

专题 10: 拓扑

教授: 陈宝权

作者: 陈铖、韩启申、刘恺博、单俊杰、杨晨阳、郑炜强

免责声明: 该笔记尚未进行正式出版物的通常审查, 未经教师允许不能向课程外分发。

拓扑学是我们用来研究物体连通性的工具。物体的拓扑只有在极端情况 (撕裂、拼合) 下才会改变。因此, 拓扑不变量是一种更加鲁棒的全局性质。而只有有相同的拓扑性质 (拓扑不变量), 不同的物体间才能有平滑的变形 (不能分裂或合并物体)。

10.1 基本概念

定义 10.1 流形 (*Manifold*)

设 M 是 Hausdorff 空间, 若对任意一点 $x \in M$, 都有 x 在 M 中的一个邻域 U 同胚于 m 维欧几里得空间 R^m 的一个开集, 就称 M 是一个 m 维流形。

其中, Hausdorff 空间是指任意两点都存在不相交邻域的空间。

直观来讲, 流形就是局部具有欧几里得空间性质的空间, 比如对于一维流形, 我们在每一点都有和直线一样的连通性; 而对于二维流形, 每一点附近都像是一个圆盘。

如果一个 d 维流形可以被镶嵌到 n 维欧氏空间中, 那么 $d \leq n$ 。

是否存在不是流形的表面呢? 这样的平面是存在的: 考虑三个平面相交于一条直线, 这条直线上的点的邻域便不是一个圆盘。

定义 10.2 边界 (*Boundary*)

对于拓扑空间的子集 S , 它的边界就是 S 的闭包减去 S 的内部: $\partial S = \bar{S} - S^\circ$ 。

边界的一个等价定义是所有满足以下条件的点 x 的集合: x 的每个邻域都包含至少一个点属于 S , 且至少一个点不属于 S 。

而对于多面体网格, 我们称只和一个表面相交的边为边界边, 其余的边为内部边。一个没有边界边的流形网格称为闭合流形。

定义 10.3 连通性 (*Connectivity*)

拓扑空间 X 称为是连通的, 当且仅当 X 不能表示为两个不相交的非空开集的并集。

直观上来讲, 连通性就是沿表面能从一个点到另一个点 (曲线连通)。我们可以将不连通的物体分成若干连通分支, 如何计算连通分支的数目呢?

对于每个拓扑空间, 都有一族重要的不变量: 贝蒂数 (Betti numbers)。直观地看, b_0 是连通分支的个数, b_1 是沿着闭曲线剪开空间而保持连通的最大的剪裁次数, b_2 是物体中空腔的个数。更高次的 b_k 可借同调群定义。

定义 10.4 亏格 (*Genus*)

连通、可定向曲面的亏格 g 是一个整数, 代表沿闭简单曲线切开但不切断曲面的最大不相交曲线条数。

例如: 球面和圆盘的亏格都为 0, 而甜甜圈的亏格为 1。

直观的来看, 亏格就是物体中洞的个数, 也是物体中把柄 (handle) 的个数。

定义 10.5 可定向性 (*Orientability*)

在曲面上一点作关于这个曲面的法线, 对于任意从该点出发并回到该点的在曲面上的任意平滑游走, 回到该点时法线的方向不变, 那么曲面就是可定向的, 否则就是不可定向的。

莫比乌斯环和克莱因瓶是著名的不可定向曲面的例子。

对于三角网格, 我们可以这样定义可定向性: 通过对一个三角的每条边选取一个方向 (想象在每条边上画一个箭头), 使得当我们沿着三角形的边界绕一圈时箭头头尾相连, 这样我们可定向每个三角形。如果我们可使得共有一条边的两个三角形在这条边上的箭头相反, 则我们说对这个平面做了一个定向。

定义 10.6 欧拉示性数 (*Euler characteristic*)

对于拓扑空间 X , 欧拉示性数 χ 可定义为贝蒂数的交错和, $\chi = b_0 - b_1 + b_2 - b_3 \dots$

对于二维拓扑多面体 (三角网格), 我们有 $\chi = V - E + F$, 其中 V, E, F 分别为顶点数、边数、面数。对于闭合可定向曲面, 我们有 $\chi = 2(1 - g)$, g 为曲面的亏格。

下面考虑欧拉示性数的应用:

对于闭合流形三角网格, 每条边恰好在两个面内, 而每个面恰好含三条边, 故有 $2E = 3F$ 。

进而, $\chi = V - E + F = V - F/2 = 2(1 - g)$ 。

当三角形很多时, 右侧可以忽略, 故有 $F \approx 2V$, 类似地有 $E \approx 3V$ 。

10.2 应用

在图形学中，一些噪音可能会导致物体表面连通性的重大改变，这会给许多图形处理问题带来困难。这就需要一些简化手段来处理。此外，表面重建、形状匹配等应用场景，也都和物体的拓扑密切相关。

10.2.1 表面重建

这一节主要讲了 [JZH07] 的思想方法（详见参考文献）。

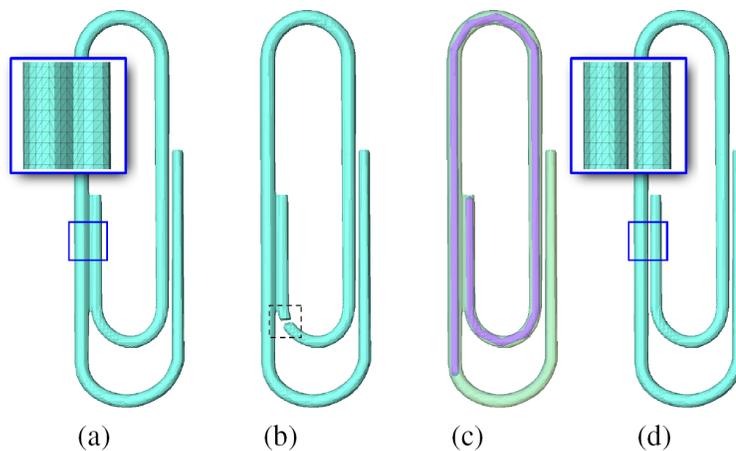


Figure 1: The Clip model (a) incorrectly merges two parts (highlighted) and has genus 1. A typical topology repair method would produce a model with the right genus (b) yet cutting the model at an undesired location. Our approach utilizes a simple target (purple in c) created by sketching and produces a satisfactory result (d).

图 10.1: 主要目标

如图 10.1，我们的目标就是通过提供一个拓扑体模板（称为 target）来修正一个拓扑结构有缺陷的原物体（称为 source）。其中图 (a) 为 source，因为其拓扑结构和经典的回形针不符合（有一部分粘连在一起），而 (c) 中的紫色部分为 target。而最终我们的算法将 (a) 变化为了 (d)。接下来我们介绍如何进行这个过程。

首先有一些方法可以把三维空间问题规约到二维空间上，并且将其格点化。所以接下来的讨论我们围绕一个二维格点网络。

整体思路如图，a 中灰色部分为 source，紫色部分为 target（注意，target 都是 source 的一个子集，且是认为给定的。）首先我们通过一些方法生成 source 的一个包含 target 的骨架，称为 Fat Skeleton。如图 b 中红色部分。要求 FS 与 source 是拓扑同构的。然后我们递归地删除 FS 中 target 以外的所有部

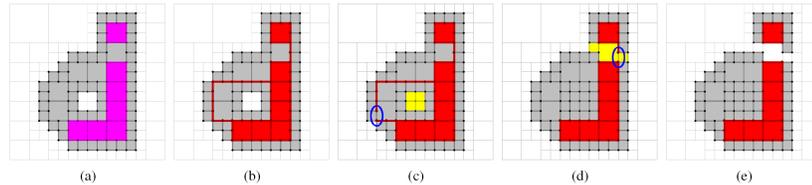


Figure 3: Guided topology editing. (a): The source (gray) and target (purple) represented as cell complexes. (b): An FS (red) of the source that contains the target. (c,d): FS reductions that modify the source to maintain topological consistency with the reduced FSs (modifications shown in yellow). (e): When the FS is reduced to the target, the modified source is topologically consistent with the target.

图 10.2: 整体思路

分, 在这个过程中同时也编辑 source, 保证 source 始终和改变后的 FS 拓扑同构。那么显然, 当 FS 的删除结束时, source 就和 target 同构了, 算法也就结束了。

我们在网格上考虑问题, 一个物体是一系列元素 (点, 线, 面, 体等) 的集合, 并且要求任何一个元素的所有子集也包含在这个物体中。比如一个面属于该物体, 则面上的所有边也属于物体。我们把这样的一个物体称为 cell complexes.

10.2.1.1 FS generation

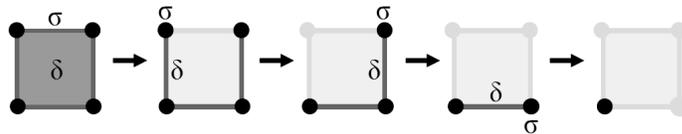


Figure 4: A sequence of simple removals from a cell complex.

图 10.3: simple removal

定义 10.7 A simple removal from a cell complexes V is the deletion of a pair of elements $\{\delta, \sigma\} \subset V$ such that $D(\delta) = D(\sigma) + 1$ and that δ is the only element in V that contains σ .

如图, 我们不断进行 simple removal, 最终就可以得到任意一个网格体的骨架。而在从 source 生成 FS 的过程中, 我们要注意不要删除 target 的任何部分, 那么最后剩下的部分确实就是我们所需要的 FS。(再次注意, FS 是人为给定的)

10.2.1.2 FS Reduction-Subtraction

首先我们定义 $R_{V,S}$ 就是一个从物体 V 生成对应骨架 S 的过程中的所有 simple removal 的序列, 即一系列 pairs δ, σ , 其中 $D(\delta) = D(\sigma) + 1$ and δ contains σ .

定义 10.8 A path associated with an isolated element $e \in S$ is any sequence $\{\sigma_0, \delta_1, \sigma_1, \dots, \delta_k, \sigma_k\} \subseteq V$ where $k \geq 0$ such that $\sigma_0 = e$ and, for all $1 \leq i \leq k, \{\delta_i, \sigma_i\} \in R_{V,S}$ and δ_i contains σ_{i-1} .

定义 10.9 The generating set of an isolated element $e \in S$, denoted as $W_{V,S}(e)$, is the union of all paths associated with e .

简单讲, 对应于骨架 S 上元素 e 的一个 path 就是一个从 V “缩” 到元素 e 的一条路径, 而 generatating set 就是所有这些路径的并集, 也就是 V 中 “缩” 到骨头 e 上的所有 “肉”。

命题 10.10 Let S be an FS of cell complex V and e be an isolated element of S . Then $V' = V \setminus W_{V,S}(e)$ is a cell complex, of which $S' = S \setminus \{e\}$ is an FS.

定理 10.10 即是说, 删除骨架 S 上的一点 e , 同时删除 source 上对应 e 的 generating set, 那么删除后的骨架和删除后的 source 一定拓扑等价。(证明略)

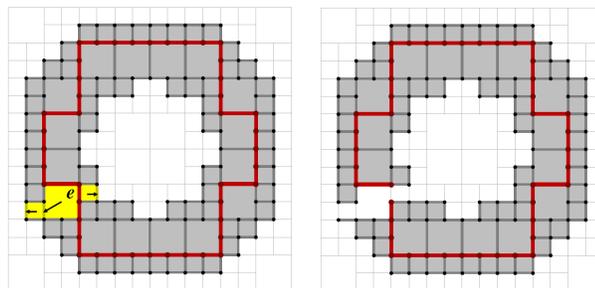


图 10.4: Subtraction

如从图 10.4, 删掉 e 的同时保证骨架和 source 的拓扑等价。

10.2.1.3 FS Reduction-Addition

除了删除 source 以外, 给 source “填” 一部分肉同样可以满足我们要求的拓扑的同构。这里的定义比较复杂我们省去细节, 总的来讲就是, 设 source 为 V , V 的骨架为 S , V 的补集为 \bar{V} (此处不是严格定义), 骨架为 T 。删除 S 中元素 e 的时候我们通过 T 上找一个对应的元素 f , 并把 f 在 \bar{V} 中的 generating set (记为 $\widehat{W_{\bar{V},T}(f)}$) 填补给 V , 即可达到 S/e 与 $V \cup \widehat{W_{\bar{V},T}(f)}$ 拓扑同构。如图 10.5。

我们讨论完了 FS reduction 以后, 整个算法的原理基本完成。接下来就是算法细节如何优化的问题。但我们要强调其中一点, 我们的算法是按照贪心法进行的。即每次进行 FS reduction 的时候, 都要选择代价最小的骨架元素 e 。显然, 这里我们用删除某一元素对应个最小增/删面积来定义低价, 那么一切都都很自然。通过这个贪心法我们就可以得到对原图修改最小的修改路径。

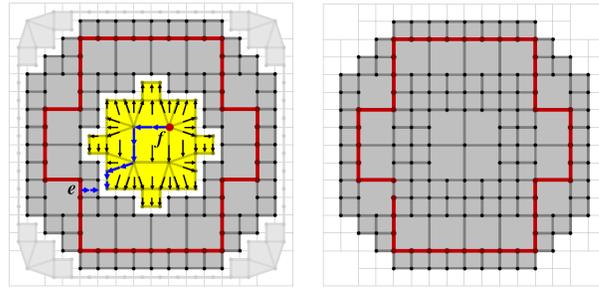


图 10.5: Addition

10.2.1.4 Contributions

我们的表面重建算法相比于传统算法，有以下优势与贡献

1. 利用深度分层技术，用户可以在 2D 屏幕上方便、快速地设置目标。
2. 用户可以控制整个表面重建过程，使得最终结果更加令人满意。例如传统的算法不能正确地重建曲别针，而我们的算法能够非常完美地复原曲别针。在某些复杂模型的重建任务中，用户可以自由地设置目标来保留想要的拓扑性质。
3. 传统算法在表面重建时只支持删除 handles 的重建方法，而我们的算法在 reduction 过程中考虑了增加与删除两种操作，由此对于原模型的改动还能够包括：添加 handles，生成孔洞，填补孔洞以及分割模型等多种操作。

10.2.2 Reeb Graph 与形状匹配

在前面学习拓扑的基本概念的时候，我们提到，拓扑不变量是一种非常鲁棒的全局性质，因此我们可以将拓扑不变量作为形状匹配的标准。比如说，现在有一个物体集合 O ，并给定一个匹配目标 s ，我们想找出 O 中与 s 匹配的物体，这时候就可以筛选出 O 中与 s 亏格数相同的物体作为候选。

然而大部分时候仅仅依靠拓扑不变量是不够的，即便是拓扑结构类似的物体在视觉上仍有很大的不同，比如咖啡杯和甜甜圈。因此我们需要的数据结构不仅仅能刻画出物体的拓扑结构，更需要更进一步，在一定程度上勾勒出物体的“骨架”。本节介绍的 Reeb Graph 就能够较好地达成我们的目的。

定义 10.11 (reeb graph) 给定一个物体 C 上的连续函数 μ ，定义在 μ 下 C 上的等价关系 \sim ：

$$p \sim q \leftrightarrow p, q \in \mu^{-1}(\mu(p)) \text{ 且 } p, q \text{ 在 } \mu^{-1}(\mu(p)) \text{ 的相同连通分量中}$$

Reeb Graph 就是被赋予商拓扑关系的商空间 C / \sim

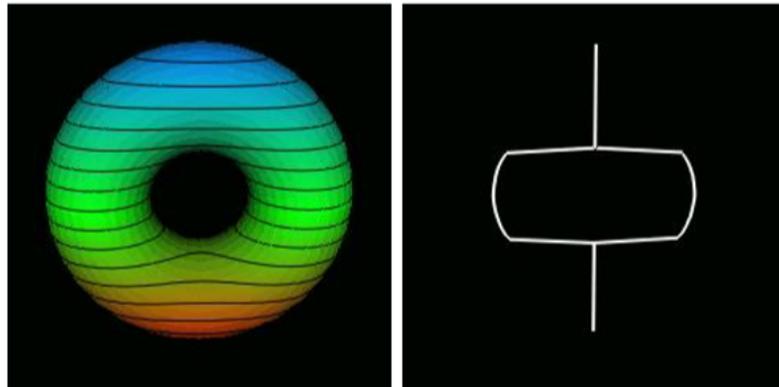


图 10.6: 甜甜圈在高度函数下的 Reeb Graph

我们通过图 10.6 来理解这个定义。首先任意一个 Reeb Graph 都伴随着一个连续函数 μ ，在这里我们用的是高度函数。任取一个值 t ， $\mu^{-1}(t)$ 在原物体上即为一个等值线，在我们的例子中即是甜甜圈上的黑色等高线。而每一个闭合的等高线圈都对应于 Reeb Graph 中的一个点。注意到，在甜甜圈的中部，同一函数值的区域被划分为两个连通分量，这两个连通分量在 Reeb Graph 中分别对应一个点 (Figure 中 Reeb Graph 的中端)。然而仅仅是这样定义的商空间还需要商拓扑关系来确保商空间中元素的连续性，可以理解为在原物体中局部的连续性要在 Reeb Graph 中保留。

注意到 μ 函数的连续性在图形学应用中并不是严格要求的。原因在于我们往往得到的是三角形网格化的物体，这些离散的三角形网格上的任意赋值都可以被连续函数拟合，因此即便采用了不连续的 μ (如随机赋值) 往往也不会影响实际的应用。

10.2.2.1 Multiresolutional Reeb Graph

Reeb Graph 很好地刻画了物体的”骨架”，但仍然难以在实际问题中使用。本小节将给出一个 Reeb Graph 的变种 Multiresolutional Reeb Graph (MRG)。

直接构造 Reeb Graph，难点在于 Reeb Graph 本身的连续性，但凡涉及连续的问题，直接处理是比较棘手的。当然，连续的问题也有比较通用的解决方法，即将其分为 $\Delta\mu(p)$ 极小的一些小片段，每一片段认为有相同的函数值。

图 10.7 即有三种不同分辨率的 MRG。在 (a) 中，我们认为这一整个图片的所有点都有相同的函数值 (取区间顶部或底部均可)，在这一个函数值中只有 1 个连通分量，因此对应的图只有一个结点。而在 (b) 中，区间划分的精细程度加倍，原图被分成了两段，可以看到此时 r_1 部分有两个连通分量，对应与 (b) 中 n_1 与 n_2 。MRG 有三个比较重要的性质：

1. 相邻分辨率级别的结点有父子关系。

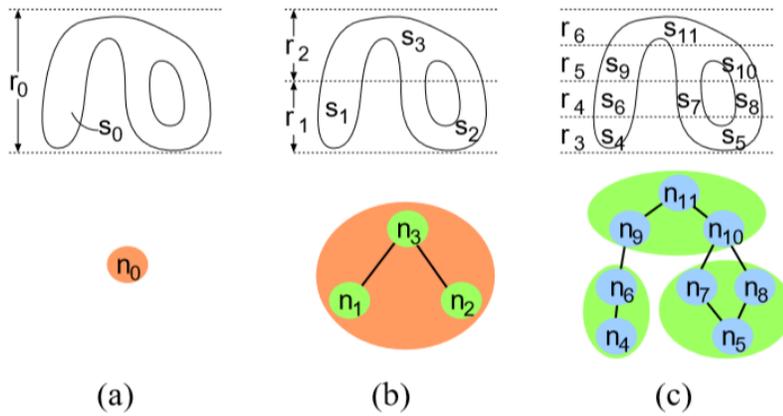


图 10.7: 有三种分辨率的 MRG

例如 (b) 中的 n_2 , 实际上对应于 (c) 中的 n_5, n_6, n_7 , 可以将 n_2 视为 n_5, n_6, n_7 的父结点。

2. 不停地进行更精细的切分, MRG 会“收敛”到原图真正的 Reeb Graph。
3. 更精细的分辨率级别的 Reeb Graph 包含了相对粗糙的分辨率级别的 Reeb Graph 的全部信息。

MRG 通过切分来逼近真正的 Reeb Graph, 并且保留了切分过程中所有分辨率下的 Reeb Graph (根据性质 1 可用树结构存储)。之所以保留低精细程度的 Reeb Graph, 是为了在之后的拓扑匹配过程中先看节点较少的粗糙的 Reeb Graph 能否匹配上, 如此可以增加匹配效率。

10.2.2.2 拓扑匹配的函数选取

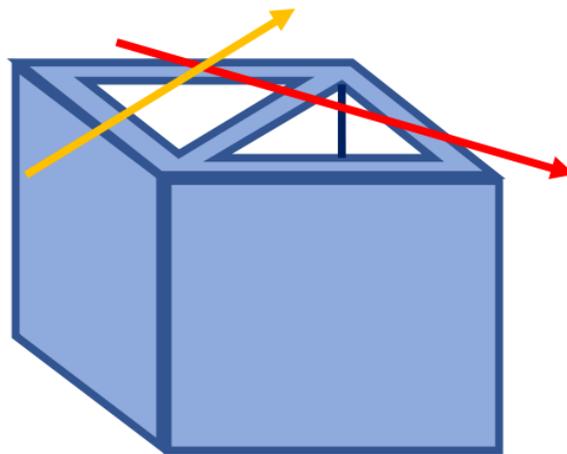


图 10.8: 旋转到不同位置时, 该物体在高度函数下的 Reeb Graph 大不相同

首先我们要讨论的是函数 μ 的选取。以高度函数为例，图 10.8 中两种方向分别作为垂直方向时，这个物体画出的 Reeb Graph 是不相同的。这意味着高度函数是无法消除旋转的影响的。这里我们选取了一个点到表面上所有点的测地距离的积分来作为我们的 μ 函数：

$$\mu(v) = \int_{p \in S} g(v, p) dS$$

其中 g 为两点的测地距离。

若我们希望物体的简单放缩不影响匹配结果，可以进行标准化处理：

$$\mu_n(v) = \frac{\mu(v) - \min_{p \in S} \mu(p)}{\max_{p \in S} \mu(p)}$$

可以看到，这个函数显然是仅仅依赖物体表面而不依赖坐标的，且物体表面简单的翻折也不影响这个函数值。越靠近中心的点函数值越小。

在网格上的测地距离积分可以用选取一些基点以后将每个基点代表的面积乘以通过最短路算法得到的表面最短距离代替。其中可以使用 resample 或添加 shortcut 的方法来增加精确度。

值得注意的是，Reeb Graph 的函数是根据我们的任务而定的。上述函数只是一个较有启发性的在拓扑匹配中比较常用的经典函数，而在其他应用中完全可以选择其他函数。事实上，即便是拓扑匹配，也不一定是这个函数，比如我们的匹配对象如果是一些形象各异的假山，那此时用高度函数更合理（假山是不能转来转去的）。

10.2.2.3 拓扑匹配的算法

此部分略去了相当的细节，原因在于实际上最终实现的 Topology Matching 的算法并没有想象中的优美，并充斥了许多弥补缺陷的细节，这一小节以理解匹配的思想为主。

首先是构建物体的 MRG。我们定义几个接下来要用到的概念：

1. R-node: MRG 上的结点
2. R-edge: MRG 上结点之间连的边。
3. T-set: 每个 R-node 对应的三角形集合。
4. μ_n -range: 每个结点对应的 μ 函数的范围。

根据我们之前对 MRG 的描述，构造的过程就很显而易见了。给定我需要的最精细的分辨率 ϵ ，根据这个分辨率对三角形网格进行划分进 K 个切片中，若有三角形横跨两个切变，则对这个三角形进行切割。

划分之后得到的每个 T-set 都看作是图上的 R-node，再根据两个 T-sets 中的三角形是否有相同的边来添上 R-edge，得到最精细的 Reeb Graph 后再通过合并结点得到粗糙的 Reeb Graph。

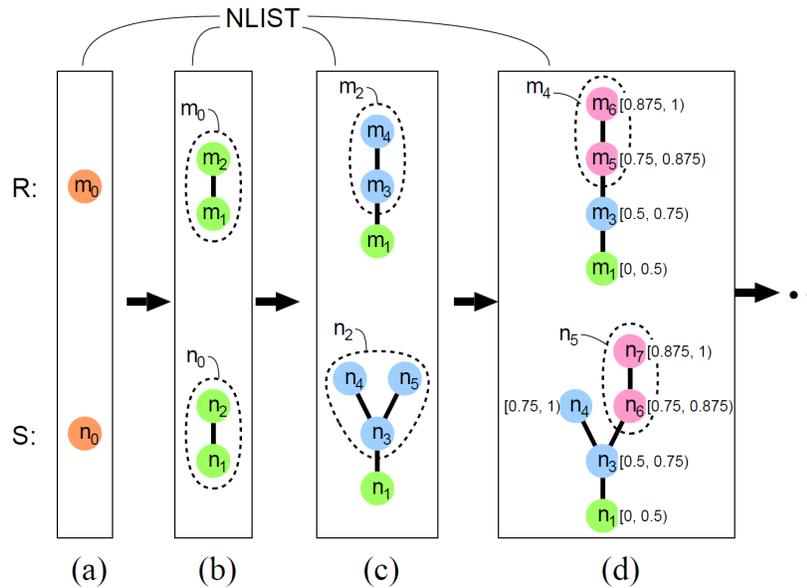


图 10.9: 点对匹配的图示

当我们得到两个物体的 MRG 时, 我们是这样评判两个物体的相似度的: 通过一定的标准, 选取两个 MRG 上匹配的结点对, 并构成集合 V , 每个结点 m 根据其对应的 T-set 有一个特征值 \bar{m} , 然后计算选取出的每个点对的相似度并求和, 用公式表示是:

$$SIM(R, S) = \sum_{(n, m) \text{ in } V} sim(\bar{n}, \bar{m}), V = \{(n_i, m_i) | n_i \in R, m_i \in S\}$$

点对的选取过程是这样的:

1. 将两个物体最粗糙的 R-node 放入各自的候选匹配列表 NLIST 中。
2. 在 NLIST 中, 找到匹配的结点。匹配的要求是: 第一, 它们有相同的 μ_n -range, 并且他们的父结点匹配 (父子关系参见 10.2.2.1 性质 1); 第二, 它们的 MLIST 相同, MLIST 为在同分辨率下, 自该结点向上下扩展得到包含其他 μ_n -range 结点的链状结构。
3. 得到匹配点对 (m, n) 后, 计算 $sim(\bar{m}, \bar{n})$, 并将这两点移去, 同时加入这两点对应的子结点到 NLIST 中。
4. 重复 2、3, 直到得不到更多的匹配结点。

由此我们通过 MRG 得到了两个物体的相似度。如何更进一步的根据相似度进行匹配, 以及 sim 函数具体是如何的等等, 这里略去不做描述, 因为这些东西更似经验与实验之下的产物, 有兴趣的同学可以看论文原文。

10.2.2.4 MRG 的问题

我们知道, MRG 通过切分的方法不断逼近真正的 Reeb Graph。这种方法合理但有弊端。比如图 10.10 中, 左边甜甜圈的内部仅仅比右边甜甜圈的内部小了一点点, 但在某个分辨率下得到了完全不同的 Reeb Graph。这之中蕴含了两个问题: 第一, 在我们的匹配算法中, 我们是从粗糙到精细地进行匹配。当这

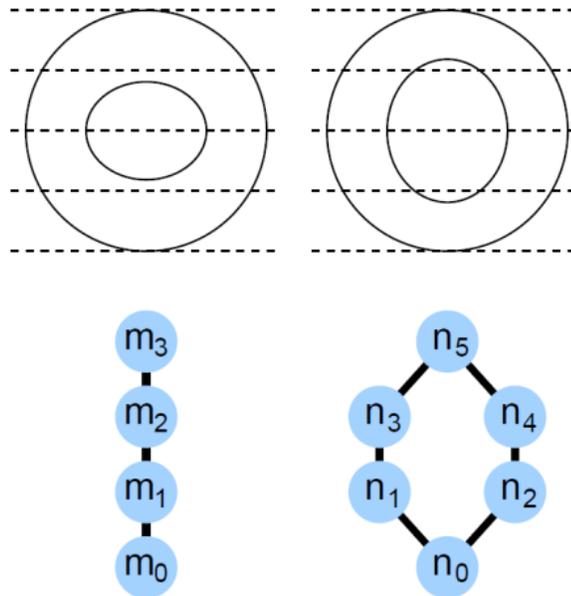


图 10.10: 相似的甜甜圈却有不同 Reeb Graph

样的情况发生在了较粗糙的分辨率级别下, 我们的算法会在很早的阶段就给出了负面的结果, 导致最终匹配失败。

第二, 物体模型中的噪音使得这样的情况在精细的分辨率下是难以避免的。过高的分辨率 (比如, 三角形网格的最短边) 似乎能解决这个问题, 但这样得到的 Reeb Graph 又失去了抽出骨架的本意。

上述两个问题的解决方案留待大家思考。

参考文献

- [FS97] Y. FREUND and R. E. SCHAPIRE, *A Decision-theoretic Generalization of On-line Learning and an Application to Boosting*, Journal of Computer and System Sciences **55** (1997)
- [JZH07] TAO JU, SHI-MIN HU and QIAN-YI ZHOU, *Editing The Topology of 3D Models by Sketching*, ACM Trans. Graph. **26** (2007)

- [BGSF08] S. BIASOTTI, D. GIORGI, M. SPAGNUOLO and B. FALCIDIENO, *Reeb graphs for shape analysis and applications*, Theor. Comput. Sci. **392** (2008)
- [HSKK01] M. HILAGA, Y. SHINAGAWA, T. KOHMURA, and T. L. KUNLI, *Topology matching for fully automatic similarity estimation of 3D shapes*, Proceedings of the 28th annual conference on Computer graphics and interactive techniques. ACM (2001)