  $Lx$  返回每个  $F_0$  和拉普拉斯算子的数量积。具体来说，对每个

$o, o' \in \mathcal{O}$ ， $L$  的  $(o, o')$  处的元素设为：

$$L_{o, o'} \equiv \left\langle \frac{\partial^2 F_o}{\partial x^2}, F_{o'} \right\rangle + \left\langle \frac{\partial^2 F_o}{\partial y^2}, F_{o'} \right\rangle + \left\langle \frac{\partial^2 F_o}{\partial z^2}, F_{o'} \right\rangle.$$

这样，解  $\tilde{\chi}$  相当于找到  $\min_{x \in \mathbb{R}^{|\mathcal{O}|}} \|Lx - v\|^2$ 。

注意到矩阵  $L$  是稀疏且对称的。（稀疏是因为  $F_0$  是经过简化的，对称是因为

$\int f'' g = - \int f' g'$ 。）此外，在空间  $\mathcal{F}_{\mathcal{O}, F}$  上有一个固有的多尺度结构，所以我们使用一种类似于

文献 [GKS02] 中多栅极的方法，算出  $L$  对深度为  $d$  的函数相应的空间的约束  $L_d$ （使用共轭梯度算子）

并且把固定深度的解投影回  $\mathcal{F}_{\mathcal{O}, F}$  以更新剩余的解。

**处理内存问题** 在实践中，随着深度的增加，矩阵  $L_d$  变大，把它存储在内存中可能不现实，虽然  $L_d$

每一列的元素个数被一个常数约束，但常数值可以很大。比如，即使使用一个分段二次基函数  $F$ ，在一列中有 125 个非零元素，会导致在内存中需要八叉树大小的 125 倍的空间。

为了解决这个问题，我们把我们的算子增强为高斯-赛德尔算子。也就是说，我们把第  $d$  层的三

维空间分解为重叠的区域，然后在不同的区域中解  $L_d$  的约束，把局部解投影回  $d$  维空间中并且更新

剩余解。通过选择区域数作为深度  $d$  的函数，我们确保算子所使用的矩阵的大小不会超过所需内存的阈值。

#### 4.4 等值面提取

为了获得重建表面 $\partial\tilde{M}$ ,有必要首先选择一个等值,然后通过计算指示函数提取对应的等值面。

我们选择等值使得提取的等值面密切逼近输入的样本点的位置。做法是通过在样本点的位置估计

计 $\tilde{\chi}$ 然后使用平均值来提取等值面：

$$\partial\tilde{M} \equiv \{q \in \mathbb{R}^3 \mid \tilde{\chi}(q) = \gamma\} \quad \text{with} \quad \gamma = \frac{1}{|S|} \sum_{s \in S} \tilde{\chi}(s.p).$$

选择的等值有一个属性：缩放 $\tilde{\chi}$ 不会改变等值面。这样，知道了向量场 $\vec{V}$  (即使差一个乘常数)就能够提供足够的信息重建模型表面。

为了从指示函数中提取等值面，我们对八叉树的表示 (如 [MG92, SFYC96, WKE99])使用一种类似于先前提出的改进的步进立方体 (Marching Cubes)算法[LC87]。然而，由于我们的八叉树不够理想，我们稍微修改了重建方法，根据与边邻近的最优层的节点计算的零交叉，来定义在该条边的零交叉点的位置。在一个叶子节点的对应边有多个零交叉与之相连时，把节点进行再分。跟先前的方法一样，当粗节点和精节点共面时，我们通过将精节点所在面的等值线段投影到粗节点所在面上，来避免产生裂缝。

#### 4.5 非均匀样本

我们现在把我们的方法扩展到样本点非均匀分布的情况。如[Kaz05]所述，我们的方法是通过估计局部采样密度，来衡量每个点相应的贡献率。然而，我们不是简单的调整与每个点相应的固定宽度中心的权重，而是同时也改变了中心宽度。这使得重建时在密集采样区保持了清晰的特征而在稀疏采样区提供了平滑拟合。

**估计局部采样密度** 根据[Kaz05]的方法，我们使用中心密度估计量[Par62]实现了密度计算。这种方法是估计样本邻近的点数，通过“泼洒”样本到三维网格中、把“泼洒”函数与一个滤波器进行卷积，然后评估在每个样本点处的卷积。

我们实现了在某种程度上与等式3类似的卷积。给定一个深度 $\hat{D} \leq D$ ，我们设密度估计量为深度为 $\hat{D}$ 处的节点函数的和：

$$W_D(q) \equiv \sum_{s \in S} \sum_{o \in \text{Ngb}_{\hat{D}}(s)} \alpha_{o,s} F_o(q).$$

由于八叉树在低分辨率处的节点与较宽的高斯估计函数相联系，参数 $\hat{D}$ 附加地指定了密度估计的位置，取值较小的 $\hat{D}$ 对应在较大的区域上给定的采样密度。

**计算向量场** 使用密度估计量，我们把等式3中的求和进行了转化，以使得每个样本的贡献率正比于它在面上对应的面积。具体而言，基于面积反比于采样密度的事实，我们设：

$$\vec{V}(q) \equiv \sum_{s \in S} \frac{1}{W_{\hat{D}}(s.p)} \sum_{o \in \text{Ngb}_{\hat{D}}(s)} \alpha_{o,s} F_o(q).$$

然而，仅仅调整样本贡献的权重会在稀疏采样区造成差的噪声滤波效果。如图7所示。于是，我们附加地调



整滤波器  $\tilde{F}$  的宽度为局部采样密度。调整滤波器宽度以保留在密集采样区域的细节，同时平滑掉在稀疏采样区的噪声。

基于深度较小的节点函数与较宽的平滑滤波器相联系这一事实，我们定义：

$$\vec{V}(q) \equiv \sum_{s \in S} \frac{1}{W_D(s.p)} \sum_{o \in \text{Ngb}_{\text{Depth}(s.p)}(s)} \alpha_{o,s} F_o(q).$$

在这个定义中， $\text{Depth}(s.p)$  表示采样点  $s \in S$  对应的深度。它通过计算所有样本的平均采样密度  $W$  来定义和设置：

$$\text{Depth}(s.p) \equiv \min(D, D + \log_4(W_D(s.p)/W))$$

使得平滑滤波器的宽度和  $s$  对  $\vec{V}$  的贡献率正比于与它相联系的面片  $P_s$  的半径。

**选择等值** 最后，我们通过选择等值，即  $\tilde{\chi}$  在样本位置的值的加权平均，把表面提取步骤进行修改：

$$\partial \tilde{M} \equiv \{q \in \mathbb{R}^3 \mid \tilde{\chi}(q) = \gamma\} \quad \text{with} \quad \gamma = \frac{\sum \frac{1}{W_D(s.p)} \tilde{\chi}(s.p)}{\sum \frac{1}{W_D(s.p)}}.$$

## 5、结果

为了评估我们的方法，我们进行了一系列实验。我们的目标是处理三个单独的问题：该算法重建表面的效果如何？如果跟其它重建方法比较？它的性能特点是什么？

表面重建的许多实际的动机来自于三维扫描，所以我们把实验的焦点集中在来源于真实世界数据的三维模型重建。

### 5.1 分辨率

我们首先考虑在重建表面上八叉树最大深度处的效果。

图3 展示了我们对“龙”模型分别在八叉树深度为6, 8, 10的重建结果。(在规则格网的重建中，它分别和分辨率  $64^3$ ,  $256^3$  和  $1024^3$  相联系。)随着树深的增加，高分辨率的函数用来拟合指示函数，因此重建捕捉了精密的细节。比如龙的鳞片，由于太过精细以致于在粗分辨率处无法捕捉，而当八叉树深度增加时这种效果急剧的明显起来。

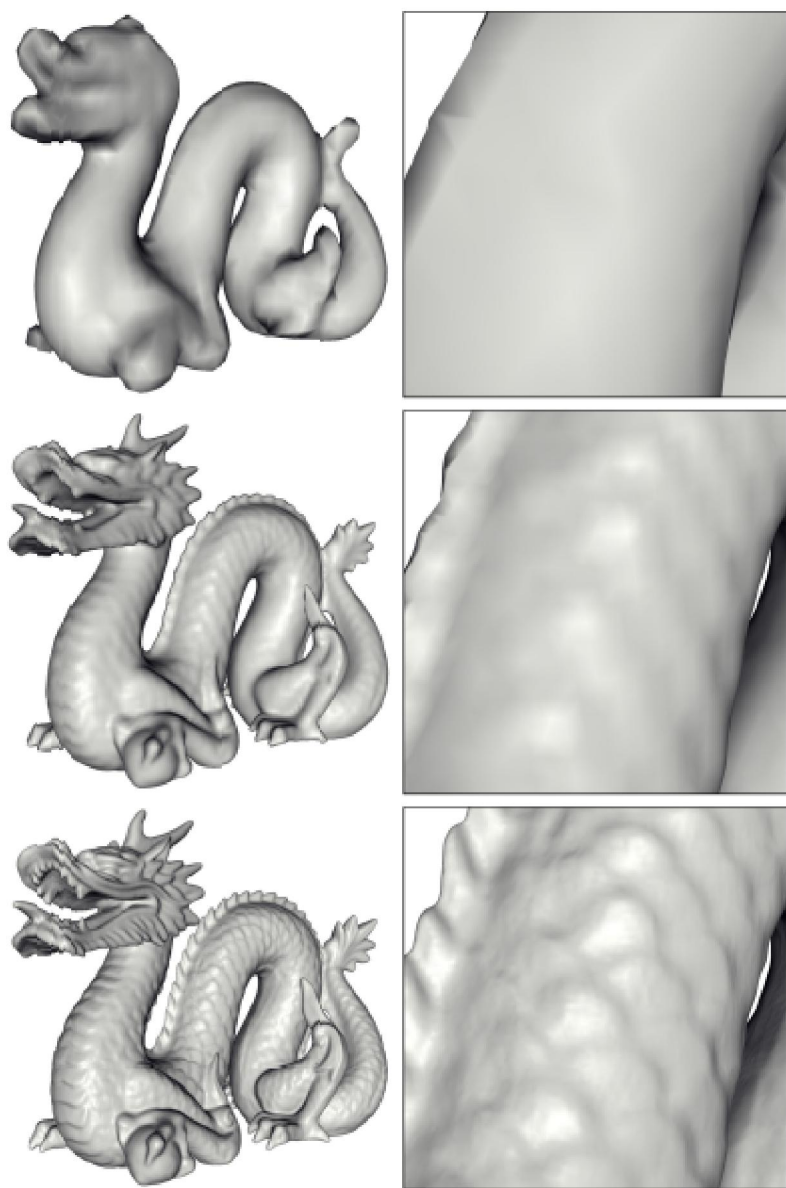


图 3 龙模型在树深分别为 6(上) ,8(中) ,10(下)时的重建效果

**我们方法的局限性** 我们的方法的局限性是它不引入跟模型形态相关的信息。图6展示了对佛像基座的重建的例子。由于在两脚之间没有样本，我们的方法(右)连接了这两个区域。相比之下，使用次要信息如视线VRIP(左)进行空间切分，用来分开两脚，可以产生更精确的重建模型。



图 6 “弥勒佛”模型的重建，使用 VRIP(左)和泊松重建(右)

## 6、结论

我们已经展示了表面重建可以被表示为一个泊松问题，它寻找一个指示函数，使之最佳近似一个有噪声的、非均匀观测的点集，并且论证了这种方法能够稳健的恢复有噪声的真实世界的扫描数据的精密细节。

我们未来的工作有三个方面的途径：

- l 利用样本的信任值扩展我们的方法。
- l 把扫描过程中的视线信息引入到解的过程中。
- l 扩展系统以支持对大数据集的out-of-core处理。

## 7、致谢

## 8、参考文献

## 9、附录 A：

这篇论文中描述的表面重建方法与[Kaz05]中的方法类似，因为二者重建的表面都是通过首先计算指示函数然后提取合适的等值面来获得的。

虽然这两种方法似乎用不同的方式计算指示函数（[Kaz05]使用斯托克斯定理来定义指示函数的傅里叶系数而我们使用泊松方程），但这两种方法实际上是等价的。

为了表达它，我们利用泊松方程  $\Delta u = f$ ，其中  $f$  是周期性的，可以通过傅里叶变换来求解。

傅里叶级数展开式为  $-|\zeta|^2 \hat{u}(\zeta) = \hat{f}(\zeta)$ ，或者等价地， $\hat{u}(\zeta) = \frac{-1}{|\zeta|^2} \hat{f}(\zeta)$ 。

这样，我们的泊松方程  $\Delta \chi = \nabla \cdot \vec{v}$  可以通过  $\hat{\chi} = \frac{-1}{|\zeta|^2} \widehat{\nabla \cdot \vec{v}}$  来求解。由于众所周知的同一性  $\hat{f}' = -i\zeta \hat{f}$  以及它的推广  $\widehat{\nabla \cdot \vec{v}} = -i\zeta \cdot \hat{\vec{v}}$ ，我们得到  $\hat{\chi} = \frac{i}{|\zeta|^2} \zeta \cdot \hat{\vec{v}}$ ，而它与[kaz 05]中的等式是相同的。