EUROGRAPHICS 2023 / K. Myszkowski 和 M. Nießner（客座编辑）

计算机图形学论坛 第 *42* 卷（*2023*），第 *2* 期

arXiv:2303.05965v1 [cs.GR] 10 Mar 2023

稠密网格上的可扩展且高效的函数映射计算

RobinMagnet[1] Maks Ovsjanikov[1]

[1]LIX，巴黎综合理工学院，巴黎理工学院

摘要

我们提出了一种新型、可扩展的函数映射处理流程，可高效计算可能极为稠密的网格之间的对应关系。不同于现有依赖临时网格简化来处理稠密网格的方法，我们建立了一套集成式端到端处理流程，并对其进行理论近似分析。具体而言，我们通过近似函数空间和函数映射本身，克服了计算基以及函数对应和逐点对应时的计算负担。我们利用理论上界控制近似误差，并据此评估该处理流程的适用范围。基于这一构造，我们提出了一种可扩展的实用算法，并展示了在稠密网格上的结果；这些结果与标准函数映射算法所得结果近似，而计算时间仅为其一小部分。此外，我们的方法大幅优于标准加速流程，即使在具有挑战性的情况下也能得到准确结果。

CCS 概念

• 计算方法 → 形状分析；•计算理论 → 计算几何；

1. 引言

复杂三维物体的处理与分析是一个重要研究领域，应用范围涵盖计算机图形学、医学影像及其他领域。这类数据的底层结构可能十分精细，因此需要稠密的点集和网格来捕捉重要特征。与此同时，形状分析方法通常只针对由数万个点构成的物体，因此，对于现实应用中可能出现的包含数百万个点的网格，需要采用抽稀算法进行处理。尽管网格简化能够取得良好结果，但也存在若干缺点。首先，简化过程可能产生伪影，并造成细节显著丢失。其次，在许多应用中，将分析结果从简化后的形状准确迁移至原始形状仍然极具挑战。最后，迁移过程可能引入误差和混叠伪影。

在本文中，我们专注于计算非刚性形状之间的对应关系。这是几何处理及相关领域中一个由来已久的问题，过去几年中人们已开发出大量技术 [DYDZ22，Sah20]。该领域一项值得关注的研究路线采用所谓的函数映射框架，通过在约化基中将对应关系表示为矩阵并对其进行操作 [OBS*12]。基于这一框架的方法近年来在多种具有挑战性的非刚性形状匹配任务中取得了很高的准确率 [MMR*19，MRR*19，DLR*20]。遗憾的是，这些方法需要耗费大量成本和时间预先计算拉普拉斯基，并且可能还要计算其他辅助量-

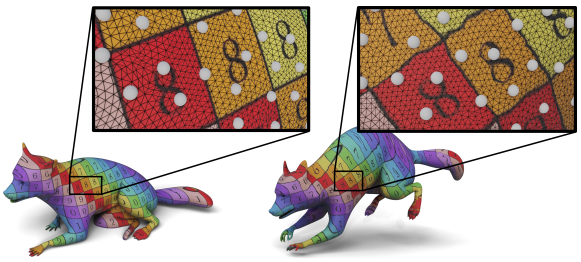

图 1： 我们的方法能够高效地计算稠密网格之间的点对点对应关系，仅使用位于稀疏采样点处的值，这些采样点以白色显示。来自 DeformingThings4D 数据集 *[LTT*21]* 中的源形状和目标形状由大约 100 000 个顶点组成，并通过纹理迁移显示对应关系。映射计算（包括所有预处理）在一台标准计算机上耗时 *60* 秒。

iary 数据结构 [RPWO19]。因此，这些技术难以扩展到密集采样的网格，故通常仅用于简化后的形状。此外，尽管近期已有针对部分方法提出的加速版本 [MRR*19]，但这些方法缺乏理论近似保证，且容易出错。

与此同时，近期已有多种方法被提出，用于高效近似拉普拉斯-贝尔特拉米基

© 2023 Eurographics - 欧洲计算机图形学协会和 John Wiley & Sons Ltd.

[NBH18，NH22]。这些方法能够成功扩展至超大网格，在计算低频特征函数时尤为有效。虽然已有研究表明，这些方法在例如将近似谱用作形状描述符时 [RWP06] ，或用于单个形状处理时效率很高，但应用于形状对应场景时，效果可能不尽如人意。从概念上讲，这是因为 [NBH18，NH22] 中的目标和保证仅适用于单个形状的全局尺度，而非函数映射计算和点到点映射计算所需的局部函数近似或函数传递。

在这项工作中，我们朝着创建可扩展且高效的非刚性形状对应方法迈进一步。这些方法能够处理超大规模网格，并具有理论近似界作为支撑。我们聚焦于函数映射框架 [OCB*17], 尤其是其近期基于谱上采样的变体，例如 ZoomOut 方法 [MRR*19] 及其后续工作 [HRWO20、XLZ21、RMWO21]。这些方法通过迭代更新函数映射和点到点映射，在多种情况下都能取得高质量结果。遗憾的是，其中两个主要步骤——基预计算和逐点映射的迭代更新——在稠密形状上计算代价高昂。

为应对这一挑战，我们提出了一种集成流水线，使这两个步骤都具备可扩展性，并且还提供近似保证。为此，我们首先构建了一个受 [NBH18], 启发的新函数空间，并展示如何利用它定义函数映射的近似，而无需稠密逐点对应，也无需在稠密网格上使用基。随后，我们为这一构造给出理论近似界；与 [OBS*12] 中的原始定义不同，该构造完全不受原始网格点数的影响。在此分析的基础上，我们扩展了 [NBH18] 中提出的方法，以改进函数映射近似，并提出了一种基于我们构造的高效、可扩展映射细化算法。正如图 2所示，该算法最终能以标准处理所需时间的一小部分生成准确结果。

2. 相关工作

我们的主要目标是在函数映射框架下，为非刚性形状对应设计一种可扩展且有理论依据的方法。因此，我们回顾与我们的方法最为密切相关的工作，尤其是那些使用谱技术进行形状匹配的研究，并请感兴趣的读者参阅近期综述 [VKZHCO11，TCL*12，BCBB16，Sah20，DYDZ22] ，以全面了解其他方法。

形状匹配中的谱方法 在许多已有研究中，人们探讨了利用拉普拉斯-贝尔特拉米算子的谱性质，尤其是其特征函数进行形状对应的思路。早期方法侧重于直接对齐特征函数，并将其视为描述符， [MHK*08，JZVK07] 或者利用扩散过程推导描述符或嵌入空间，例如[SH10，OMMG10], 等。

在 [OBS*12]中提出了一种更具理论依据的框架，其基础是函数映射的思想。总体策略是将函数的拉回表示为约化基上的算子，并根据此类算子的理想性质构建目标函数。该方法的主要优势在于，它能够形成规模较小的优化问题，且未知数的数量与底层网格的大小无关。

尽管原始方法简单，但其性能高度依赖于准确的描述符和超参数调优。因此，许多后续工作都在这一基本策略上进行了显著扩展，既利用几何洞见 [KBB*12，AK13，OMPG13，BDK17，ERGB16], 改进优化策略 [KGB16，NO17，RMOW20，RMWO21], , 也采用了更丰富的对应模型，使其不再局限于完整形状之间的等距变换， [RCB*17，ROA*13，LRBB17], 等等。

函数映射与逐点映射 尽管函数映射领域的许多方法都侧重于谱域中的优化，但研究也发现， 逐点对应关系与函数对应关系之间的交互在实践中能够带来显著改进。最初的论文及后续研究 [OBS*12,MDK*16,OMPG13] 将这一思路用于迭代最近点（ICP）细化，此后又将其扩展到映射去模糊和去噪 [EB17], , 以及强大的细化方法，乃至映射优化策略 [MRR*19, RPWO19,HRWO20, ELC20,RMWO21]。所有这些研究都基于这样一个认识：在谱域和空间（原始）域中同时操作映射，能够整体提升结果的质量。

遗憾的是，这类方法往往会牺牲可扩展性，因为逐点映射的复杂度直接取决于网格分辨率，因此难以扩展到高度稠密的网格。

多分辨率谱方法我们的工作也与用于近似谱量的多分辨率技术有关，例如 [VBCG10], , 尤其与近期针对复杂网格上的拉普拉斯特征函数、旨在实现精确且可扩展的特征求解器的进展有关 [NBH18，NH22]。研究表明，后一类方法在涉及单个形状的任务中表现出色且具有良好的可扩展性，例如计算其 Shape-DNA [RWP06]描述符，或执行网格滤波。类似地，也有一些谱粗化与简化方法 [LJO19，LLT*20，CLJL20] ，这些方法明确致力于粗化算子（如拉普拉斯算子），同时保留其低频特征对。遗憾的是，这些方法通常依赖于稠密形状上的特征函数，而前述方法在函数映射领域中的实用性尚未得到充分分析和利用，部分原因是，正如下文所示，这需要局部近似界。

最后，我们提到，我们的工作也与层次化技术相关，包括 [SVBC19], 中提出的细分曲面之间的函数映射 [MRR*19], 以及与通过谱上采样进行细化的关系更为密切的工作。然而，前一种方法依赖于细分层次结构，而后一种方法的加速策略（我们将在下文讨论）则基于一种方案；遗憾的是，在存在完整信息的情况下，该方案可能无法收敛。

现有技术的局限性与我们的贡献总而言之，现有基于函数映射的方法的可扩展性通常受两个因素限制：首先是预处理

© 2023 Eurographics - 欧洲计算机图形学协会和 John Wiley & Sons Ltd.

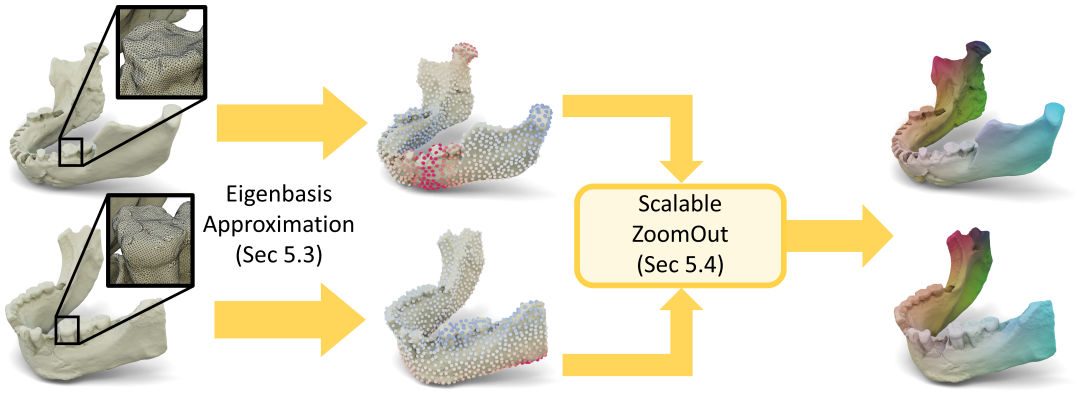

图 2： 我们方法的整体处理流程，使用来自 *[RWHO20]* 的真实数据。给定两个稠密输入形状，我们首先采用 *[NH22]*（第 *5.3* 节）中介绍方法的改进版本，生成近似的特征基计算结果。随后，我们提出 *ZoomOut* 的一种新型可扩展版本（第 *5.4* 节），该方法利用我们的函数映射近似（第 *5.1* 节），并具有理论近似误差界。最终，我们得到了两个输入形状之间的稠密逐点对应关系，并通过颜色传递将其可视化。

计算拉普拉斯-贝尔特拉米算子特征函数的成本；其次是同时处理逐点对应关系和函数对应的复杂度。

在此背景下，我们的主要贡献包括：

1. 我们定义了函数映射的近似，它只需要稀疏对应，并为这一构造提供了理论依据。

2 我们分析了 [NBH18] 中用于函数映射计算的基近似方法，得出了明确的理论上界。随后，我们对该方法进行改进，以提升近似保证，从而得到更准确的映射。

3. 我们基于所提出的构造，给出了一种原理清晰且可扩展的函数映射细化算法；与同类方法相比，该算法只需很短的时间便能得到精确的结果。

3. 方法概述

如上所述，我们的总体目标是设计一种可扩展的非刚性形状匹配处理流程，能够处理可能非常稠密的网格。我们的方法基于函数映射框架的 ZoomOut 变体 [MRR*19]。不过，我们的构造也可以轻松扩展到其他近期的函数映射方法，例如 [RMOW20、RMWO21],，这些方法具有相同的一般算法结构。具体而言，ZoomOut 及相关方法基于两个主要构件：首先计算拉普拉斯-贝尔特拉米算子的特征函数，然后迭代更新逐点映射和函数映射。

我们的总体处理流程如图 2 所示，主要包括以下步骤。首先，我们通过对 [NBH18], 中第 5.3 节介绍的方法进行修改，为每个形状生成一个稀疏样本集和一个因子化函数空间。其次，我们使用所提出的函数映射近似方法（第 5.1 节），定义一种可扩展的 ZoomOut 算法，以生成稀疏逐点映射。最后，利用所考虑的函数子空间的性质，将该稀疏映射扩展为具有亚采样精度的稠密逐点映射。

本文其余部分安排如下：第 4 节介绍我们的方法所需的记号和背景知识。

在第5.1节中，我们介绍基于 [NBH18]中的基构造方法的函数映射近似。第5.2 节给出了明确的近似误差，第5.3 节介绍了我们对 [NBH18], 方法所做的修改，该修改有助于改进我们为函数映射计算推导出的理论上界。基于这些构造，我们在第5.4 节展示如何定义类似 ZoomOut 的算法：先在约化函数空间中迭代更新函数映射和逐点映射，再将计算得到的函数映射高效地扩展到稠密形状上。

第5.5 节介绍实现细节，而第6 节则专门对我们的方法进行全面的实验评估。

4. 符号与背景

4.1. 符号

对于三角形网格，我们用$\mathbf{W}$ 和$\mathbf{A}$ 分别表示其刚度矩阵和质量矩阵，两者共同定义（半正定的）拉普拉斯-贝尔特拉米算子：$L = \mathbf{A}^{-1}\mathbf{W}$ 。给定两个形状 $\mathcal{N}$和 $\mathcal{M}$，它们分别有n 个和m 个顶点，任意顶点间映射$T : \mathcal{N} \to \mathcal{M}$ 都可以表示为一个二值矩阵 $\mathbf{\Pi} \in \{0,1\}^{n\times m}$，其中$\mathbf{\Pi}_{ij} = 1$当且仅当$T(x_i) = y_j$，其中$x_i$表示 $\mathcal{N}$上的第i个顶点，而y_j表示 $\mathcal{M}$中的第j个顶点。

拉普拉斯-贝尔特拉米算子的特征函数可以通过求解广义特征值问题获得：

$$\mathbf{W}\psi_i = \lambda_i \mathbf{A}\psi_i, \tag{1}$$

实际应用中，我们通常考虑对应于K 个最小特征值的特征函数。

4.2. 函数映射与 ZoomOut 算法

函数映射是在 [OBS*12] 中提出的，旨在实现稠密的非刚性形状匹配。其核心思想是，任意

© 2023 Eurographics - 欧洲计算机图形学协会
和 John Wiley & Sons Ltd.

逐点映射 $T: \mathcal{N} \to \mathcal{M}$ 都可以通过复合映射转化为函数映射 $F_T: f \in \mathcal{F}(\mathcal{M}) \mapsto f \circ T \in \mathcal{F}(\mathcal{N})$，其中 $\mathcal{F}(\mathcal{S})$ 是曲面 $\mathcal{S}$ 上实值函数的空间。由于 F_T 是线性的，因此可以在各空间 $\left(\psi_i^{\mathcal{M}}\right)_i$ 和 $\left(\psi_i^{\mathcal{N}}\right)_i$ 的给定基下表示为矩阵。

如果形状 $\mathcal{N}$ 上的基相对于 $\mathbf{A}^{\mathcal{N}}$ 是正交归一的，那么函数映射 $\mathbf{C}$ 可以在每个形状上大小为 K 的截断基中表示为一个 $K \times K$ 矩阵：

$$\mathbf{C} = \left(\mathbf{\Psi}^{\mathcal{N}}\right)^{\top} \mathbf{A}^{\mathcal{N}} \mathbf{\Pi} \mathbf{\Psi}^{\mathcal{M}}, \tag{2}$$

其中，$\mathcal{M}$ 上的每个基函数（$\mathcal{N}$ 上的基函数亦然）都作为一列排列在 $\mathbf{\Psi}^{\mathcal{M}}$（$\mathbf{\Psi}^{\mathcal{N}}$）中，$\mathbf{\Pi}$ 是表示底层逐点映射的矩阵，而我们用 $^{\top}$ 表示矩阵转置。

ZoomOut 给定拉普拉斯-贝尔特拉米特征基，ZoomOut 算法 [MRR*19]可通过在以下两个步骤之间迭代，从近似初始化开始恢复高质量对应关系：(1) 将一个 $k \times k$ 函数映射转换为逐点映射；(2) 将逐点映射转换为大小为 $k+1 \times k+1$ 的函数映射。此外，该方法也已扩展到其他设置，例如促进循环一致性 [HRWO20] 以及优化各种能量函数 [RMWO21] 等。遗憾的是，尽管这种方法简单高效，其可扩展性仍受到限制：首先，拉普拉斯基的预计算会造成限制；其次，逐点映射的恢复依赖于在稠密网格上执行可能代价高昂的最近邻搜索。

已有研究提出了若干临时性的加速策略[MRR*19]。然而，正如下文所述，这些策略并不具备近似保证；事实上，即使信息完整，它们也可能无法收敛。

4.3. 特征基近似

为提高谱方法的可扩展性，近期研究 [NBH18,XLHH21] 尝试通过缩小搜索空间，构造拉普拉斯-贝尔特拉米特征基的近似。具体而言，在 [NBH18],中，作者首先在形状 $\mathcal{M}$ 上采样一组 p 个点 $\mathcal{S} = \{v1, \ldots, v_p\}$，并构造一组 p 个局部函数（$u1, \ldots, u^p$），每个函数以一个特定采样点为中心。每个函数 u_j 由一个未归一化函数 $\tilde{u}_j$ 构造而成，该函数的支撑位于以采样点 vi 为中心、半径为 ρ 的测地球上，并随其到中心的测地距离增大而衰减：

$$\tilde{u}_j : x \in \mathcal{M} \mapsto \chi_\rho \left(d^{\mathcal{M}}(x, v_j) \right) \in \mathbb{R} \tag{3}$$

其中 $d^{\mathcal{M}}$ 表示形状 $\mathcal{M}$ 上的测地距离，且 $\chi_\rho : \mathbb{R}_+ \to \mathbb{R}$ 是一个可微的非递增函数，满足 $\chi_\rho(0) = 1$ 且 $\chi_\rho(x) = 0$ 对于 $x \geq \rho$ 而言。附录 χ A讨论了 ui 的选取。最后，通过以下方式定义局部函数，使其满足单位分解：

$$u_j(x) = \frac{\tilde{u}_j(x)}{\sum_k \tilde{u}_k(x)} \quad \forall\, x \in \mathcal{M} \tag{4}$$

现在只考虑属于张成空间 $\{u_1, \ldots, u_p\}$ 的函数，方程 (1) 中原有的特征分解系统可化简为一个规模为 $p \times p$ 的广义特征值问题：

$$\overline{\mathbf{W}}\, \bar{\phi}_i = \bar{\lambda}_i \overline{\mathbf{A}}\, \bar{\phi}_i \tag{5}$$

其中 $\overline{\mathbf{W}} = \mathbf{U}^{\top} \mathbf{W} \mathbf{U}$，且 $\overline{\mathbf{A}} = \mathbf{U}^{\top} \mathbf{A} \mathbf{U}$，其中 $\mathbf{W}$ 和 $\mathbf{A}$ 分别是 $\mathcal{M}$ 的刚度矩阵和面积矩阵，而 $\mathbf{U}$ 是一个稀疏矩阵，其列为函数 $\{u_j\}_j$ 的取值。特征向量 ϕ_i 是描述相对于 $\{u_j\}$ 的系数的 p 维向量，它们将近似特征向量定义为 $\psi_i = \mathbf{U}\phi_i$。注意，由于 ϕ_i 相对于 $\mathbf{A}$ 是正交归一的，因此这意味着 ψ_i 相对于 $\mathbf{A}$ 也是正交归一的。

尽管原始工作 [NBH18] 专注于针对单个形状的全局应用，例如滤波和 Shape-DNA [RWP06] 计算，但我们在此处理流程的基础上进行改进，以获得可靠的函数来执行稠密形状对应。

5. 我们的方法

本节首先介绍一种函数映射定义，该定义采用 [NBH18], 中的基近似策略，并给出近似误差的理论界（分别见第5.1 节和 5.2 节）。基于这些结果，我们随后在第 [NBH18] 节介绍对其的修改，并在我们的方法中使用该修改，以尽量减小计算所得的界。最后，我们在第5.4 节和第5.5节介绍扩展 ZoomOut 算法，并详述其实现。

5.1. 近似函数映射

如第4.3节所述，采用 [NBH18] 中的方法计算得到的特征函数，按构造相对于面积矩阵 $\mathbf{A}$ 正交归一。因此，可以根据方程（2）计算函数映射。由此得到以下定义：

定义 5.1 给定两个形状 $\mathcal{M}$ 和 $\mathcal{N}$，其近似特征函数 $\left(\Psi_i^{\mathcal{M}}\right)_i$ 按列堆叠为矩阵 $\mathbf{\Psi}^{\mathcal{M}}$（相应地，$\mathcal{N}$），则与逐点映射 $\mathbf{\Pi}$: $\mathcal{N} \to \mathcal{M}$ 相关联的降维函数映射定义如下：

$$\overline{\mathbf{C}} = \left(\overline{\mathbf{\Psi}}^{\mathcal{N}}\right)^{\top} \mathbf{A}^{\mathcal{N}} \mathbf{\Pi} \overline{\mathbf{\Psi}}^{\mathcal{M}} \tag{6}$$

请注意，这一定义的函数映射使用了近似基。不过，它仍然依赖于完整形状（可能非常稠密）之间完整逐点映射的已知信息。为减轻这一限制，我们引入另一种函数映射 $\mathbf{C}$，它仅依赖于采样点集之间的映射，而与原始点数无关：

定义 5.2 采用与定义 5.1相同的设定，特征函数由方程 (5) 得出，$\left(\phi^{\mathcal{M}}{}_i\right)_i$（相应地，$\mathcal{N}$）按列堆叠为矩阵 $\mathbf{\Phi}^{\mathcal{M}}$（相应地，$\mathcal{N}$）。给定逐点映射 $\overline{\mathbf{\Pi}}: \mathcal{S}^{\mathcal{N}} \to \mathcal{S}^{\mathcal{M}}$，我们定义受限函数映射 $\overline{\mathbf{C}}$ 如下：

$$\widehat{\mathbf{C}} = \left(\overline{\mathbf{\Phi}}^{\mathcal{N}}\right)^{\top} \overline{\mathbf{A}}^{\mathcal{N}} \overline{\mathbf{\Pi}}\, \overline{\mathbf{\Phi}}^{\mathcal{M}} \tag{7}$$

回顾一下，正如第4.3 $\mathcal{S}$ 节所述，表示每个形状上的稀疏样本集。因此，为了定义 $\widehat{\mathbf{C}}$，我们只需能够获得两个形状上采样点之间的逐点映射。由于两组基相对于……均为正交归一，受限函数映射 $\widehat{\mathbf{C}}$ 是与约化空间 $\mathrm{Span}\{\phi^{\mathcal{M}}{}_k\}_k$ 和 $\mathrm{Span}\{\phi^{\mathcal{N}}{}_k\}_k$ 相关联的拉回算子

© 2023Eurographics - 欧洲计算机图形学协会和 John Wiley & Sons Ltd.

.....$\overline{\mathbf{A}}$。此外，利用式（6）中每个形状上的分解 $\boldsymbol{\Psi} = \mathbf{U}\boldsymbol{\Phi}$，以及 $\mathbf{A}$的定义，我们注意到，从方程（6）到（7）只需进行近似 $\boldsymbol{\Pi}\mathbf{U}^{\mathcal{M}} \simeq \mathbf{U}^{\mathcal{N}}\boldsymbol{\Pi}$；我们将在命题 5.2中推导其上界。请注意，也可以在方程（7）中用 $\boldsymbol{\Phi}^{\mathcal{M}}$替换 $\boldsymbol{\Psi}^{\mathcal{M}}$，使映射 $\boldsymbol{\Pi}$ 实际传递逐点值而非系数。正如附录 B中的评估结果所示，实际上我们没有观察到采用这一修改带来任何改进。

与方程（7）中的精确函数映射相比，方程（6）中的近似函数映射的首要优点是只使用小尺寸矩阵。注意，函数 $(\phi_i)_i$ 与面积矩阵和刚度矩阵 $\mathbf{A}$ 和 $\mathbf{W}$相关联，它们定义了 L_2 和 W_1内积，因此可以直接使用所有与函数映射相关的算法，无需采用任何额外的近似或加速启发式方法。最终，可以通过识别两个拉回算子 $\widehat{\mathbf{C}}$ 和 $\mathbf{C}$，得到完整形状之间的稠密逐点映射，相关描述见第5.4节。我们将看到，由此得到的对应关系优于使用形状的重新网格化版本和最近邻外推得到的对应关系，因为我们的函数映射具有亚采样精度。

其次，正如下一节所示，我们的方法具有强有力的理论收敛保证，并给出了近似误差的界限。相比之下，先前的方法（例如 ZoomOut 的加速版本 [MRR*19]（第 4.2.3 节）可能无法收敛到真实的函数映射，即使使用了所有可用信息也是如此。具体而言，Fast ZoomOut 确实会在形状上采样 q个点 $\mathcal{M}$ 和 $\mathcal{N}$，并使用$\mathbf{C}$ 进行近似

$$\mathbf{C}_{\text{F-ZO}} = \underset{\mathbf{X}}{\operatorname{argmin}} \|\mathbf{Q}^{\mathcal{N}}\boldsymbol{\Psi}^{\mathcal{N}}\mathbf{X} - \boldsymbol{\Pi}\mathbf{Q}^{\mathcal{M}}\boldsymbol{\Psi}^{\mathcal{M}}\|_F^2 \tag{8}$$

其中 $\mathbf{Q}^{\mathcal{N}} \in \{0,1\}^{q \times n^{\mathcal{N}}}$满足 $\mathbf{Q}^{\mathcal{N}}_{ij} = 1$ 当且仅当 x_j是形状 $\mathcal{N}$上的第 i^{th}个采样点（ $\mathcal{M}$ 同理）。使用所有点意味着 $\mathbf{Q}$矩阵均为单位矩阵。这种近似对所有采样点赋予相同的重要性，不考虑其面积，因此随着采样点数量增加，无法收敛到潜在的 $\mathbf{C}$。这意味着在实践中需要采用近似均匀采样策略，而这在非常稠密的网格上很难实现。

下一节中，我们将给出所定义的函数映射的近似误差界限，并在之后利用这些界限修改 [NBH18] 中的方法，以减小误差，并获得更准确且理论依据更充分的对应方法。

5.2. 近似误差

上述大多数表达式都涉及曲面 $\mathcal{N}$之间给定的逐点映射 $\boldsymbol{\Pi}$ $\mathcal{M}$。下面的引理给出了若干简单的假设，据此可求得其关联函数映射的利普希茨常数；这对于推导我们估计量的近似误差界非常有用：

引理 5.1 设 $\mathcal{M}$ 和 $\mathcal{N}$为紧曲面，且 $T\colon \mathcal{N} \to \mathcal{M}$ 为微分同胚。则存在 $B_T \in \mathrm{R}$，使得：

$$\|f \circ T\|_{\mathcal{N}} \leq B_T \|f\|_{\mathcal{M}} \quad \forall f \in L^2(\mathcal{M}) \tag{9}$$

其证明可见 [HCO18]（命题 3.3）。

我们的总体目标是在类似 ZoomOut 的函数映射估计算法中使用新设计的函数映射 $\mathbf{C}$。因此，当计算得到的特征向量 Ψ_k 能够很好地近似真实特征向量 Ψ_k 时，我们期望近似函数映射能够模仿底层映射 $\mathbf{C}$。以下命题给出了这两个函数映射之间误差的界：

命题 5.1 设 $\boldsymbol{\Psi}^{\mathcal{N}}$（分别为 $\boldsymbol{\Psi}^{\mathcal{M}}$）和 $\boldsymbol{\Psi}^{\mathcal{N}}$（分别为 $\boldsymbol{\Psi}^{\mathcal{M}}$）是曲面 $\mathcal{N}$（分别为 $\mathcal{M}$）上拉普拉斯算子的前 K 个近似特征向量和真实特征向量。设 C 和 C 分别为与映射 T 相关的原始函数映射和约化函数映射（见公式（6））， 其大小为 K。假设 T 是微分同胚，并令 B_T 为引理 5.1 给出的界。若存在 $\varepsilon \in \mathrm{R}^*_+$，使得对任意 $\mathrm{j} \in \{1,\ldots,\mathrm{K}\}$：

$$\|\boldsymbol{\Psi}^{\mathcal{N}}_j - \overline{\boldsymbol{\Psi}}^{\mathcal{N}}_j\|_\infty \leq \varepsilon \text{ and } \|\boldsymbol{\Psi}^{\mathcal{M}}_j - \overline{\boldsymbol{\Psi}}^{\mathcal{M}}_j\|_\infty \leq \varepsilon$$

于是：

$$\frac{1}{K}\left\|\mathbf{C} - \overline{\mathbf{C}}\right\|_2^2 \leq \varepsilon^2\left(1 + B_T^2\right) \tag{10}$$

证明见附录C。该命题保证，对谱的良好估计意味着函数映射近似准确，从而使其在匹配算法中表现良好。

另一个更根本、需要控制的误差，是函数映射$\mathbf{C}$（定义 5.1 ）与 $\widehat{\mathbf{C}}$（定义5.2）之间的估计误差。如上所述，估计依赖于识别$\boldsymbol{\Pi}\boldsymbol{\Psi}^{\mathcal{M}} \simeq \mathbf{U}^{\mathcal{N}}\boldsymbol{\Pi}\,\boldsymbol{\Phi}^{\mathcal{M}}$，其中 $\boldsymbol{\Pi}$ 是两个采样点集之间的映射 $\mathcal{S}^{\mathcal{N}}$和 $\mathcal{S}^{\mathcal{M}}$，我们期望它在这些空间上与 $\boldsymbol{\Pi}$ 相似。该近似将以下两种过程视为等价：1）在 $\mathcal{S}^{\mathcal{M}}$ 上的数值之间进行插值，然后通过映射 $\boldsymbol{\Pi}$进行传递；2）将 $\mathcal{S}^{\mathcal{M}}$ 上的数值$\mathcal{S}^{\mathcal{N}}$ 通过 $\boldsymbol{\Pi}$ 传递至数值，再在 $\mathcal{N}$上进行插值。下面的命题将给出该近似的误差界：

命题 5.2 设$T\colon\ \mathcal{N} \to \mathcal{M}$ 是由 $\boldsymbol{\Pi}$ 表示的形状之间的逐点映射，并设B_T 为引理 5.1 给出的界。假设$\eta^{\mathcal{S}^{\mathcal{N}}}\colon\ \mathcal{S}^{\mathcal{N}} \to \mathcal{S}^{\mathcal{M}}$ 由 $\boldsymbol{\Pi}$ 表示。令 $\alpha = \min_j u_j^{\mathcal{M}}(v_j) \in [0,1]$。进一步假设存在 $\varepsilon > 0$，使得对于任意$k \in \{1,\ldots,K\}$ 以及$x,\ y \in \mathcal{S}^{\mathcal{M}}$:

$$d^{\mathcal{M}}(x,y) \leq \rho^{\mathcal{M}} \Rightarrow |\overline{\Psi}^{\mathcal{M}}_k(x) - \overline{\Psi}^{\mathcal{M}}_k(y)| \leq \varepsilon \tag{11}$$

and

$$d^{\mathcal{M}}(x,y) \leq \rho^{\mathcal{M}} \Rightarrow |\overline{\Phi}^{\mathcal{M}}_k(x) - \overline{\Phi}^{\mathcal{M}}_k(y)| \leq \varepsilon. \tag{12}$$

Then

$$\frac{1}{K}\left\|\boldsymbol{\Pi}\overline{\boldsymbol{\Psi}}^{\mathcal{M}} - \mathbf{U}^{\mathcal{N}}\overline{\boldsymbol{\Pi}}\,\overline{\boldsymbol{\Phi}}^{\mathcal{M}}\right\|^2_{\mathcal{N}} \leq \varepsilon^2(1-\alpha) + \varepsilon^2 B_T^2 \tag{13}$$

证明见附录D。该命题表明，估计误差取决于两个参数：1）特征函数相对于采样距离的变化 ε ρ；2）局部函数的自权重 $u_j(v_j)$，这些局部函数是在基底近似中定义的。注意，由于基函数u_j 满足 $0 \leq u_j \leq 1$ 且构成单位分解，因此可以将它们解释为从采样点处的值插值到整个形状上的值时所用的插值权重。这使得 α 中的依赖关系更直观，因为我们的近似依赖于基系数与函数值之间的局部对应。命题5.2 中所用各量的数值讨论见附录E。

© 2023 Eurographics - 欧洲计算机图形学协会
和 John Wiley & Sons Ltd.

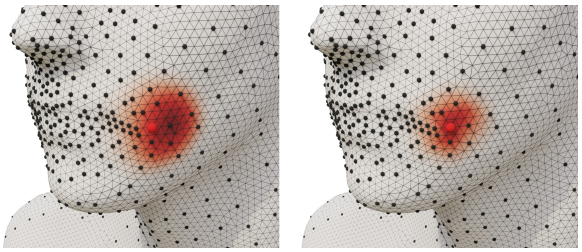

图3： 以 v_j（红色顶点）为中心的局部函数 u_j（红色），分别在未使用（左）和使用（右）我们的自适应半径策略时的可视化结果。其他采样点 v_k 以黑色显示。

因此，接下来我们将尝试修改基底近似 [NBH18]，以最大化 α，同时保持对真实拉普拉斯谱的近似质量；这是应用与函数映射相关算法所必需的。

5.3. 改进的特征基近似

本节中，我们基于上述理论，对 [NBH18], 中的方法提出了一项修改。在本节余下部分，我们聚焦于单个形状，因为基的计算是在每个形状上独立进行的。

如命题5.2所示，较大的自权重有助于稳定我们的函数映射近似。有趣的是，按照 [NBH18], 中的构造，值 $u_j(v_j)$ 仅取决于 v_j 与其他采样点 v_i 之间的测地距离，其中 $i \neq j$:

$$u_j(v_j) = \frac{1}{1+\sum_{i\neq j} \tilde{u}_i(v_j)}. \tag{14}$$

其中 $\tilde{u}_i$ 是未归一化的局部函数。为了提高这些值，我们对 [NBH18]的处理流程进行了如下修改：首先，为每个采样点 $j \in \{1,\dots,p\}$ 定义一个半径 ρ_j，而不是使用单一的全局值 ρ。对于自权重 $u_j(v_j)$ 较小的采样点 v_j，保持其半径 ρ_j不变，因为它不会影响自权重；相反，我们缩小影响较大的邻点的半径 ρ_i，也就是取值 $\tilde{u}_i(v_j)$ 最高的点 v_i的半径。根据方程 (14)，这最终会增大 $u_j(v_j)$ 的值。请注意，这一修改不会改变 $u_i(v_i)$ 的值，并会增大其所有邻点的自权重。这样一来，算法执行过程中所有自权重都不会减小，且至少有一个自权重会增大。这种额外的调整所需的计算成本可以忽略不计，因为它只需在一组固定顶点上重新计算 u_j。尤其是，这不需要额外的局部测地距离计算。更多细节见第 5.5 节，计算这些新函数的算法见算法 1。我们发现，与 [NBH18], 中介绍的方法相比，自适应半径策略生成的局部函数更好，尤其是在非均匀采样的情况下；图3展示了 DFaust数据集 [BRPB17] 中的一个曲面示例。请注意，由于我们关注局部分析，在仅提供采样点处的值时，局部插值函数在不同形状之间保持一致是一项理想属性。使用单一的全局半径时，从图3可以看到，这些函数可能会被归一化过程严重扭曲，而我们的方法能够纠正这一问题。

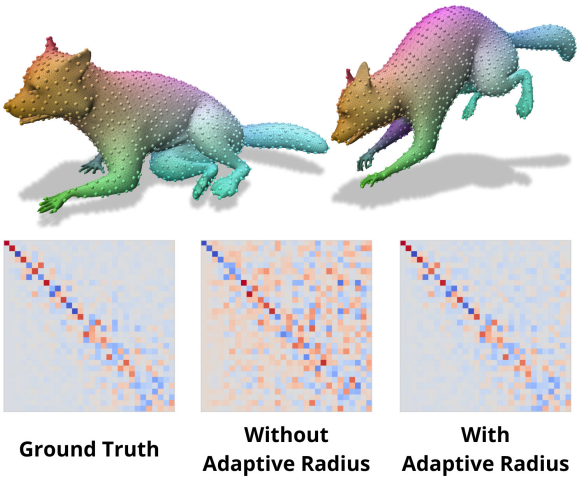

图 4： 自适应半径对函数映射近似的影响。上排通过颜色传递显示了从右侧网格到左侧网格的逐点映射 T。下排显示了$\mathbf{C}$（左）、 $\hat{\mathbf{C}}$ 使用 *[NBH18]* 中的处理流程时的结果（中），以及我们的函数映射 $\hat{\mathbf{C}}$ （右）。

然而，将自权重提高到过于接近 1 也会使结果变差，因为半径范围内若只有一个采样点，则其中的任意顶点 x 都会被赋予该采样点的值。因此，这一过程存在一个不再有益的极限；此时唯一的解决办法是在形状上增加采样点的数量。

算法 1使用自适应半径计算局部函数

要求： 网格 $\mathcal{M}$，采样点集 $(v_k)_k$，初始值 ρ_0，阈值 ε

```
ρ_j ← ρ_0  ∀j
计算局部函数U，半径为 ρ : (3), (4)
必要时添加采样点
当 某个 k 满足 u_k(v_k) < ε 时执行
    j ← argmax_{i≠k} u_i(v_k)
    ρ_j ← ρ_j/2
    更新所有 u使用方程 (3), (4)
当条件不成立时结束循环
向采样点集中添加未见顶点
```

图 4进一步展示了我们的自适应半径算法对函数映射估计的积极作用。给定一个逐点映射 T，我们展示了近似空间 $\mathbf{C}$上的精确函数映射，以及两个近似函数映射 $\hat{}$：其中一个使用共享半径 [NBH18] 计算，另一个则使用我们的自适应半径方案。我们强调，随着约化函数空间发生变化，每个近似 $\hat{\mathbf{C}}$**对应的真实函数映射实际上也各不相同，因此这些数值不能直接比较。不过，我们观察到两个真实映射的稀疏结构几乎相同（见附录** F），因此图 4 中只展示其中一个。值得注意的是，使用自适应半径策略后，矩阵 $\hat{\mathbf{C}}$的稀疏模式与真实映射的稀疏模式非常接近。

© 2023Eurographics - 欧洲计算机图形学协会与 John Wiley & Sons Ltd.

5.4. 可扩展 ZoomOut

鉴于前述讨论和理论分析，我们现在介绍如何在标准 ZoomOut 流程 [MRR*19]中使用近似函数映射 $\mathbf{C}$ 。我们的完整处理流程总结于算法 2中，其中记号 $\mathbf{\Phi}_{1:k}$ 表示我们只使用矩阵k 的前 $\mathbf{\Phi}_{1:k}$列。

算法 2可扩展 ZoomOut

要求： 网格 $\mathcal{M}$ 和 $\mathcal{N}$、阈值 ε、初始映射

1: 使用泊松盘采样对 $\mathcal{S}^{\mathcal{M}}$ 和 $\mathcal{S}^{\mathcal{N}}$ 进行采样
2: 计算$\mathbf{U}^{\mathcal{M}}$ 和 $\mathbf{U}^{\mathcal{N}}$使用算法 1
3: 近似特征向量 $\mathbf{\Phi}_{\mathcal{M}}$ 以及 $\overline{\mathbf{\Phi}}^{\mathcal{M}}$ 求解（5）
4: 设置 $\overline{\mathbf{\Psi}}^{\mathcal{M}} = \mathbf{U}^{\mathcal{M}}\mathbf{\Phi}^{\mathcal{M}}$ 和 $\mathbf{\Psi}^{\mathcal{N}} = \mathbf{U}^{\mathcal{N}}\mathbf{\Phi}^{\mathcal{N}}$
5: 使用初始映射获取 $\mathbf{\Pi}$ 之间的采样点
6: **对于**$k = k_{\text{init}}:k_{\text{final}}$**执行**
7: $\widehat{\mathbf{C}} = \left(\mathbf{\Phi}_1^{\mathcal{N}}{}_{:k}\right)^{\dagger}\mathbf{A}\mathbf{\Pi}\overline{\mathbf{\Phi}}^{\mathcal{M}}_{1:k}$
8: $\mathbf{\Pi} =$ 最近邻搜索$(\mathbf{\Phi}_1^{\mathcal{M}}{}_{:k}, \mathbf{\Phi}_1^{\mathcal{N}}{}_{:k}\widehat{\mathbf{C}})$ 可能使用（16）
9: **</style}结束循环**
10: $\mathbf{\Pi} =$最近邻搜索$(\overline{\mathbf{\Psi}}^{\mathcal{M}}_{1:k}, \overline{\mathbf{\Psi}}^{\mathcal{N}}_{:k}\widehat{\mathbf{C}})$
11: **返回 $\mathbf{\Pi}$**

如前所述，使用$\mathbf{C}$ 以及矩阵$\mathbf{A}$ 和$\mathbf{W}$ ，即可直接应用 ZoomOut 算法，相当于将其应用于仅含p 个顶点的形状的重网格化版本。这样便能得到精化后的函数映射 $\widehat{\mathbf{C}}^*$ ，以及精化后的采样点之间的逐点映射 $\mathbf{\Pi}^*$。最后一个尚待解决的非平凡任务，是将精化后的函数映射转换为原始稠密网格之间的全局逐点映射$\mathbf{\Pi}^*$ 。

采用形状重网格化版本的标准方法会通过最近邻来扩展映射，从而得到局部常值映射。相反，我们识别$\mathbf{C}$ 和$\mathbf{C}$，由此可以通过求解标准最小二乘问题来计算逐点映射 $\mathbf{\Pi}^*$ ：

$$\mathbf{\Pi}^* = \underset{\mathbf{\Pi}}{\operatorname{argmin}} \|\overline{\mathbf{\Psi}}^{\mathcal{N}}\widehat{\mathbf{C}}^* - \mathbf{\Pi}\overline{\mathbf{\Psi}}^{\mathcal{M}}\|^2_{\mathbf{A}\mathcal{N}}. \tag{15}$$

由于$\mathbf{A}$ 是对角矩阵，该问题可简化为针对每个顶点$x \in \mathcal{N}$进行最近邻搜索。这样得到的逐点映射不再是局部常值映射，因此相较于典型方法，质量有了显著提升。

然而，对于包含数百万个顶点的网格，这种最近邻搜索仍然可能非常缓慢。在这种情况下，我们提出利用计算得到的逐点映射 $\mathbf{\Pi}$ 作为引导，按如下方式缩小搜索空间：对于 $x \in \mathcal{N}$，我们首先选出其最近采样点 $N(x) = \{j \mid u_j^{\mathcal{N}}(x) > 0\}$ 的索引，并将可能的像 构成一个集合，其中的点位于 $\mathcal{M}$ 中，且靠近该集合在映射 $\mathbf{\Pi}$下的像，即

$$\mathcal{I}(x) = \{y \mid \exists j \in N(x),\, u_{\overline{T}(j)}(y) > 0\} \tag{16}$$

其中， $\overline{T}$ 是 $\mathbf{\Pi}$的函数表示。由于局部函数u_j 具有紧支撑，实际中通常将其存储为稀疏向量，因此可以通过简单的索引查询，高效地提取给定顶点可能对应的像集。

表1： 处理一对分别含有 $50K$ 和 $200K$ 个顶点的网格，并将 *ZoomOut* 的谱维数从20 提升至 100时，不同方法所需的时间（秒）

方法	预处理	***LBO***	***ZoomOut*** 转换	总计（秒）
ZO	1	132	410	83	626
快速 ZO	10	132	1	44	187
R + ZO	14	2	3	1	21
Ours	10	7	5	44	65

5.5. 实现

我们使用 Python 实现了完整算法，并在以下地址提供代码：
https://github.com/RobinMagnet/Scalable_FM。

按照 [NBH18], 我们使用泊松圆盘采样生成稀疏采样点 $\mathcal{S}$ ，并从所有采样点v_j出发运行固定半径的迪杰斯特拉算法，以构建局部函数u_j。这些值可以存储在稀疏$n \times p$矩阵中，其中 p为采样点数量。请注意，第5.3 节提出的自适应半径算法无需额外的测地距离计算。此外，如第5.4 节所述，寻找某一点可能对应的像集，只需检查稀疏矩阵中的非零索引即可。更多详情见附录G。

6. 结果

本节评估我们的方法，重点关注两个方面。首先，我们验证该方法在处理流程的所有步骤（即预处理和 ZoomOut 算法）中，速度均优于现有方法。其次，我们表明，这一速度提升仅以定量指标的轻微损失为代价。具体而言，我们验证了尽管处理流程依赖稀疏采样点，最终仍能在对应关系中达到明显的亚采样精度。

6.1. 计时

NBH 中介绍的方法 [18] 旨在高效地近似拉普拉斯-贝尔特拉米算子的谱。如上所述，我们在第 5.3 节中引入的额外构件几乎不会增加计算成本，因为主要瓶颈在于局部测地距离的计算，而这些距离无需重新计算。我们的方法在处理 ZoomOut 算法的运行时间时，优势尤为明显。事实上，由于我们的算法不涉及任何 n 维矩阵，其运行时间完全不受原始顶点数量的影响。只有最后的转换步骤（将细化后的函数映射转换为稠密逐点映射）会随顶点数量而变化。表 1 展示了在两个网格之间应用 ZoomOut 算法的计时示例，这两个网格分别有 5 万和 20 万个顶点。我们比较了标准 ZoomOut 算法（ZO）、Fast ZoomOut 算法（Fast ZO）、应用于形状重网格化版本并使用最近邻外推的标准 ZoomOut（R+ZO），以及我们在每个形状上使用 $p = 3000$ 个采样点的完整处理流程。请注意，与泊松采样相比，Fast ZoomOut 中使用的最远点采样在稠密网格上可能会相当慢

© 2023 Eurographics - 欧洲计算机图形学协会
及 John Wiley & Sons Ltd.

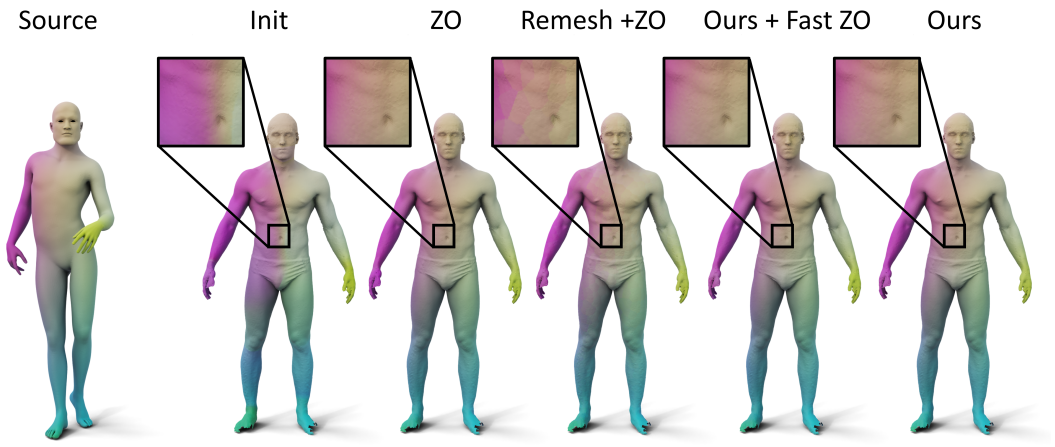

图 5： *SHREC19* 数据集上的定性结果。尽管处理时间差异很大，但我们的方法与 *ZoomOut* 的结果之间没有显著差异。然而，在运行 *ZoomOut* 之前对曲面进行重新网格化，会导致对应关系在局部呈常量。

，这也解释了为什么我们的方法与 Fast ZoomOut 的预处理时间相近。

6.2. 评估

数据集 由于大多数形状匹配方法在顶点数量增加时扩展性较差，因此用于评估的稠密网格和真实对应关系基准数据集寥寥无几。SHREC19 数据集 [MMR*19] 由 430 对连通性各异的人体形状组成，且每对都附带初始对应关系。该数据集中的网格平均有 38 000 个顶点，顶点数量最少和最多的网格分别有 4700 个和 200 000 个顶点。由于现有形状匹配方法的局限性，通常会使用该数据集的重新网格化版本。与此不同，我们展示了完整的稠密数据集上的结果，并表明我们的方法能够在仅用一小部分时间的情况下取得与 ZoomOut 相近的结果 [MRR*19] 。

指标我们使用标准指标评估不同方法 [RPWO19] 来衡量稠密形状对应，即准确率、覆盖率和平滑度。计算得到的稠密映射 $T: \mathcal{N} \to \mathcal{M}$ 表示$T(x)$ 与$T^*(x)$ 之间的平均测地距离，其中$x \in \mathcal{N}$且T^* 表示真实映射。请注意，由于 SHREC19 上的映射仅在包含 6890 个点的小型子集上进行评估，因此该指标只能反映部分信息，而局部常值映射仍可能获得较高的准确率。覆盖率和平滑度指标可提供有关对应关系质量的额外信息，并且对局部常值对应关系十分敏感。覆盖率定义为逐点映射所覆盖面积的比例；平滑度则定义为转移坐标梯度的平方L^2 范数，即狄利克雷能量。

ZoomOut 我们首先使用 3000 个采样点，将我们的方法（Ours）与不使用自适应半径的同一算法（Ours w/o radius）、标准 ZoomOut [MRR*19] 算法应用于稠密网格（ZO）以及具有 3000 个顶点的重新网格化版本（R+ZO）。我们不与其他标准形状匹配基线方法进行比较 [ELC20, ESB19] ，首先是因为我们只希望近似 ZoomOut 的结果，其次是因为这些基线方法无法扩展到顶点数较多的情况。此外，尽管缺乏理论保证，我们还评估了 Fast ZoomOut 的一个新版本，该版本在近似函数空间 8上使用函数映射近似（ $\mathcal{F}$ 于第5.1节介绍（我们的方法 + Fast ZO）。表 2展示了 SHREC19 数据集上的评估指标值，准确率曲线见图 6；图5 展示了在两个稠密网格上计算出的映射示例。可以看到，除 R+ZO 以外，所有方法的指标都相近，尽管处理时间差异显著。相比之下，速度最快的方法 R+ZO 生成的是局部常值映射，如图 5所示，因此其覆盖率和平滑度指标较差。尽管我们的结果与 ZoomOut 和 Fast ZoomOut 相近，但我们强调，我们仅用 ZoomOut 一小部分的处理时间就获得了这些结果，同时还具备理论上界和近似控制参数，而 Fast ZoomOut 并不具备这些。

表 2： 在完整的 *SHREC19* 数据集上对不同方法的评估。蓝色突出显示了表现最好的两种方法。

方法	准确率	覆盖率	平滑度
Init	60.18	26.5 %	9.5
GT	–	33.0 %	10.43
ZO	**26.84**	**61。5 %**	6.2
实数集 +ZO	28.57	18。0 %	15.0
我们的方法（无半径）	71.35	29 %	52.2
我们的方法 +Fast ZO	29.5	59.7 %	6.4
Ours	27.78	56.7 %	**5.6**

© 2023 Eurographics - 欧洲计算机图形学协会和 John Wiley & Sons Ltd.

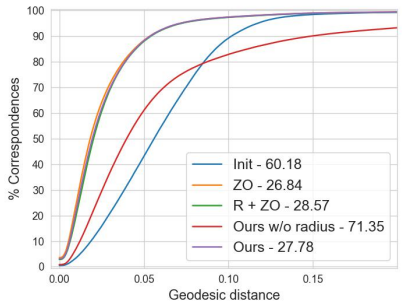

图 6： 表2中所列不同方法的准确率曲线。图例中的数字表示平均测地误差（$\times 10^3$）。

亚采样精度 图 7中，我们展示了将可扩展 ZoomOut 应用于一对真实扫描的肱骨 CT扫描仪数据后进行纹理迁移的结果 [ROA*13]。该图展示了我们的算法如何实现亚采样精度：尽管每个形状上的采样点相当稀疏，迁移后的纹理仍然平滑。图 8 以及附录 H中还展示了在 SHREC19 数据集上进行纹理迁移的类似结果，其中提供了有关这些形状的更多细节。

自适应半径虽然表2中的结果凸显了自适应半径方案的效率，但我们还评估了这一$\hat{}$启发式方法如何改善估计$\Delta = \|\mathbf{C} - \mathbf{C}\|$（第5节所述）。为此，我们只需针对 SHREC19 数据集中的所有初始映射计算$\mathbf{C}$ 和 $\hat{}\mathbf{C}$，其中$K = 20$，并评估估计误差 Δ 的范数，结果见表3。在这项实验中，我们发现我们的方法比基线方法提升了两个数量级。

表3： *SHREC19* 数据集上使用和不使用自适应半径时的估计误差范数 Δ

	不使用半径	我们的方法
$\Delta\,(\times 10)$	1.486	0.018

7. 结论、局限性与未来工作

本文提出了一种新的可扩展方法，用于计算非刚性形状之间的对应关系，这些形状以可能非常稠密的网格表示。我们的方法基于一种高效方法，通过优化由稀疏样本集构建的局部延拓函数的系数来估计拉普拉斯-贝尔特拉米特征基 [NBH18]。我们方法的关键在于仔细分析采样点上的函数空间与原始稠密形状上的函数空间之间的关系。为此，我们扩展了 [NBH18] 中提出的方法，并展示了如何以极少的额外工作获得性质更好的局部函数。我们利用这一构造定义了一种函数映射近似，它仅依赖于存储在采样点中的信息，并为此构造提供理论保证。最后，我们利用这些见解来提出一种可扩展的 ZoomOut 算法变体 [MRR*19]，它能够以标准方法极小一部分的计算成本，在稠密网格之间计算高质量的函数映射和点到点映射。

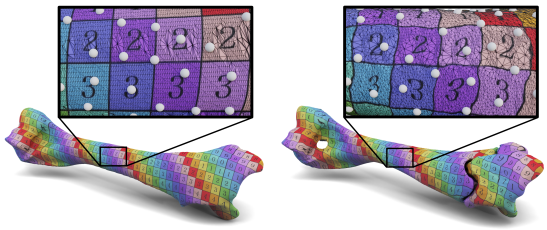

图7： 使用我们可扩展版本的 *ZoomOut* 进行纹理迁移。处理流程中使用的采样点以白点表示。

尽管我们的方法能够取得高质量的结果，但仍存在若干局限性。首先，该方法严重依赖网格结构，无法直接应用于点云等其他表示形式。其次，我们的方法依赖一个关键超参数，即采样点的数量。我们发现，在范围非常广泛的设置下，3000 个采样点都能取得良好效果；不过，研究如何根据谱基的大小确定最优数量将很有意义。此外，我们采用了 [NBH18], 所倡导的泊松采样，实践中效果良好。然而，根据所研究形状的几何性质选择最优采样过程，同样是一个值得探索的方向。最后，原则上我们的方法可以扩展为外存实现，以处理包含数千万至数十亿个顶点的网格；这将是极具实用价值的未来扩展。

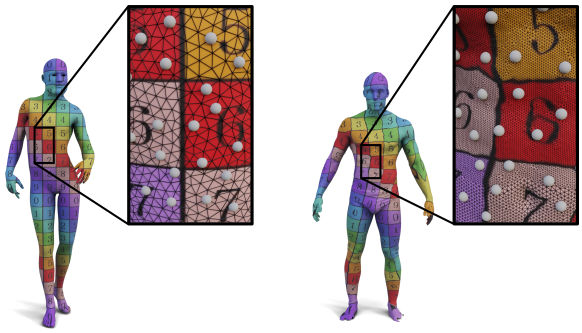

图8： 在一对 *SHREC19* 数据集模型上使用我们可扩展版本的 *ZoomOut* 进行纹理迁移。处理流程中使用的采样点以白点表示。

致谢 作者感谢匿名审稿人提出的宝贵意见和建议。本研究的部分工作得到了 ERC 初始资助项目第758800号（EXPROTEA）和 ANR 人工智能讲席 AIGRETTE 的支持。

参考文献

[AK13] AFLALO Y., KIMMEL R.：谱多维尺度分析。《美国国家科学院院刊》*110*，45（2013年11月），

© 2023 Eurographics - 欧洲计算机图形学协会
和 John Wiley & Sons Ltd.

18052–18057。2[BCBB16] BIASOTTI S., CERRI A., BRONSTEIN A., BRONSTEIN M.：三维形状相似性评估的最新趋势、应用与展望。载于 Computer Graphics Forum（2016），第35卷，Wiley Online Library，第87–119页。2[BDK17] BURGHARD O., DIECKMANN A., KLEIN R.：利用格林函数嵌入形状以实现全局形状匹配。《计算机与图形》68（2017），1–10。2[BRPB17] BOGO F., ROMERO J., PONS-MOLL G., BLACK M. J.：动态 FAUST：运动人体的配准。载于2017年IEEE计算机视觉与模式识别会议（CVPR）论文集（夏威夷州檀香山，2017年7月），IEEE，第5573–5582页。6[CLJL20] CHEN H., LIU H.-T. D., JACOBSON A., LEVIN D. I. W.：用于谱粗化的弦分解。《ACM图形学汇刊》39，6（2020年11月）。2[DLR*20] DYKE R. M., LAI Y.-K., ROSIN P. L., ZAPPALÀ S., DYKES S., GUO D., LI K., MARIN R., MELZI S., YANG J.：SHREC'20：非等距变形下的形状对应。《计算机与图形》92（2020），28–43。1[DYDZ22] DENG B., YAO Y., DYKE R. M., ZHANG J.：非刚性三维配准综述。arXiv预印本 arXiv:2203.07858（2022年）。1，2[EB17] EZUZ D., BEN-CHEN M.：形状间映射的去模糊与去噪。Computer Graphics Forum 36，5（2017年8月），165–174。2[ELC20] EISENBERGER M., LAHNER Z., CREMERS D.：《光滑壳体：利用函数映射进行多尺度形状配准》。载于 IEEE/CVF计算机视觉与模式识别会议论文集（2020），第12265–12274页。2，8[ERGB16] EYNARD D., RODOLA E., GLASHOFF K., BRONSTEIN M. M.：耦合函数映射。载于2016年第四届国际三维视觉会议（3DV）论文集（2016），IEEE，第399–407页。2[ESB19] EZUZ D., SOLOMON J., BEN-CHEN M.：离散曲面间的可逆调和映射。《ACM图形学汇刊》38，2（2019年3月），15:1–15:12。8[HCO18] HUANG R., CHAZAL F., OVSJANIKOV M.：函数映射与形状差异算子的稳定性。Computer Graphics Forum 37，1（2018年2月），145–158。5

[HRWO20] HUANG R., REN J., WONKA P., OVSJANIKOV M.: 一致的 ZoomOut：高效的谱映射同步。 *Computer GraphicsForum 39*，5（2020年8月），265–278。2，4

[JZVK07] JAIN V., ZHANG H., VAN KAICK O.：三角网格的非刚性谱对应。《形状建模国际期刊》13, 01 (2007), 101–124. 2[KBB*12] KOVNATSKY A., BRONSTEIN M. M., BRONSTEIN A. M., GLASHOFF K., KIMMEL R.：耦合准调和基。arXiv:1210.*0026 [cs]* (2012年9月)。arXiv:1210.0026. 2[KGB16] KOVNATSKY A., GLASHOFF K., BRONSTEIN M. M.：MADMM：一种用于流形上非光滑优化的通用算法。载于《欧洲计算机视觉会议》(2016), Springer，第680–696页。2[LJO19] LIU H.-T. D., JACOBSON A., OVSJANIKOV M.：几何算子的谱粗化。《ACM图形学汇刊》38, 4 (2019年7月), 1–13. 2[LLT*20] LESCOAT T., LIU H.-T. D., THIERY J.-M., JACOBSON A., BOUBEKEUR T., OVSJANIKOV M.：谱网格简化。Computer Graphics Forum 39, 2 (2020年5月), 315–324. 2[LRBB17] LITANY O., RODOLÀ E., BRONSTEIN A. M., BRONSTEIN M. M.：全谱部分形状匹配。Computer Graphics Forum 36, 2 (2017年5月), 247–258. 2[LTT*21] LI Y., TAKEHARA H., TAKETOMI T., ZHENG B., NIESSNER M.：4DComplete：超越可观测表面的非刚性运动估计。载于《IEEE/CVF国际计算机视觉会议论文集》(2021)，第12706–12716页。1

[MDK*16] MARON H., DYM N., KEZURER I., KOVALSKY S., LIPMAN Y.：通过高效凸松弛进行点配准。《ACM图形学汇刊》35, 4 (2016年7月), 1–12. 2[MHK*08] MATEUS D., HORAUD R., KNOSSOW D., CUZZOLIN F., BOYER E.：使用拉普拉斯特征函数和无监督点配准进行关节形状匹配。载于《2008年IEEE计算机视觉与模式识别会议》(2008)，IEEE，第1–8页。2[MMR*19] MELZI S., MARIN R., RODOLÀ E., CASTELLANI U., REN J., POULENARD A., WONKA P., OVSJANIKOV M.：SHREC 2019：匹配具有不同连通性的形体。载于Eurographics三维物体检索研讨会 (2019)。1, 8

[MRR*19] MELZI S., REN J., RODOLÀ E., SHARMA A., WONKA P., OVSJANIKOV M.: ZoomOut：用于高效形状对应的谱上采样。《*ACM*图形学汇刊》*38*，6（2019年11月），1–14。1，2，3，4，5，7，8，9

[NBH18] NASIKUN A.，BRANDT C.，HILDEBRANDT K.：快速近似 - 拉普拉斯-贝尔特拉米特征问题。 *Computer Graphics Forum 37*, 5（2018年8月），121–134。2,3,4,5,6,7,9,11

[NH22] 自然数集ASIKUN A., HILDEBRANDT K.：《分层子空间迭代法：用于拉普拉斯-贝尔特拉米特征问题。 *ACMTransactions on Graphics (TOG) 41*, 2（2022年），1–14。 2, 3

[NO17] NOGNENG D.，OVSJANIKOV M.：用于形状匹配的信息描述符保留 - 与交换性。 *Computer Graphics Fo- rum 36*, 2（2017年5月），259–267。 2

[OBS*12] OVSJANIKOV M., BEN-CHEN M., SOLOMON J., BUTSCHER A., GUIBAS L.：《函数映射：一种灵活的形状间映射表示》。《ACM图形学汇刊》31，4（2012年8月），1– 11。1，2，3[OCB*17] OVSJANIKOV M., CORMAN E., BRONSTEIN M., RODOLÀ E., BEN-CHEN M., GUIBAS L., CHAZAL F., BRONSTEIN A.：使用函数映射计算和处理对应关系。载于《ACM SIGGRAPH 2017课程》（美国纽约州纽约市，2017年7月），SIGGRAPH '17，美国计算机协会，第1–62页。2[OMMG10] OVSJANIKOV M., MÉRIGOT Q., MÉMOLI F., GUIBAS L.：利用热核进行单点等距匹配。载于《Computer Graphics Forum》（2010），第29卷，Wiley Online Library，第1555–1564页。2[OMPG13] OVSJANIKOV M., MÉRIGOT Q., P ˇATR ˇAUCEAN V., GUIBAS L.：《通过商空间进行形状匹配》。《Computer Graphics Forum》32，5（2013年8月），1–11。[RCB*17] RODOLÀ E., COSMO L., BRONSTEIN M. M., TORSELLO A., CREMERS D.：部分泛函对应：部分泛函对应。《Computer Graphics Forum》36，1（2017年1月），222–236。2

[RMOW20] REN J.，MELZI S.，OVSJANIKOV M.，WONKA P.：映射 - Tree：在映射空间中恢复多个解。*ACM* 图形学汇刊 *39*，6（2020年11月），264:1–264:17。2，3

[RMWO21] REN J., MELZI S., WONKA P., OVSJANIKOV M.：《用于形状匹配的离散优化》。Computer Graphics Forum 40, 5（2021年8月），81–96。2, 3, 4。[ROA*13] RUSTAMOV R. M., OVSJANIKOV M., AZENCOT O., BEN-CHEN M., CHAZAL F., GUIBAS L.：《基于映射的形状内在差异与变化性探索》。《ACM图形学汇刊》32, 4（2013年7月），1。2, 9[RPWO19] REN J., POULENARD A., WONKA P., OVSJANIKOV M.：《基于函数映射的连续且保持方向的对应关系》。《ACM图形学汇刊》37, 6（2019年1月），1–16。1, 2, 8

[RWHO20] REN J., WONKA P., HARIHARA G., OVSJANIKOV M.：史前人类下颌骨形状变化性的几何分析。人类学 *124*，5（2020年12月），102808。3

[RWP06] REUTER M., WOLTER F.-E., PEINECKE N.: 曲面与实体的"形状 DNA"：拉普拉斯–贝尔特拉米谱。 计算机辅助设计*38*,4(2006),342–366. 2,4

© 2023 Eurographics - 欧洲计算机图形学协会及 John Wiley & Sons Ltd.

[Sah20] 萨希利奥 ¨格鲁 Y .：形状对应领域的最新进展 · *TheVisualComputer36*，8（2020年8月），1705–1721。1，2

[SH10] SHARMA A., HORAUD R.：基于扩散嵌入和互等距一致性的形状匹配。载于*2010 年 IEEE 计算机学会计算机视觉与模式识别会议研讨会*（2010），IEEE，第 29–36 页。2

[SVBC19] SHOHAM M., VAXMAN A., BEN-CHEN M.：细分曲面之间的分层函数映射。载于*Computer Graphics Forum*（2019），第 38 卷，Wiley Online Library，第 55–73 页。2

[TCL*12] TAM G. K., CHENGZ.-Q., LAI Y.-K., LANGBEIN F. C., LIU Y., MARSHALL D., MARTIN R. R., SUN X.-F., ROSIN P. L.：三维点云和网格的配准：从刚性到非刚性的综述。*IEEE Transactions on Visualization and Computer Graphics 19*，7（2012），1199–1217。2

[VBCG10] VAXMAN A .，BEN -CHEN M .，GOTSMAN C .：一种多 - 分辨率离散曲面上的热核方法。在 *ACM SIG- GRAPH2010* 论文集。2010 年，第 1–10 页。2

[VKZHCO11] VAN KAICK O., ZHANG H., HAMARNEH G., COHEN- OR D.: 关于形状对应的综述。载于 *Computer Graphics Forum*（2011），第30卷，Wiley Online Library，第1681–1707页。2

[XLHH21] XU C., LIN H., HU H., HE Y.: 通过细分线性子空间快速计算拉普拉斯-贝尔特拉米特征问题。《计算机与图形》*97*（2021年6月），236–247。4[XLZ21] XIANG R., LAI R., ZHAO H.: 一种用于非刚性形状匹配的双重迭代细化方法。载于 *IEEE/CVF*计算机视觉与模式识别会议论文集（2021），第15930–15939页。2

附录A：函数 χ

函数 $\chi : \mathrm{R}_+ \to [0,1]$，可微，满足 $\chi(0) = 1$ 且 $\chi(\mathrm{x}) = 0$，对 $\mathrm{x} \leq 1$，可以通过两种方式定义。按照 [NBH18],，我们使用多项式插值函数 $\chi : \mathrm{x} \mapsto 1 - 3\mathrm{x}^2 + 2\mathrm{x}^3$。另一种方法是使用标准的 C^∞ 紧支撑函数 $\underline{\chi} : \mathrm{x} \mapsto \exp\left(1 - \frac{1}{1-\mathrm{x}^2}\right)$；我们发现，在结果方面，它不如多项式插值函数。两种函数均展示于图9。

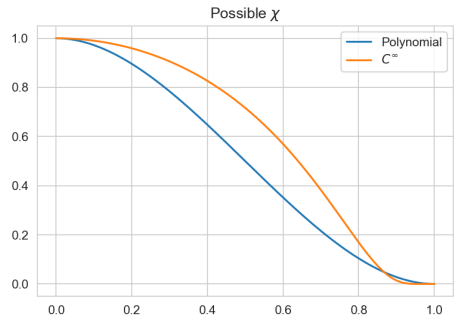

图9： 函数的可能选择 χ

附录 B：系数加权

表4比较了我们的算法（我们的方法）与一种类似算法（我们的方法 +重新加权），其中我们在方程（$\mathbf{\Phi}^{\mathcal{M}}$ 中将 $\mathbf{\Psi}^{\mathcal{M}}$ 替换为7），使映射$\mathbf{\Pi}$ 实际传输的是逐点函数值，而非系数，正如

表4： 在 *SHREC19* 数据集上对重新加权方案的评估。

方法	准确率	覆盖率	平滑度
Init	60.18	26。5 %	9.5
我们的方法 +重加权	28.1	54。6 %	6.3
Ours	27.78	56。7 %	5.6

第5.1节所述。这相当于对局部系数重新加权，使其真正成为函数值。我们注意到，该方法并未改善我们在 SHREC19 数据集上的处理流程。

附录 C：命题的证明5.1

证明 首先注意到，函数映射$\mathbf{C}$ 和$\mathbf{C}$的元素可以写成

$$\mathbf{C}_{i,j} = \langle \psi_j^{\mathcal{N}}, \psi_i^{\mathcal{M}} \circ T \rangle_{\mathcal{N}} \tag{17}$$

$$\overline{\mathbf{C}}_{i,j} = \langle \overline{\psi}_j^{\mathcal{N}}, \overline{\psi}_i^{\mathcal{M}} \circ T \rangle_{\mathcal{N}} \tag{18}$$

此外，给定f_1, g_1, f_2, g_2 上的函数。
那么对于任意的 $x \in \mathcal{N}$

$$\begin{aligned} f_1(x)g_1(x) - f_2(x)g_2(x) =& f_1(x)\left(g_1(x) - g_2(x)\right) \\ &+ g_2(x)\left(f_1(x) - f_2(x)\right) \end{aligned} \tag{19}$$

根据假设，对于$f_1 = \psi_i^{\mathcal{N}}$、$g_1 = \psi_j^{\mathcal{M}} \circ T$、$f_2 = \psi_j^{\mathcal{N}}$和 $g_2 = \psi_i^{\mathcal{M}} \circ T$，有

$$\begin{aligned} \|f_2 - f_1\|_\infty &\leq \varepsilon \\ \|g_2 - g_1\|_\infty &\leq \varepsilon \\ \|g_2\|_{\mathcal{N}} &\leq B_T \end{aligned} \tag{20}$$

因此

$$\begin{aligned} |\langle f_1, g_1 \rangle_{\mathcal{N}} - \langle f_2, g_2 \rangle_{\mathcal{N}}|^2 &= \left| \int_{\mathcal{N}} \left(f_1(x)g_1(x) - f_2(x)g_2(x)\right) d\mu^{\mathcal{N}}(x) \right|^2 \\ &\leq \int_{\mathcal{N}} f_1(x)^2 \left(g_1(x) - g_2(x)\right)^2 d\mu^{\mathcal{N}}(x) \\ &+ \int_{\mathcal{N}} g_2(x)^2 \left(f_1(x) - f_2(x)\right)^2 d\mu^{\mathcal{N}}(x) \\ &\leq \varepsilon \int_{\mathcal{N}} f_1(x)^2 d\mu^{\mathcal{N}}(x) + \varepsilon \int_{\mathcal{N}} g_2(x)^2 d\mu^{\mathcal{N}}(x) \\ &\leq \varepsilon \left(\|f_1\|_{\mathcal{N}}^2 + \|g_2\|_{\mathcal{N}}^2 \right) \\ &\leq \varepsilon^2 \left(1 + B_T^2\right) \end{aligned}$$

对矩阵$\mathbf{C}$ 的所有元素求和，即得所需结果。 □

附录 D：命题5.2的证明

此证明依赖于以下命题，
命题 D.1 给定，$\mathcal{M}$一个曲面，$(v_j)_j$以及 $(u_j)_j$按照第 4.3 节所述构建。

给定f：$\mathcal{M} \to \mathrm{R}$，假设存在 $\varepsilon > 0$，使得对于任意x，$y \in \mathcal{M}$，都有$d(x,y) \leq \rho \implies |f(x) - f(y)| \leq \varepsilon$。

© 2023 Eurographics - 欧洲计算机图形学协会
和 John Wiley & Sons Ltd.

f与 $\tilde{f}=\sum_j f(v_j)u_j$之间的插值误差有界于 ε:

$$|\tilde{f}(x)-f(x)|\leq\varepsilon \quad \forall x\in\mathcal{M} \tag{21}$$

并且对于任意 $j\in\{1,\dots,p\}$

$$|\tilde{f}(v_j)-f(v_j)|\leq\varepsilon\left(1-u_j(v_j)\right) \tag{22}$$

命题的证明*D.1*
令 $f\colon \mathcal{M}\to\mathbb{R}$, $\mathcal{S}=\{v_1,\dots,v_p\}$ 为与半径 ρ 相关联的一个采样点集 $\mathcal{M}$。由于 (u_j) 验证了 $\sum_j u_j=\mathbb{1}$，对于任意 $x\in\mathcal{M}$，我们有

$$f(x)=\sum_{j=1}^{p}u_j(x)f(x) \tag{23}$$

因此，如果 $\tilde{f}=\sum_j f(v_j)u_j$

$$\begin{aligned} f(x)-\tilde{f}(x)&=\sum_{j=1}^{p}u_j(x)(f(x)-f(v_j))\\ &=\sum_{j,\,d(v_j,x)<\rho}u_j(x)(f(x)-f(v_j)) \end{aligned} \tag{24}$$

利用三角不等式以及 $u_j(x)^2\leq u_j(x)$（因为$0\leq u_j(x)\leq 1$），可得:

$$\begin{aligned} |f(x)-\tilde{f}(x)|^2&\leq\sum_{j,\,d(v_j,x)<\rho}u_j(x)^2|f(x)-f(v_j)|^2\\ &\leq\sum_{j,\,d(v_j,x)<\rho}u_j(x)|f(x)-f(v_j)|^2 \end{aligned} \tag{25}$$

由此得到 $|f(x)-\tilde{f}(x)|^2\leq\varepsilon$，这依据了命题的假设以及事实 $\sum_j u_j=\mathbb{1}$。
此外，如果存在 k 使得 $x=v_k$，我们可以去掉索引为k 的项，得到

$$\begin{aligned} |f(x)-\tilde{f}(x)|^2&\leq\sum_{j\neq k,\,d(v_j,x)<\rho}u_j(v_k)^2|f(v_k)-f(v_j)|^2\\ &\leq\varepsilon^2\sum_{j\neq k,\,d(v_j,x)<\rho}u_j(v_k)\\ &\leq\varepsilon^2(1-u_k(v_k)) \end{aligned} \tag{26}$$

□

命题证明*5.2*
我们再次假设所有形状都经过面积归一化。使用 $\tilde{f}$命题 D.1中的记号，我们可以对 $\left\|\boldsymbol{\Pi}\boldsymbol{\Psi}^{\mathcal{M}}-\mathbf{U}^{\mathcal{N}}\boldsymbol{\Pi}\,\boldsymbol{\Phi}^{\mathcal{M}}\right\|_{\mathcal{N}}$应用三角不等式:

$$\begin{aligned} \left\|\boldsymbol{\Pi}\boldsymbol{\Psi}^{\mathcal{M}}-\mathbf{U}^{\mathcal{N}}\boldsymbol{\Pi}\,\boldsymbol{\Phi}^{\mathcal{M}}\right\|_{\mathcal{N}}^2&\leq\left\|\boldsymbol{\Pi}\boldsymbol{\Psi}^{\mathcal{M}}-\widetilde{\boldsymbol{\Pi}\boldsymbol{\Psi}^{\mathcal{M}}}\right\|_{\mathcal{N}}^2\\ &+\left\|\widetilde{\boldsymbol{\Pi}\boldsymbol{\Psi}^{\mathcal{M}}}-\mathbf{U}^{\mathcal{N}}\boldsymbol{\Pi}\,\boldsymbol{\Phi}^{\mathcal{M}}\right\|_{\mathcal{N}}^2 \end{aligned} \tag{27}$$

第一项可以分解为其各列范数之和，其中每一项的形式为 K $\widetilde{\quad}$，并且可以通过应用命题 $\|\circ T-\Psi^{\mathcal{M}}{}_j\circ T\|_{\mathcal{N}}^2$ 中关于插值误差的界来控制，该界与有界畸变相关联 D.1 。

引理，即

$$\left\|\boldsymbol{\Pi}\boldsymbol{\Psi}^{\mathcal{M}}-\widetilde{\boldsymbol{\Pi}\boldsymbol{\Psi}^{\mathcal{M}}}\right\|_{\mathcal{N}}^2\leq KB_T^2\varepsilon^2 \tag{28}$$

着眼于第二项，下面的引理对于对其进行界定非常有用:

引理 D.1 给定 $\beta\in$ 实数集$^{p^{\mathcal{N}}}$, $\|\mathbf{U}^{\mathcal{N}}\beta\|_{\mathcal{N}}^2\leq\|\beta\|_F^2$

我们确实注意到，第二项可以写成$\|\mathbf{U}^{\mathcal{N}}\mathbf{A}-\mathbf{U}^{\mathcal{N}}\mathbf{B}\|_{\mathcal{N}}^2$的形式。利用引理 D.1，现在我们可以着重界定 $\|A-B\|_F^2$，尤其是 $A-B$各列的平方范数。实际上，每一列都可以写成$\left(\psi_j^{\mathcal{M}}\circ T(v_k)-\phi^{\mathcal{M}}{}_j\circ T^{|\mathcal{S}^{\mathcal{N}}}(v_k)\right)_k$，因为我们假设 $T^{|\mathcal{S}^{\mathcal{N}}}$在这些子样本之间有定义。

Given $k\in\{1,\dots,p^{\mathcal{N}}\}$, there exists $i_0\in\{1,\dots,p^{\mathcal{M}}\}$ so that $T(v_i^{\mathcal{N}})=v_{i_0}^{\mathcal{M}}$.
此外，根据近似特征向量 Ψ_j 的定义，对于所有 $x\in\mathcal{M}$，都有 $\psi_j^{\mathcal{M}}(x)=\sum^{p^{\mathcal{M}}}{}_{k=1}\phi^{\mathcal{M}}{}_j(v_k^{\mathcal{M}})u_k^{\mathcal{M}}(x)$。因此，记 $\Delta_j(i)=\psi_j^{\mathcal{M}}(v_{i_0}^{\mathcal{M}})-\phi^{\mathcal{M}}{}_j(v_{i_0}^{\mathcal{M}})$

$$\Delta_j(i)=\sum_{k=1}^{p^{\mathcal{M}}}\bar{\phi}_j^{\mathcal{M}}(v_k^{\mathcal{M}})u_k^{\mathcal{M}}(v_{i_0}^{\mathcal{M}})-\bar{\phi}_j^{\mathcal{M}}(v_{i_0}^{\mathcal{M}}) \tag{29}$$

$$=\sum_{k=1}^{p^{\mathcal{M}}}u_k^{\mathcal{M}}(v_{i_0}^{\mathcal{M}})\left(\bar{\phi}_j^{\mathcal{M}}(v_k^{\mathcal{M}})-\bar{\phi}_j^{\mathcal{M}}(v_{i_0}^{\mathcal{M}})\right) \tag{30}$$

现在可以采用与命题 D.1 证明中完全相同的步骤，从而对该项进行界定

$$\left\|\widetilde{\boldsymbol{\Pi}\boldsymbol{\Psi}^{\mathcal{M}}}-\mathbf{U}^{\mathcal{N}}\boldsymbol{\Pi}\,\boldsymbol{\Phi}^{\mathcal{M}}\right\|_{\mathcal{N}}^2\leq K\varepsilon(1-\alpha) \tag{31}$$

将式（28）和（31）中的各项相加，即可得到命题5.2的上界。

□

附录 E: 理论量的取值

此处给出命题 5.2中各个命名量的取值。再次强调，所提出的界并不紧致，仅用于指导参数选择。

首先请注意，B_T 是一个利普希茨常数；当T为等距变换时，该常数为 1，否则则与T引起的面积畸变有关。

α 的取值范围为 0 到 1，但我们的自适应半径方案确保其最小值为 0.3。实际上，该值通常更高，在 SHREC19 数据集上的平均值约为.43。

最后，ε控制近似特征向量在局部邻域中的变化，并且可以通过减小 ρ 的值将其设得任意小（还可能需要增加采样点集的数量，以确保单位分解）。请注意，特征向量的频率越高，ε 的最大值也越高；该最大值是在所有局部邻域中取最大值。实践中，我们

© 2023 Eurographics - 欧洲计算机图形学协会和 John Wiley & Sons Ltd.

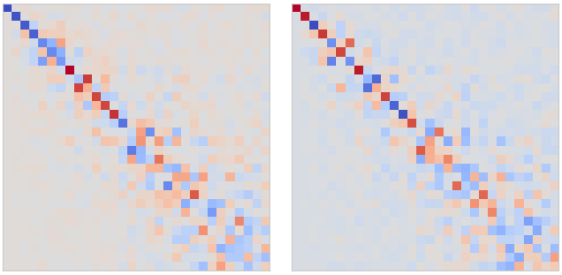

图 10： 使用函数空间 $\mathcal{F}$ 计算的真实函数映射$\mathbf{C}$，未采用（左）和采用（右）自适应半径。请注意，除了符号相反之外，这两个函数映射看起来相似

在使用约 1500 个采样点集时，观察到前 150 个特征向量的最大值介于 0 和 8 之间。相比之下，仅比较边两侧的精确特征向量值时，得到的值介于 0 和 4 之间。

附录 F： 函数映射近似

我们在图 10 中展示了图 4对应的真实函数映射的像

附录 G： 实现细节

我们在此补充说明实现所用的参数和算法。

每个顶点的半径最初都设为相同的初始值 ρ_0，定义为 $\rho_0 = 3\tilde{\rho}_0$ with $\tilde{\rho}_0 = \sqrt{\frac{\text{Area}(\mathcal{M})}{\pi \cdot p_0}}$ 的值通过假设每个采样点 v_j 占据半径为 $\tilde{\rho}_0$的测地圆盘来确定，该圆盘最终将覆盖整个形状——即 $p\pi\rho^2 = \text{Area}(\mathcal{M})$。如果采样点的选择不受限制，我们建议使用泊松盘采样，以快速获得近似均匀分布的采样点。

从采样点出发的局部 Dijkstra 算法可以通过并行化以及将搜索空间缩减为每个采样点周围半径为 ρ_0的欧几里得球来加速，因为我们有 $d^{\mathcal{M}}(x_i, x_j) \leq \|x_i - x_j\|_2$。

如果在此过程中有些点 $x_i \in \mathcal{M}$ 尚未被到达，则应增大初始半径 ρ_0，或者直接将 x_i 添加到采样点集 $\mathcal{S}$ 中，并从x_i出发再运行一次局部 Dijkstra 算法。

现在可以计算 $(\tilde{u}_j)_j$的值，并将其存储在稀疏的 $n \times p$矩阵 $\tilde{\mathbf{U}}$ 中，其中每一列存储一个局部函数。最后，为了检测过小的自权重 $u_j(v_j)$，我们从公式 (14) 注意到，$u_j(v_j) \leq \alpha$等价于 $\sum_i \tilde{u}_i(v_j) \geq \frac{1}{\alpha}$，其中第一项是从 $\tilde{\mathbf{U}}$中提取的$p \times p$ 子矩阵的一行之和。减小采样点的半径 ρ_j只需根据第一次 Dijkstra 运行计算出的U 的j列U 从同一批 距离值重新计算即可。这样就无需额外运行 Dijkstra，从而几乎不增加成本地改进局部函数。

附录 H： 纹理迁移

我们进一步通过展示 SHREC19 数据集中的一对稠密网格上的另一个纹理迁移示例，说明我们的方法在准确率方面的效率，如图 11所示。这里，最左侧的形状包含 50 000 个顶点，最右侧的形状包含 200 000 个顶点，但我们仅使用近 1500 个采样点便获得了这样的对应关系。请注意，在这种情况下，目标形状上的顶点数多于源形状上的顶点数，纹理迁移尤其具有挑战性，因为目标形状的多个顶点会投影到同一个三角形上。这使得纹理迁移对估计映射的质量非常敏感。我们强调，这一处理流程获得的对应关系具有亚采样精度，而所需时间仅为运行精确 ZoomOut 流程所需时间的一小部分。我们还要指出，源形状肘部和肩部处的纹理并不平滑，这解释了目标形状上的失真。这一结果仅用于可视化。

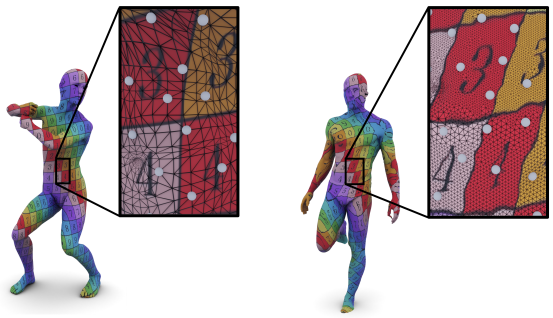

图 11： 在 *SHREC19* 数据集中的一对模型上，使用我们可扩展版本的 *ZoomOut* 进行纹理迁移。处理流程中使用的采样点以白点表示。

© 2023 Eurographics - 欧洲计算机图形学协会
和 John Wiley & Sons Ltd.