arXiv:2305.01220v1 [cs.GR] 2 May 2023

将非结构化数据转换为构造实体几何模型的方法综述

Pierre-Alain Fayolle[1] 和 Markus Friedrich[2]

[1]日本会津大学[2]德国慕尼黑应用科学大学

摘要： 本文旨在综述从非结构化数据（如三维点云或多边形网格）中恢复 CSG表示的现有方法。我们回顾并讨论相关主题，例如输入数据的分割与拟合。我们介绍实体建模和 CAD 领域中用于多面体到 CSG以及边界表示到 CSG转换的技术，并探讨源自程序综合、进化技术（如遗传编程或遗传算法）以及深度学习的方法。最后，我们讨论用于生成表示实体（而不仅仅是CSG模型）的计算机程序以及高层表示的技术，例如基于草图和拉伸或基于特征的操作。

1 引言

用于采集三维数据集的设备日益普及：激光扫描仪、激光雷达、RGB-D相机，甚至借助摄影测量技术的普通RGB相机 [HZ03], ，使得大量三维数据集得以获取。可以使用曲面重建算法，从给定的输入三维点云生成多边形网格。这通常涉及通过样条或多项式集合构建隐式曲面，如 [CBC+01、OBA+03] 中所示；也可以像 [KBH06] 中那样，将其表示为结构化或非结构化网格上的采样点。近年来，研究重点转向采用深度学习技术，将隐式曲面定义为多层感知机（MLP）的零水平集。例如，参见 [SMB+20] 所述的方法。这些方法的主要问题在于，它们生成的输出（多边形网格、多项式/样条的加权和或MLP）并非可编辑模型。理想情况下，用户应能控制恢复出的模型，并能够对其进行编辑、存储和制造（例如，使用3D打印机）。图1展示了一个通过摄影测量获得的桌子三维点云、该物体的CSG模型，以及经过编辑的CSG模型。当然，基于CSG的恢复方法并不一定适用于所有类型的三维物体，而自由形状的某些实体则更适合采用前述方法处理。

1.1 研究目标

本综述旨在全面介绍从非结构化数据源恢复构造实体几何模型的现有方法。我们将三维点云以及多边形（三角形或四边形）网格或网格集合视为典型的输入数据；当然，也可以考虑一组图像，并通过摄影测量将其转换为三维点云。由于这些问题彼此相关，我们还将介绍把多面体（仅涉及半平面）转换为构造实体几何模型的现有算法，以及将边界表示（可能包含曲面）转换为构造实体几何模型的现有技术。此外，我们还将探讨生成表示实体的计算机程序（程序化形状/实体程序）以及高层表示（例如草图和拉伸）的技术。

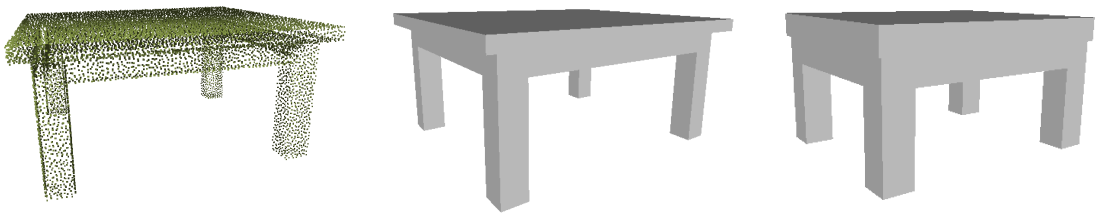


图 1：左：对应一张桌子的三维点云。中：桌子的 CSG 模型。右：编辑后的模型，其中修改了桌面的宽度、深度和高度。

1.2 组织结构

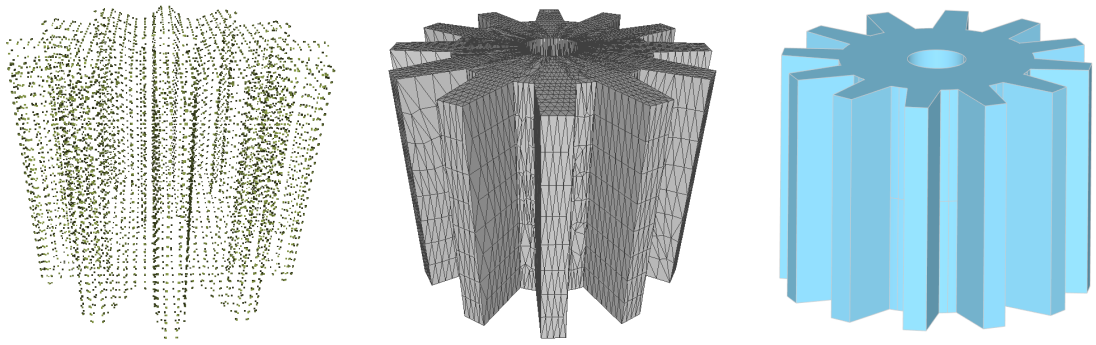
本文结构如下：首先，我们在第2节介绍本文其他部分所用工具和技术的一些细节，包括输入数据和所用基元的信息（第2.1节），以及构造实体几何表示（第2.3节）。第3节讨论将多面体和B-rep模型转换为构造实体几何表达式的方法。第4节介绍从非结构化数据中提取（或恢复）构造实体几何模型的技术。我们将介绍形状分解和基元拟合技术（第4.1节），以及恢复一般构造实体几何表达式的方法（第4.2节）。最后，我们将讨论生成高层表示和形状程序的技术，并在第5节作结。

2 预备知识

本节介绍本文其余部分中使用的术语和技术的背景知识。我们将描述所处理的数据类型，以及对 CSG 表示的介绍。

2.1 输入数据

本文研究非结构化三维数据。这包括点云和多边形网格或多边形网格集合，其中最常见的情况由三角形和四边形组成。



(a) 一个三维点云。　　(b) 三角形网格。　　(c) 边界表示模型。

图 2：输入和数据类型示例。

2.1.1 点云

三维点云是点的集合 $\{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_n\}$，其中 $\mathbf{x}_i \in \mathbb{R}^3$ 是点的坐标，且 $i = 1, \ldots, n$。通常，这些点还带有其他属性，例如法向量 $\mathbf{n}_i \in \mathbb{R}^3$或颜色属性 $\mathbf{c}_i \in [0,1]^3$。法向量对于几何问题很有用，因为它们能够确定物体的朝向。由于本文研究的是几何问题，我们将忽略颜色属性。如果传感器未捕获法向量，则可以在之后对其进行近似。用于估计法向量及其朝向的经典方法由 Hoppe 等人在 [HDD+92] 中提出。近年来，人们研究了基于深度学习的方法，例如 [GKOM18] 中的 PCPNET。图 2(a) 展示了一个机械零件的三维点云示例。

2.1.2 多边形网格和边界表示

多边形网格在计算机图形学中无处不在。它们由一组多边形构成，这些多边形通过其端点坐标指定。最常用的是三角形和/或四边形的集合。通常，三角形网格定义为顶点坐标列表 $V = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ 以及三角形列表 $T = \{\ldots, (i, j, k) \ \ldots\}$，其中 i, j, k 是顶点坐标列表 V中的索引。图 2(b) 展示了一个三角形网格示例。

在工程领域，边界表示（B-rep）是实体的首选表示形式。B-rep 模型由一组拓扑实体组成：顶点、边、面、它们之间的连接关系，以及它们的几何实现（点坐标、曲线定义、曲面片定义）。图 2(c) 展示了一个 B-rep 模型示例。与基于多边形网格的表示相比，B-rep 是一种信息丰富得多的表示形式。在本研究中，我们关注的是如何从非结构化数据中恢复模型，并假设输入为三维点云或由三角形组成的非结构化集合（即所谓的多边形汤）。

2.1.3 数据集集合

多年来，人们不断收集、整理并组织结构化和非结构化点云、三角形网格及多边形网格集合，用于算法基准测试或训练数据驱动算法等目的。Thingi10K [ZJ16] 是一个三角形网格集合，旨在为网格生成算法提供基准测试。它由 Thingiverse [Thi]上的一系列模型构成。ShapeNet [CFG+15, WSK+15] 面向机器学习算法和计算机视觉应用，且仅包含少数几类（或类别）的对象。ABC 数据集在 [KMJ+19] 中首次提出，是一个大型 CAD 模型集合。这些模型提供多种格式，例如多边形网格和边界表示。该数据集旨在为几何深度学习方法和应用训练数据驱动算法。其对象取自 OnShape 的公开模型 [PTC]。同样，Fusion 360 数据集 [WPL+21] 是一个 CAD 模型／程序集合，对应于由草图／拉伸构成的序列。其对象取自使用 Fusion 360 [Aut]设计的模型。该数据集旨在训练数据驱动算法，使其能够将 CAD 模型学习为简短程序，即由草图和拉伸构成的程序。DeepCAD [WXZ21]所用的数据集基于 ABC 数据集，但侧重于采用草图曲线和拉伸表示，与[WPL+21]中介绍的数据集类似。最后，Fit4CAD [RRQ+21]提出了一项基准测试，用于评估和比较在对应于 CAD 对象的点云中拟合简单几何基元的方法。大型数据集集合有两个主要目标：1）提供通用数据集，以便对不同算法进行基准测试和比较；2）提供充足的数据来训练机器学习算法。如今，第二个目标似乎已成为收集三维数据的主要原因，而近期公开的数据集，例如 [KMJ+19、WPL+21、WXZ21], ，已推动多种基于深度学习的方法应运而生。

2.2 基元

几何基元可作为表示 CAD 几何体的基本构件。常用的基元包括球体、圆柱体、圆锥体、环面、平面，以及其他所谓的二次曲面，其曲面可以是】【。

可描述为二次多项式的零水平集。根据所选的表示模型和所使用的系统，可以使用更复杂的基元。对于从非结构化数据集恢复模型的任务，所支持基元的复杂度受所采用的分割和拟合方法限制，参见第4.1节。一般来说，只要能够在输入数据集中识别并拟合任何有符号距离函数（Signed Distance Function，SDF，即返回$\mathbf{x}$到给定曲面 ∂S的有符号距离的函数 f）作为基元，就没有任何因素阻止我们这样做。

半空间 一词用于表示集合 $p = \{\mathbf{x} \in \mathbb{R}^3 : f(\mathbf{x}) \leq 0\}$，其中函数为 $f: \mathbb{R}^3 \to \mathbb{R}$。半空间补集 $\bar{p}$ 由 $\bar{p} = \{\mathbf{x} \in \mathbb{R}^3 : f(\mathbf{x}) > 0\}$给出。当 f 为多项式时，使用术语**多项式半空间**。当 f 对应于$\mathbf{x} \in \mathbb{R}^3$ 到实体 S 的曲面 ∂S的距离时，这种情况被称为有符号距离函数（SDF）。通常，对于 f的符号，以及它对应于实体 S 的内部还是外部，并不存在统一约定。

2.3 CSG表示

一种广为人知且直观的实体表示方法，是将布尔集合运算与几何基元结合成树状结构。这种表示称为CSG（构造实体几何）表示，相应的树结构称为CSG树。

CSG建模软件包中通常使用的集合运算包括并集、交集和差集。为了确保对实体执行操作后始终得到实体，会使用所谓的正则化集合运算 [Req80]。这些所谓的构造实体几何（CSG）树的表达能力高度取决于所支持的基元集合。

给定一个点集为 S、基元集为 P 且二元运算集合为 R的实体，当且仅当 $|\Phi(P)| = S$时，该实体可由CSG树表达式 Φ 完全描述，其中 $|\cdot|$ 描述了由CSG表达式 Φ 和基元 P得到的点集。我们在此使用的形式化方法源自 [SV91a]。

2.3.1 CSG树计数与复杂度

令 $|P|$ 表示基元的数量，即基元集合 P的基数。对于给定数量 n 的内部节点，二叉树的数量为

$$C_n = \frac{1}{n+1}\binom{2n}{n},$$

这就是所谓的卡特兰数。对于一棵CSG树，我们将其视为一棵二叉树，其中内部节点标记为并集或交集操作，每个叶节点标记为一个基元或其补集。给定内部节点数量 n 时，CSG树的数量为

$$\frac{1}{n+1}\binom{2n}{n} \cdot 2^n \cdot (2|P|)^{n+1},$$

通过计算二叉树的数量以及内部节点和叶节点的可能情况得出。内部节点数量介于 $n_{\min}$ 和 $n_{\max}$ 之间的可能树的数量为（另见 [FIFLP22]）

$$\sum_{n=n_{\min}}^{n_{\max}} (2|P|)^{n+1} \cdot 2^n \cdot \frac{1}{n+1}\binom{2n}{n}. \tag{1}$$

每个基元（或其补集）都应至少在表达式（树）中出现一次，因此我们可以假设 $n_{\min} = |P| - 1$。然而，通常情况下 $n_{\max}$ 未知，必须通过经验进行估算。公式 1 对应于 CSG 生成问题的搜索空间大小。

存在若干不同的 CSG 表达式子集和超集，它们采用二元运算，导致复杂度（搜索空间的大小）各不相同。

2.3.2 析取范式表达式（DE，子集）

Res将表达式集合限制为析取范式（DNF）中的表达式，我们得到

$$\Phi(P) = \bigcup_{k=1}^{2^{|P|}-1} \epsilon_k \left(g_1 \cap^* g_2 \cdots \cap^* g_{|P|} \right), \qquad g_i \in \{p_i, \setminus^* p_i\}, \tag{2}$$

其中，如果 k^{th}基本乘积（$g_1 \cap^* g_2 \cdots \cap^* g_{|P|}$）的点集位于 S内部，则 ϵ_k为 1；否则为 0。这大幅缩小了搜索空间，使其降至 $O(2^{|P|})$。然而，不再能保证表达式大小达到最优。

请注意，布尔逻辑中的一个已知结论指出，布尔函数总能写成完全析取范式。因此，理论上我们总能搜索形式为 (2) 的 CSG表达式，并以此得到复杂度的上界。

2.3.3 并集表达式（UE，子集）

假设所有基元都属于最终实体，则得到的 CSG表达式就是所有基元的并集：

$$\Phi(P) = \bigcup_{k=1}^{|P|} p_i \tag{3}$$

由于并集运算具有对称性，树结构并不重要。大多数形状分解方法都采用这一假设（见第4.1.3节）。

2.3.4 可分解实体的树表达式（DTE，子集）

如果一个实体的所有基元都完全位于该实体内部或完全位于其外部（在 [SV91a] 中称为主导基本体），则称该实体是可分解的。在递归执行的分解过程中，可以从实体中移除（提取）这些基元：

$$S^i = ((...(S^{i+1} \oplus |d_1^i|) \oplus ...) \oplus |d_{n-1}^i|) \oplus |d_n^i|, \tag{4}$$

其中，S^i 是剩余实体的点集，$D^i = \{d_1^i, ..., d_n^i\}$ 是递归级别 i 对应的主导基元序列，$|d|$ 表示由基元 d 诱导的点集。$\oplus$ 或者是集合差运算（如果后续主导基元完全位于 S^i 外部），或者是集合并运算。在完全可分解实体的情况下，所有可能的 CSG表达式的搜索空间缩小为$O(|P|^2)$。此外，可以保证所得表达式具有最优大小。然而，并非所有实体都能以这种方式分解。

2.3.5 m元树表达式（MTE，超集）

如果要支持 m 元操作，例如一元补运算（也称为否定），树结构就必须允许子节点数量不同，这会使搜索空间大小的估算难上许多。

如果将并集、交集、差集（二元）和补集（一元）视为操作，那么得到的树中，每个节点可以有 0、1 或 2 个子节点。这种树称为一元二叉树。

具有 n 条边（或等价地具有 $n+1$ 个节点）的一元二叉树数量由莫茨金数给出

$$M_n = \frac{1}{n} \sum_{k=0}^{n-1} \binom{n}{k} \binom{k}{n-1-k}.$$

然而，如果我们希望计算具有 n 个内部节点的一元二叉树的数量，则该数量由第 n 个大施罗德数 S_n 给出。与莫茨金数 M_n 不同，大施罗德数没有闭式表达式，而是由以下递推关系给出。

$$(n+1)S_n = 3(2n-1)S_{n-1} - (n-2)S_{n-2}.$$

若将一元运算符的数量记为 u，二元运算符的数量记为 b，叶节点的数量记为 L（$L = |P|$ 或 $L = 2|P|$，其中 $|P|$ 为基元的数量），则 n 个内部节点对应的表达式数量由下列关系给出

$$(n+1)T_n = (u+2bL)(2n-1)T_{n-1} - u(n-2)T_{n-2}.$$

虽然不存在 S_n 和 T_n的闭式表达式，但可以计算渐近估计[FS09]。与二元情形类似，搜索空间通过考虑所有可能的内部节点数量获得：$\sum_n T_n$。

2.3.6 图表达式（GE，超集）

如果将基于树的表达式空间扩展为基于图的结构，就可以使用循环等常见的程序设计结构。这会将可能表达式的搜索空间扩展到以更高级的 CAD 专用程序设计语言编写的程序所对应的空间，此时程序综合方法便派上用场。

这里我们再次可以利用渐近分析，得到具有 n 个内部节点的（无标号）图的数量估计。

$$G_n \ \frac{1}{n!}2^{\binom{n}{2}},$$

其中 $G'_n = 2^{\binom{n}{2}}$ 是带标记图的数量，并且渐近地有 $G_n \sim 1/n!G'_n$.

2.4 构造实体几何和有符号距离函数

给定一个 CSG表达式，我们可以将基元替换为其有符号距离函数表示，并将布尔运算表示为最小值/最大值函数，从而构造出一个有符号距离函数（严格来说，这只是一个近似）：

$$|\Phi| \cap^* |\Psi| := \max(f_\Phi, f_\Psi) \tag{5}$$
$$|\Phi| \cup^* |\Psi| := \min(f_\Phi, f_\Psi) \tag{6}$$
$$\setminus^* |\Phi| := -f_\Phi \tag{7}$$
$$|\Phi| -^* |\Psi| := \max(f_\Phi, -f_\Psi), \tag{8}$$

请注意，如果选择基元的不同方向，则 min 和 max 的使用方式也会互换。上述表达式假设实体 $|\Phi|$ 对应于点集 $\{\mathbf{x} : \ f_\Phi(\mathbf{x}) \le 0\}$。最小值/最大值函数的另一种表述是所谓的 r函数 [Sha91]:

$$|\Phi| \cap^* |\Psi| := \frac{1}{2}(f_\Phi + f_\Psi + \sqrt{(F_\Phi - F_\Psi)^2 + \alpha}) \tag{9}$$
$$|\Phi| \cup^* |\Psi| := \frac{1}{2}(f_\Phi + f_\Psi - \sqrt{(F_\Phi - f_\Psi)^2 + \alpha}) \tag{10}$$
$$\setminus^* |\Phi| := -f_\Phi \tag{11}$$
$$|\Phi| -^* |\Psi| := \frac{1}{2}(f_\Phi - f_\Psi - \sqrt{(f_\Phi - f_\Psi)^2 + \alpha}), \tag{12}$$

使用 $\alpha \in \mathbb{R}$。设置 $\alpha = 0$ 即可得到最小值/最大值函数。

3 CSG 转换 计算几何与 CAD 中的转换方法

3.1 多面体到 CSG

对于一个具有 p 个面的多面体，Paterson 和 Yao 在 [PY90] 中提出了一种算法，其时间复杂度为 $O(p^3)$，可生成大小为 $O(p^2)$的 CSG表达式，但该算法仅适用于一类受限的多面体。在二维中，多边形可转换为含有 $O(p)$项的 CSG表达式，这称为 Peterson 风格公式 [Pet86]。Dobkin 等人在 [DGHS88]中提出了一种 $O(p\log p)$ 算法，用于计算给定二维多边形的 Peterson 风格 CSG 公式；他们还证明，在三维中，并非所有多面体都有 Peterson 风格公式。Dey 证明了大小的下界为 $O(p^2)$ [Dey91]。CSG表达式的构造基于凸分解算法。对于每个检测到的、会导致多面体非凸性的特征，都会添加一个额外的平面以消除非凸性。Dey 将凹口 称为导致非凸性的特征，并将凹口平面 称为这些额外的平面。最终的 CSG表达式由凸分解算法得到的凸单元的并集构成。因此，它属于 DE 形式 (2)。最近，Rossignac 提出使用等价布尔运算，并将相应的表示称为 MAS（扇区匹配聚合） [Ros22]。

3.2 边界表示到 CSG

将多面体转换为 CSG 的研究促成了更为困难的问题：边界表示到构造实体几何的转换。Shapiro 和 Vossler 开创了这一领域的研究，最初于二维情形开展，见 [SV91b, SV91a, Sha01], ，随后扩展到三维情形，见 [SV91a, SV93]。Buchele 实现了从边界表示到 BSP，以及从 BSP 到 CSG 的转换，见 [Buc99], ；此后，该方法在 [BC04] 中得到扩展和改进。Shapiro 和 Vossler 在其研究中提出的一些思想和概念具有基础性，并被用作其他点云到 CSG 恢复研究的基础。

3.2.1 自然半空间、分离半空间与可描述性

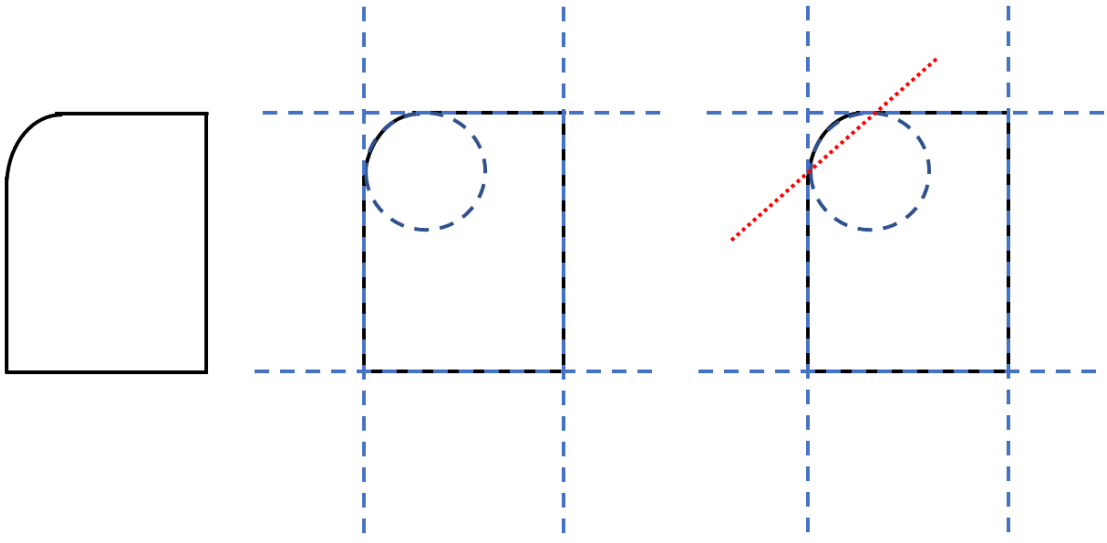


图 3：自然半空间与分离半空间示意图。左：一个二维实体。中：自然半空间以蓝色虚线表示。右：分离半空间以红色点线表示。

自然半空间 由实体的自然曲面诱导，即它是描述实体边界的最小曲面集合。分离半空间 是实体的 CSG表示中必需的半空间，但不是该实体的自然半空间。图 3 说明了以下概念的

自然半空间和分离半空间。一组 n 个半空间将 $\mathbb{R}^3$ 划分为 2^n 个子集。我们将这些集合中的每一个称为基本乘积（或规范单元、规范交项或最小项）。图 4 展示了基本乘积概念的示意图。Shapiro 和 Vossler 还引入了可描述性的概念。给定一组半空间，当且仅当对于每个基本乘积，其中所有点相对于该实体都具有相同的点成员资格分类时，该实体才可由这组半空间描述。

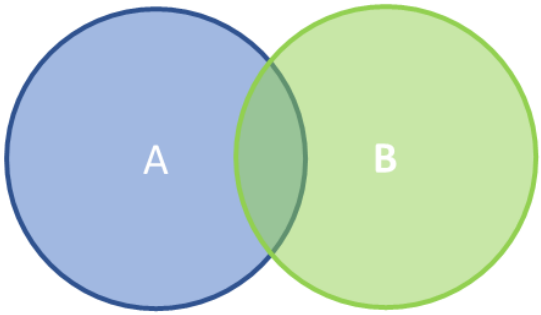


图 4：基本乘积示意图。给定集合 A 和 B，基本乘积为：$A \cap B$、 $A \cap B$、 $A \cap B$ 和 $A \cap B$。

3.2.2 构造实体几何生成与最小化

现在，我们可以给出第一个算法，用于根据给定的边界表示生成CSG表达式：一个实体被定义为完全包含在该实体内部的基本乘积的并集。这一思想在算法1中得到了形式化。

算法1 使用基本乘积进行边界表示到构造实体几何转换

构造一组半空间，使其足以表示实体的CSG表示。找出所有完全位于该实体内部的基本乘积。对所有这些基本乘积取并集。

该算法简单地表达了这样一个布尔逻辑事实：任何布尔函数都可以表示为完全析取范式。它还与DE表示（2）直接相关。这种方法的问题在于，它过于冗长，并且会导致对象的表示形式不够直观。Shapiro和Vossler[SV91a] 注意到了使用算法1构建的CSG表示的最小化问题与开关函数最小化问题（也称为布尔函数最小化）之间的相似性。在布尔逻辑中，蕴含项是一个乘积项（或文字的合取），其为真的性质意味着某个布尔函数为真。在我们的几何场景中，蕴含项是完全包含于待表示实体内部的半空间交集。如果无法从蕴含项中删除任何半空间文字，则该蕴含项就是主蕴含项。也就是说，如果从蕴含项中删除任何半空间文字，所得结果将不再包含于待表示实体内部（因此也不再是蕴含项）。CSG表达式的最小化等价于为该实体寻找一个主蕴含项覆盖。这是一个困难问题（NP完全问题）。Shapiro和Vossler在 [SV91a] 中提出，使用近似算法来寻找最小覆盖，即计算主蕴含项近似最小覆盖的算法。

我们可以做得更好：注意在最佳情况下，每个半空间或其补集都会在 CSG 表示中出现一次。支配半空间 是一个完全包含在实体内或实体补集中的半空间。提取支配半空间可以简化 CSG表达式。

这对应于　　上文介绍的 DTE 表示（4），并由此得到算法2。Buchele 和

算法2 通过主导基元因式分解进行边界表示到构造实体几何转换

实体的支配半空间因子。
如果结果式为空（或为全集）**则**
返回 // 我们有一个构造实体几何表达式
else
迭代尝试对结果式进行分解。
如果结果式不可分解，则计算主蕴含项的近似最小覆盖。
结束 if

Crawford 在 [BC04] 中提出了该算法的改进版本。当结果式不可分解时，他们没有计算主蕴含项的近似最小覆盖，而是改为用其中一个半空间对结果式进行分割，并对每个得到的子集应用分解。

3.3 范围图像到构造实体几何

令人惊讶的是，早期的研究曾尝试直接从物体的范围图像中恢复CSG表达式 [LC88, Che92]。首先对范围图像进行分割，并为每个子集拟合隐式二次曲面。与许多其他系统类似，这些研究假设物体由立方体、球体、圆柱体、圆锥体和椭球体构成。随后，确定拟合基元的正负属性，并利用优先关系图，将其与相邻基元通过并集或差集进行组合。这些研究采用了一个严格且具有局限性的假设：每个基元最多只能被差集运算一次。

4 从非结构化数据中提取构造实体几何

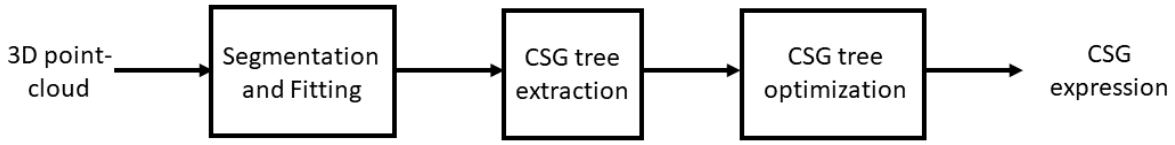


图5：从输入三维点云中提取CSG表达式的典型流程。

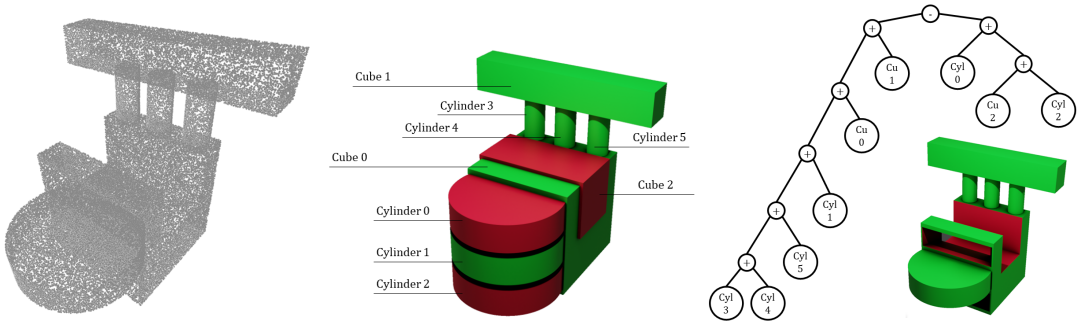


图6：图5所示流程的一个实例。左：输入点云；中：分割后的基元；右：优化后的CSG树/CSG表达式及对应的实体。

现在，我们将注意力转向从非结构化输入数据集（三维点云、三角形网格或三角形集合）中恢复CSG模型这一问题。提取CSG表达式的常见流程

三维点云的转换过程如图5所示，图6展示了一个示例。该流程在 [FPK+08]中首次提出。首先，将输入的三维点云分割为若干子集，并对每个子集进行基元拟合。分割和基元拟合本身就是一项具有挑战性的任务，也是多篇论文的研究主题。例如，可以参阅近期的综述 [KYZB19]。我们还将在第4.1节中讨论一些现有的拟合方法。下一步是提取一个CSG表达式，该表达式涉及前一步得到的基元，用于描述与输入三维点云相对应的物体。最后，还可以通过一个可选步骤进一步优化CSG表达式。有趣的是，这一流程并非总是被采用，一些方法尝试在构建CSG树的同时发现基元，从而将流程中的不同步骤合并在一起。此类方法的示例包括 [SFV+05、SGL+18、RZC+21]等。

4.1 基元拟合与形状分解

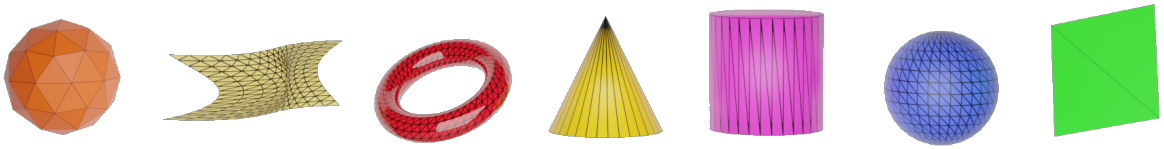


图7：CSG恢复中通常使用的几何基元列表。

重建流程的第一步是对输入的三维点云进行分割，并为不同子集拟合几何基元。图7列出了恢复CSG表达式时通常考虑的基元，这些基元也构成拟合与分割步骤的候选列表。随后，可将拟合得到的几何基元集合用于CSG树提取步骤。有些方法在这第一步之后便停止，并尝试直接将实体表示为几何基元的并集（3）。当拟合基元的层级足够高时，例如[GCV+19、PUG19]，这种做法是可行的。

4.1.1 基于CAD的分割与拟合

逆向工程和CAD中的分割与拟合技术通常分为自底向上或自顶向下两类方法 [VMC97]。前者采用区域生长技术，从种子点开始处理。 [BMV01], 介绍了此类方法的一个示例，其中描述了一种从无组织点云重建边界表示的系统。自底向上方法的问题包括难以选择种子点，以及由于噪声的存在，难以判断是否应向给定区域添加点。自顶向下方法在图像分割中十分常见，但在曲面分割任务中较少使用。其主要难点在于找到合适的准则，以确定应在何处以及如何细分输入点云。

另一种技术是根据特征曲线网络计算区域边界。基于最小二乘拟合的常用曲面拟合技术可应用于每个区域。这是 [VFT07] 提出的方法，其中利用莫尔斯理论对输入点云进行三角剖分，以计算这些特征曲线。 [MLM01] 提出了对多种常见基元（球体、圆柱体、圆锥体和环面）进行最小二乘拟合的算法。

另一类技术基于将更复杂的参数化基元拟合到点云上。这涉及通过非线性最小二乘法（例如 Levenberg-Marquardt 算法）优化非线性函数。例如，这种方法已应用于工业工程领域[Rab06]。

4.1.2 基于 RANSAC 的方法

RANSAC 是一种常用于分割和拟合步骤 [FB81]的方法。它的主要优势在于能够处理数据中的噪声。在几何计算领域，[SWK07] 中描述的高效 RANSAC 变体是最受欢迎的方法。RANSAC 的核心思想很简单：选取少量点（这些点足以定义一个几何基元），直接计算相应基元的参数，检查输入点云中有多少个点与该基元的距离足够近，最后根据统计学原理决定是否接受该基元。Schnabel 等人提出的高效 RANSAC 方法 [SWK07]在应用于三维点云和三维几何基元时，采用了额外的启发式方法来加速这一过程。 [LWC+11] 在高效 RANSAC 方法中引入了基于图的约束，旨在通过考虑平面的平行性、平面的正交性等额外约束，提高 RANSAC 拟合的鲁棒性。

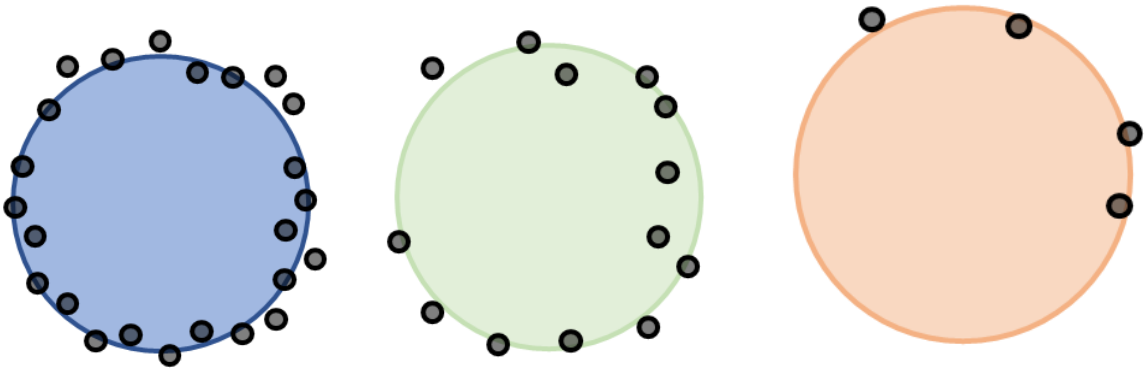
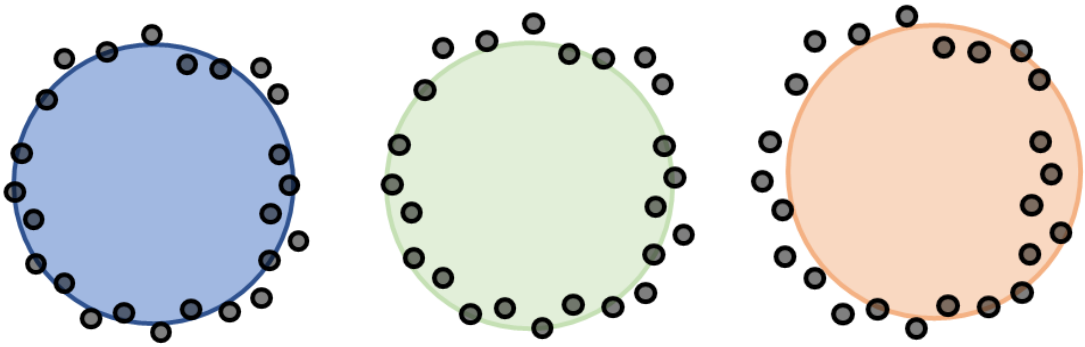


图 8：RANSAC 的一次运行。由于噪声，部分点与拟合的圆（最左侧）距离不够近。随后，这些点被拟合到不同但彼此接近的圆上（中间和右侧）。



Fig图 9：由 RANSAC 分割并拟合的含噪点云和多个彼此接近的圆 .

尽管广受欢迎，基于 RANSAC 的方法仍存在一些问题：它们依赖少数几个有时不太直观的参数（不同参数会产生不同的基元/参数），或者该方法具有随机性（因此，不同运行会产生不同的基元/参数）。图 8 和图 9 展示了这一现象的二维示例。图 9 展示了 RANSAC 的若干次迭代。给定第一个拟合出的圆，有些点距离该圆不够近，因此被留在外面。随后，这些点在不同步骤中被拟合到其他但相近的圆上（中间和右侧图像）。请注意，使用略有不同的

参数（例如点的最小支持数，或将点匹配到基元时使用的距离阈值）可能会产生不同结果。在图 9 中，RANSAC 对含噪输入点云进行分割和拟合，每张图像对应一次使用略有不同参数的 RANSAC 运行。同样，生成的输出是一组彼此接近的圆。在实践中，可以使用略有不同的参数值（从给定分布中采样）多次运行 RANSAC，然后合并相近的基元。如果两个基元的定义参数均在给定阈值范围内，则认为它们彼此接近。改进 RANSAC 输出的另一种方法是尝试发现并强制基元之间满足约束（平行性、垂直性等）。 [LWC+11] 中采用了这种方法。

4.1.3 基于深度学习的方法

用于分割和基元拟合的深度学习方法 RANSAC 对多个参数的依赖性，是 [LSD+19] 中所述监督拟合方法的主要动机之一。该方法将传统拟合方法与一种基于 Point-Net++ [QYSG17] 的深度神经网络相结合，以将基元拟合到点云上。在 Point-Net++ 中加入全连接层，以预测每个输入点的属性（点到基元的归属关系、该点处的表面法线，以及该点所属基元的类型）。每个基元的参数均根据这些点属性，通过可微拟合方法估计。 [YYM+21] 中也介绍了一种类似的点云分割与基元拟合方法。该方法称为 HPNet，其核心是学习多种描述符（语义描述符、谱描述符和尖锐边缘描述符），并学习这些描述符的组合权重。这类方法的一项主要局限是输入点云的大小，通常仅限于几千个点。 [LSC+21] 后来通过引入选择网络解决了这一限制。

在[SLM+20]中，研究者将通常的二次曲面扩展到了参数曲面（B样条曲面片）。该方法使用神经网络将输入点云中的点分类为不同类别，每一类对应一种基元类型（平面、圆柱体、B样条等）。二次曲面通过最小二乘拟合进行估计，而B样条则由另一个神经网络进行拟合。在 [UCS+21]中，研究者提出了广义圆柱面的分割与拟合方法（沿给定曲线对二维域进行扫掠）。该方法结合了用于分割、估计某些量（如表面法线）和基曲线拟合的神经网络，以及用于计算拉伸圆柱参数的求解器。所有这些方法都依赖监督（或至少部分依赖监督）。

前文所述的大多数方法都依赖监督学习，即需要用于训练模型的真实标注。在实践中，这些方法是在从特定对象分布中收集的大型数据集上进行训练的（见第2.1.3节），因此对于来自不同分布的点云，其表现往往较差。为了解决这一缺陷，研究者提出了AutoGPart [LXR+22] 。该方法通过利用编码后的几何先验构建监督空间，并在该空间中搜索最优监督来实现。

基于深度学习的形状分解与形状近似方法 基于神经网络的方法也用于解决形状分解问题。这些方法大多是无监督的，通常仅针对形状近似。它们与并集表达式形式（3）有关，无需经过构造实体几何生成步骤（CSG表达式是所有基元的并集）。总体而言，这些方法只能提供近似结果，通常无法还原尖锐特征，也不一定能得到可用的（可编辑）表达式。

TSG+18]提出了一种通过一组简单体积基元（盒体基元）学习近似（或抽象）形状的方法。该方法局限于使用数量较少且固定的盒体基元（3 至 6 个）。YC21], 提出的相关方法则通过基于点云的联合分割，采用无监督学习来学习长方体形状抽象。另一种可能的推广方式是考虑分解为一组凸多面体，DGY+20]对此进行了提议。PUG19]提出了用更一般的超二次曲面基元替代盒体/长方体/凸多面体。GCV+19]提出了通过椭球体/高斯体集合从数据中学习通用形状模板。使用

简单基元会导致对形状的粗略近似，PKGF21]中的作者通过为基元采用不同的表示形式（可逆神经网络）来解决这一问题。这种方法需要以表示形式的大小增加为代价。GCS+20] 提出了另一种可能的解决方案，即使用局部隐式曲面。

4.1.4 分离基元

与边界表示到构造实体几何转换方法类似，拟合到曲面的基元集合可能不足以描述相应的实体，也无法形成定义良好的构造实体几何表示。参见第 3.2.1 节和图 3 右图中的讨论。解决方案是在拟合到点云（或三角形网格）的基元集合中添加额外基元。考虑与给定非平面基元 [FP16] 相关点的包围盒所对应的平面，似乎是自然而然的做法。包围盒可以与坐标系的轴对齐，也可以与点云的主方向对齐（通过主成分分析获得）。

添加此类分隔平面可能仍不足够，因此可以考虑添加其他类型的分离基元，例如为每个圆柱体添加环面，或添加更一般的管道曲面 [FP16]。

4.2 CSG 生成

下面我们转而讨论构造实体几何生成问题。输入是一个非结构化点集（通常为点云，或者三角形网格），以及一个基元列表，其中包括拟合到点云的基元和额外的分离基元。问题在于如何通过 CSG 操作组合这些基元，使相应的实体描述与输入点云相关的区域。这是一项困难的组合搜索任务。

4.2.1 基于布尔逻辑的方法

解决这一问题的第一种可能方法，是调整第3.2.2节中所述的边界表示到构造实体几何转换方法。具体而言，我们希望使用算法1、算法2，或基于半空间分割的方法 [BC04]。

为此，我们需要为点云构建点成员分类（PMC）函数。该函数将用于对基本乘积进行分类（见算法1）。获取PMC函数的典型方法，是首先根据点云进行曲面重建（例如使用 [KBH06]）。如果将隐式曲面拟合到点云上，则可将其用于点成员分类查询；否则，可以使用针对三角形网格的传统点分类方法 [BDS+18]。另一种方法是对点云进行体素化。在这里，我们关注的是对包围点云采样来源曲面的体积进行体素化（实体体素化）。同样，这要求首先对输入点云进行曲面重建，然后对由三角形网格界定的域进行体素化。这些额外的数值计算可能存在脆弱性，或可能给数据引入额外的近似误差。尽管如此，对于调整用于将B-rep模型转换为构造实体几何模型的算法，以及实现下文将要介绍的若干算法而言，它们都是必要的。

Xiao 和 Furukawa 用于博物馆自动构建 [XF12, XF14]的方法，可视为与边界表示到 CSG 转换中采用的基于布尔逻辑的方法相关。该方法先预先计算所有可能的长方体（即基元），再通过贪心策略添加它们（对应并集）或移除它们（对应差集）。为加快计算，该方法分为两个步骤：第一步在二维空间中进行（基元为矩形）。随后利用得到的二维构造实体几何模型生成候选三维长方体基本体，从而限制组合爆炸。接着通过贪心策略添加或移除候选长方体。该方法使用一个衡量体积拟合和曲面拟合的目标函数来指导搜索。

该方法同样基于布尔逻辑和启发式方法，用于从三维点云 [WXW18]重建构造实体几何模型。该方法并不局限于平面，而是使用所有典型的二次曲面基本体

（平面、球体、圆柱体、环面、圆锥体）。它采用典型流程，包括点云分割和拟合（使用 RANSAC），随后进行 CSG恢复。CSG恢复阶段与 Shapiro 和 Vossler [SV91b], 提出的边界表示到 CSG方法相似，但增加了用于执行组合搜索的启发式方法。作为额外的优化，他们将平面组合成长方体，并在基元拟合步骤中，通过与其有向包围盒求交集来裁剪其他二次曲面基本体。这一优化能够简化组合搜索过程。

[FIFLP22] 中描述的方法同样采用完整的流程，从点云中恢复CSG模型。分割和实体基元拟合步骤是对[FIFLP20]中所述方法的一种变体。CSG恢复步骤是算法2的一种变体，其中首先从实体中提取主导基元，然后通过遗传算法（GA）计算结果实体。该遗传算法采用两种不同的策略：基于选择的遗传算法和基于节点的遗传算法。在基于选择的遗传算法中，一个个体由一个位串表示，每个剩余基元对应一位。根据各基元的分类结果，决定将其添加到结果中或从结果中移除。另一方面，基于节点的遗传算法通过采用与下文第4.2.2节所介绍技术类似的方法，对结果实体的CSG子树进行演化。

4.2.2 基于进化的方法

进化计算是受生物进化启发的一系列优化技术。这些技术有助于近似求解复杂优化问题的全局最优解。典型的进化算法如图10所示的示意图所示。所研究的问题在于恢复与给定对象采样相对应的CSG表达式（或等价地，CSG树）。为此，采用遗传编程（GP）是合理的。遗传编程是一种由John Koza发明的进化技术，旨在进化满足由目标函数编码的判据的计算机程序（树） [Koz92]。

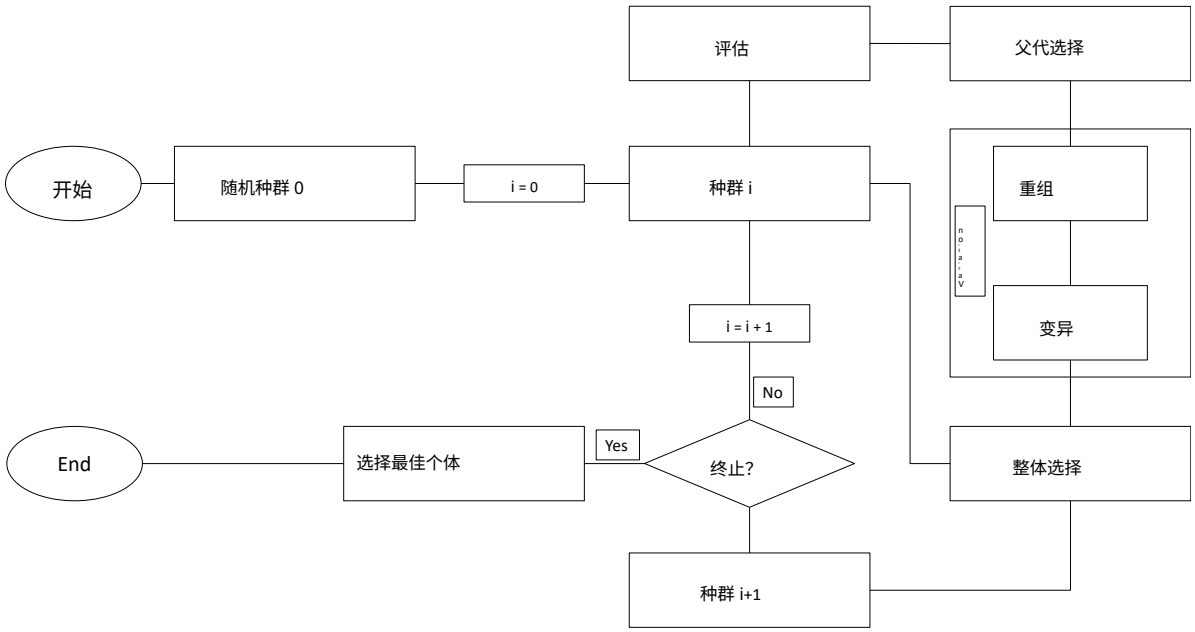


图10：典型进化算法示意图。

最早使用遗传编程从点云演化出CSG树的研究之一，是Silva等人的工作 [SFV+05]。与图5所示的流程不同，他们使用遗传编程自动演化这两种。

构造实体几何（CSG）树以及基元的参数是同时进行的。这种特殊方法使得无法使用标准遗传编程（GP），因此改用强类型遗传编程 [Mon95]。这里考虑了两种类型：第一种对应于基元（圆柱体、长方体等）和 CSG 操作；第二种对应于基元的参数。为了限制生成的 CSG 表达式的复杂度（这是基于 GP 的方法中一个众所周知的问题），采用了简约压力度量[Koz92, LP02, SC09]。遗憾的是，这种方法取得的实验结果相当有限。同时演化基元参数和 CSG 表达式需要投入过多工作，因此，将这两个问题分开处理似乎更加合理。

典型流程——点云分割、拟合和 CSG 恢复——首次出现于 [FPK+08]，如图5所示。输入点云的分割以及将基元拟合到不同簇的过程由遗传算法（GA）完成。为防止 CSG 表达式无限增长，他们决定使用一种简单的遗传算法来选择要使用的基元以及要应用于该基元的 CSG 操作，即确定应将该基元添加到表示中、从表示中减去，还是与目前为止构建的实体取交集。这种基于遗传算法的恢复方法的主要问题是，它只能处理能够完全分解的实体。这种遗传算法方法在思想上更接近于上文所述的某些布尔逻辑方法。

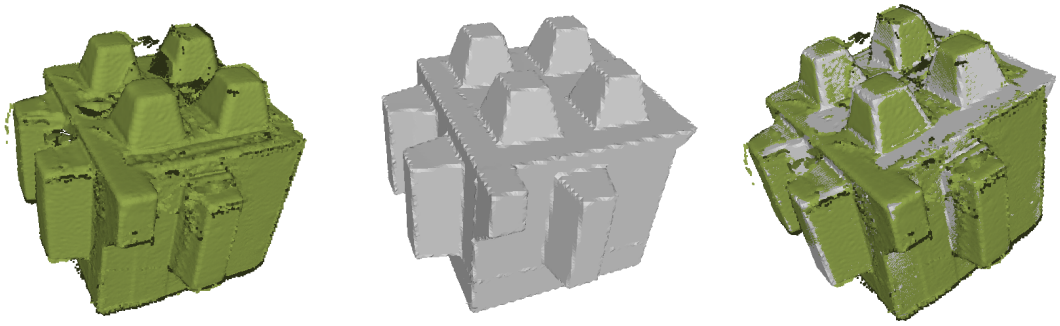


图11：复杂且含噪点云的示例（左图），以及采用 [FP16] 中所述方法恢复的相应 CSG模型（中图）。右图：为了便于可视化，将点云和 CSG 模型同时显示。CSG模型被转换为隐式模型（见第2.4节），并通过移动立方体算法生成网格。

Fayolle 和 Pasko 保留了类似的流程，但提出用功能完备的遗传编程（GP）[FP16]取代较简单的遗传算法（GA）。这解决了该方法 [FPK+08]的主要局限。为了防止 GP 演化出的 CSG树无限增长，他们采用了一种简单的启发式方法：在 GP 优化的目标函数中加入一项惩罚深层树结构的项。与 [FPK+08] 的另一项主要区别是引入了计算分离基元的步骤。研究表明，该方法能够从含噪且复杂的点云中恢复出 CSG模型。图 11 展示了一个示例。该方法有两个问题：首先，它的效率相对较低。遗传编程通过维护一个由个体组成的种群来运行，每个个体都代表一个表达式，其适应度需要在每次迭代时进行评估。在 [FP16], 中，这些个体是 CSG表达式（可能很大），评估其适应度意味着要针对输入点云中的每个点评估这些 CSG表达式。虽然可以采用缓存机制来避免不必要的重复计算，但该方法的效率仍然不高。该方法的第二个问题是，它没有利用基元的任何空间信息，因此相邻基元最终可能在 CSG树中相距甚远，这使得理解或编辑 CSG树变得困难。

Friedrich 等人在后续研究中利用图划分技术解决了这两个问题 [FFPF18, FFGLP19]。给定一组拟合基元（见图 12），通过为每个拟合基元添加一个顶点，并在两个相交的基元之间添加一条边，构建基元相交图。对基元相交图进行划分的一种可行方法是计算所有最大团（最大的完全子图），然后分别独立运行基于 GP 的 CSG 树提取。

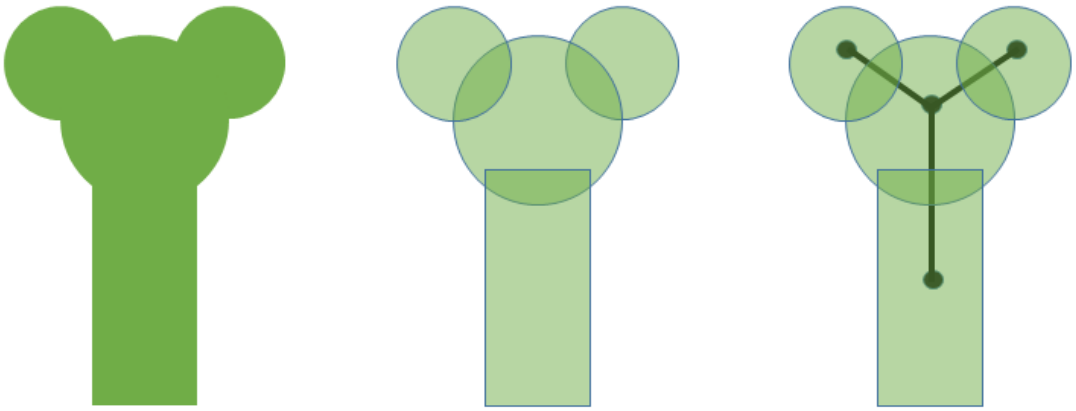


图 12：通过图划分加速进化方法。左：待恢复的实体。中：定义该实体的不同基元（通过 RANSAC 拟合）。右：基元相交图，其中每个基元对应一个顶点，每对相交的基元之间有一条边。

在每个划分（每个最大团）上分别进行处理，最后将对应于各个划分的 CSG 表达式合并，形成最终的 CSG 树 [FFPF18]。这种方法存在两个主要问题：计算图的最大团是一项困难的任务；此外，合并 CSG 树的处理过程也较为复杂。为了解决这些问题， [FFGLP19]采用了一种不同的划分方案：该方案仍然基于基元相交图进行划分，但不再计算最大团，而是通过剪枝特定类型的基元，并断开主蕴含项基元之间的连接（即删除图中对应的边），然后计算图的连通分量。遗传编程（GP）在每个连通分量上独立运行（并且可以并行执行）。最终的 CSG 表达式只需将为各个划分计算得到的 CSG 表达式进行并集运算即可。除了处理速度方面的优势外，这些基于划分的方法的一个主要优点是，每个划分中的基元在空间位置上彼此接近，因此生成的 CSG 树更便于直观编辑（其子树所涉及的基元在空间位置上彼此接近）。

4.2.3 基于程序综合的方法

遗传编程可视为更一般方法——程序综合——中的一种特定类型（随机搜索）。关于程序综合主题的介绍，读者可参阅以下近期综述 [GPS17]。程序综合技术是用于生成满足给定规范的计算机程序（通常以给定的领域特定语言表示）的方法。在我们的案例中，目标是根据给定的规范生成一个构造实体几何程序，该程序由简单基元和布尔运算组成；这里的规范是对象的一种非结构化表示（例如点云、位图图像或体素表示）。近年来，编程语言界对构造实体几何恢复这一问题（有时称为 CAD 反编译）的兴趣日益浓厚。

Du 等人提出结合几何处理和程序综合工具，解决将三维三角网格反编译为 CSG表达式的问题 [DIP+18]。他们并非仅依赖程序综合技术，而是采用了类似于 [FP16] 的流程，但将随机搜索（遗传编程）替换为一种名为“草图综合”的不同程序综合技术 [SL08], ，并对流程中的其他步骤作了若干改进。具体而言，他们使用 Schnabel 等人提出的高效 RANSAC 方法进行基元检测 [SWK07], ，并结合图割 [BK04] 及其他启发式方法改进基元拟合，例如将成对正交/平行的

平面组合成长方体）。通过草图综合为大量基元生成 CSG表达式难以处理，因此他们将输入形状细分为更小、包含较少基元且更易处理的形状。

基于程序综合技术和工具的类似方法被用于将低级三角形网格反编译为 CSG 表达式 [NCGT17, NWP+18, NWA+20, Nan21] 。与 [DIP+18] 类似，其主要方法是搜索。不过，与 Du 等人的方法不同，该方法不依赖几何算法来识别基元及其参数，而是完全依靠程序综合技术。为了引导程序综合，该方法使用评估上下文，从而能够更高效地遍历可能的 CSG 程序的搜索空间。CAD模型通常具有重复的特征或模式。为了正确恢复这些重复模式，Nandi 等人使用后处理技术来捕捉重复内容并提取循环 [NWA+20]。然而，这种方法的开销可能很大，因为它要求先提取一个不含任何循环的程序。

Feser 等人提出了一种基于度量程序合成概念的不同方法来解决 CSG 重建问题 [FDSL22]。他们的方法基于这样一种观察：多个 CSG模型会生成相似但并不完全相同的实体。基于这一观察，他们设计了一种算法，用于生成与输入近似匹配的 CSG 程序，然后通过基于禁忌搜索的局部细化来改进这些 CSG 程序 [GL98]。虽然他们的方法采用了更复杂的综合算法，能够处理用于应对重复模式的重复运算符，但其实验仅限于简单的二维位图图像，且基元仅包括圆盘和矩形，操作仅包括并集和差集。

4.2.4 基于深度学习的方法

基于深度学习的技术已日益成为解决不同应用领域问题的热门工具。最初，这些技术主要用于计算机视觉和自然语言处理领域的问题，但此后其应用领域不断扩展。我们已在第4.1.3节看到深度神经网络可用于解决三维点云分割和基元拟合问题的几种方式。现在，我们关注的是CSG表达式生成，因此也涉及组合优化和语言生成问题。传统上，组合优化并非深度神经网络大显身手的领域；然而，近期的一项工作 [SBK22] 展示了如何使用图神经网络 [BBL+17, BBCV21] 解决组合优化问题，并有望引出一些有趣的研究方向。

另一方面，自然语言处理是深度学习解决方案取得诸多成功的领域，因此，首先从这一方向着手生成CSG表达式似乎合情合理。例如，可以使用循环神经网络生成简单的CSG表达式。CSG-Net [SGL+18, SGL+20]便采用了这种方法。作者使用深度神经网络，将二维图像（像素集合）或三维体素转换为与输入形状相对应的CSG表达式。该方法通过卷积层将输入图像编码至低维潜在空间，再用循环神经网络将潜在向量解码为CSG表达式。当真实标注可用时，采用监督学习进行训练；否则，则使用策略梯度技术。CSG-Net不依赖彼此独立的基元拟合和CSG表达式恢复流程，因此似乎仅适用于非常简单的形状。此外，它仅支持完全可分解模型，因此解决的是一个简单得多的子问题（$O(n^2)$ 而非 $O(2^n)$，其中 n 为基元数量）。

介于程序综合技术与深度学习方法之间的是神经引导综合技术（即利用深度神经网络来引导程序综合）。[ENP+19]描述了一个早期用于CSG 重建问题的神经引导综合示例，其中利用部分构造的模型的输出引导搜索。 [WPL+21]也将神经引导综合用于更通用的CAD 重建。这两种方法的一个共同问题是，都需要投入大量工作来收集数据集，并在这些数据集上训练算法。

另一种可行的研究方向是改进 Shapiro 和 Vossler 提出的方法 [SV91a]，该方法基于将形状分解为规范单元（2）。其思路是构造或“学习”凸单元，并取它们的并集。凸单元由基元的交集构成，而这些基元的参数

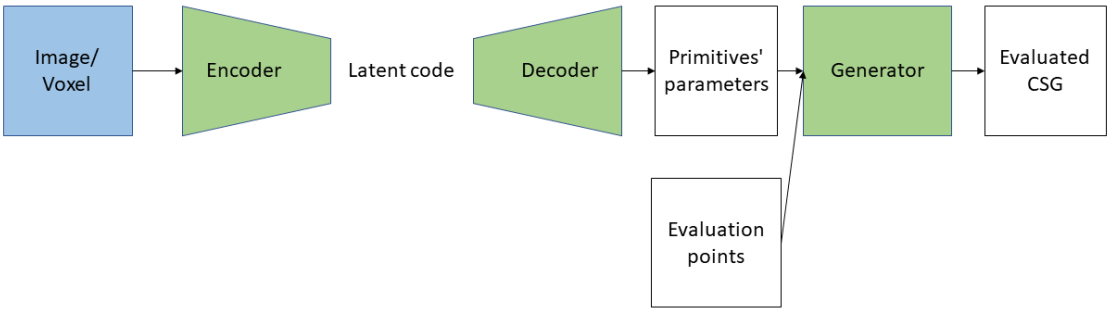


图 13：Cvx-Net [DGY+20]、BSP-Net [CTZ20], 、CSG-Stump [RZC+21] 和 CAPRI-Net [YCL+22]的整体架构。对于 Cvx-Net，生成器会生成固定数量的凸形状，并对它们取并集。对于 BSP-Net、CSG-Stump 和 CAPRI-Net，生成器会学习掩码，用于决定哪些基元通过交集和并集操作进行组合。

由自动编码器生成。Cvx-Net [DGY+20]、BSP-Net [CTZ20],、CSG-Stump [RZC+21] 和 CAPRI-Net [YCL+22]均采用了这种方法。在这些方法中（CSG-Stump 除外），输入是位图图像（二维）或体素（三维），经编码器转换为潜在编码（CSG-Stump 假设输入为三维点云，并使用 DGCNN [WSL+18] 作为编码器）。随后，潜在编码会被解码为基元的参数：Cvx-Net 和 BSP-Net 仅使用平面，CAPRI-Net 使用二次曲面，而 CSG-Stump 使用球体、盒体、圆柱体和圆锥体。Cvx-Net 会对所有凸单元取并集，其他方法则使用可训练矩阵作为掩码，以决定基元/凸单元是否应纳入交集/并集。从实践角度来看，CAPRI-Net 可视为 BSP-Net 的一种推广，即用一般二次曲面取代平面。CSG-Stump 和 CAPRI-Net 在很大程度上相似，仅在公式表述上略有不同。所有这些方法都基于 ShapeNet [CFG+15]中的形状集合进行训练（采用无监督学习）。

值得注意的是，CSG-Stump 的作者还提出了另一种方法 [RZC+21，第 3 节3], 。该方法不基于深度学习，与前文介绍的一些技术有很多相似之处。该方法通过 RANSAC [SWK07], 拟合几何基元，然后通过最小化一个二进制规划问题，求得用于交集、补集和并集操作的二进制掩码。引入深度神经网络的主要目的是处理基元数量非常庞大的情况，因为这类情况在实际中无法通过二进制规划求解器处理。

最后，UCSG-Net [KZK20] 是另一种从原始点云中无监督学习 CSG 表达式的方法。它将用于提取基元参数的自动编码器与门控循环单元 [CvMBB14] 结合起来，以无监督的方式生成解析树。所考虑的基元仅限于盒体和球体，其三维结果相当有限。与前述方法类似，它在 ShapeNet 上以无监督方式进行训练。

基于深度学习的方法可能存在的一个问题是，它们需要在大量数据集上进行训练（即使训练是无监督的），而这会耗费大量时间。此外，这些方法用于为分布外输入数据集生成 CSG 表达式时会产生什么结果，目前也尚不明确。

4.2.5 CSG 优化

一旦针对给定的非结构化输入数据获得了 CSG表达式，最后一步就是对表达式进行优化。首先需要明确的是，我们所说的优化 CSG表达式是什么意思。遵循奥卡姆剃刀，将 CSG表达式的大小作为一项标准似乎是合理的；这里，CSG表达式的大小定义为表达式中基元和操作的数量。这个数量应当尽可能小。例如，我们希望移除冗余的基元和操作（即诸如

这样的表达式应替换为 A）。

假设 CSG表达式的大小已达到最优，另一个值得考虑的标准是 CSG表达式（或 CSG树）的可编辑性。例如，编辑一个子树只应导致形状发生局部修改。我们通过考察空间邻近度来近似 CSG表达式的可编辑性。空间邻近度定义为操作数相互重叠的 CSG 操作数量与操作数彼此不相交的 CSG 操作数量之比。我们的目标是最大化给定表达式的空间邻近度。

大小优化 Shapiro 和 Vossler 最初在[SV91a]中研究了 CSG表达式的优化。从析取范式表达式（2）出发，可以使用开关理论中的技术 [Qui52, MJ56, O'R82] 来最小化该表达式。在布尔逻辑中，蕴含项 是一个乘积项，即由文字构成的合取，它蕴含布尔函数为真。在我们的情形中，蕴含项是基元（半空间）的交集，而布尔函数则是对应于某个实体的 CSG表达式。当一个蕴含项无法进一步因式分解时（即无法移除其中任何文字），它就是主蕴含项 [Qui52]。在简单情形下，布尔函数的所有主蕴含项的析取给出了函数可能具有的最短析取范式。然而，通常还可以做得更好；但寻找最佳析取范式通常很困难，并且属于 NP完全问题，因为它对应于集合覆盖问题的一个实例。Shapiro 和 Vosler 提议使用算法来计算主蕴含项的近似最小覆盖 [SV91a]。

实际上，优先对主导半空间进行因式分解有助于限制生成的 CSG表达式 [BC04] 的大小。如果某个半空间完全包含在目标实体内部或完全位于其外部，则该半空间为主导[SV91a, BC04] （另见第2.3节中的 DTE 表示（4）。

根据所用技术的不同，生成的 CSG表达式可能包含重复基元或子表达式；就其为实体描述提供的信息而言，这些内容是冗余的。在遗传编程领域中，这些冗余内容称为内含子 [Koz92]。可以采用受 [Til84]启发的方法来移除冗余子表达式。具体做法是对 CSG表达式反复应用一组化简规则，直至没有规则可再应用 [FRF+20，第5节1]。

或者，当 CSG表达式由进化方法生成时（见第4.2.2节），它可以在目标（适应度）函数中引入正则化项，以惩罚较大的树（或表达式）。例如， [FP16], 采用了这种方法。另一种做法是考虑多目标优化问题，其中一个目标函数优化 CSG表达式的几何保真度，另一个目标函数优化表达式的大小。[FFGLP19]采用了这种方法。

图14和图15展示了CSG树优化的一个示例。图14展示了一个简单的CAD模型及其对应的CSG树。基元和CSG树是通过第4.1节和第4.2节前面提到的一些技术恢复得到的。该CSG树由31个基元和31个操作组成。通过前文所述技术得到的优化后的CSG树如图15所示，其中包含20个基元和19个操作。

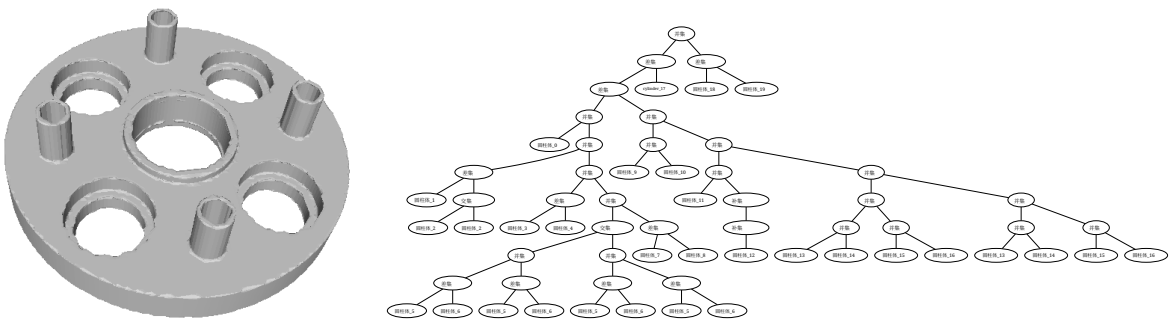
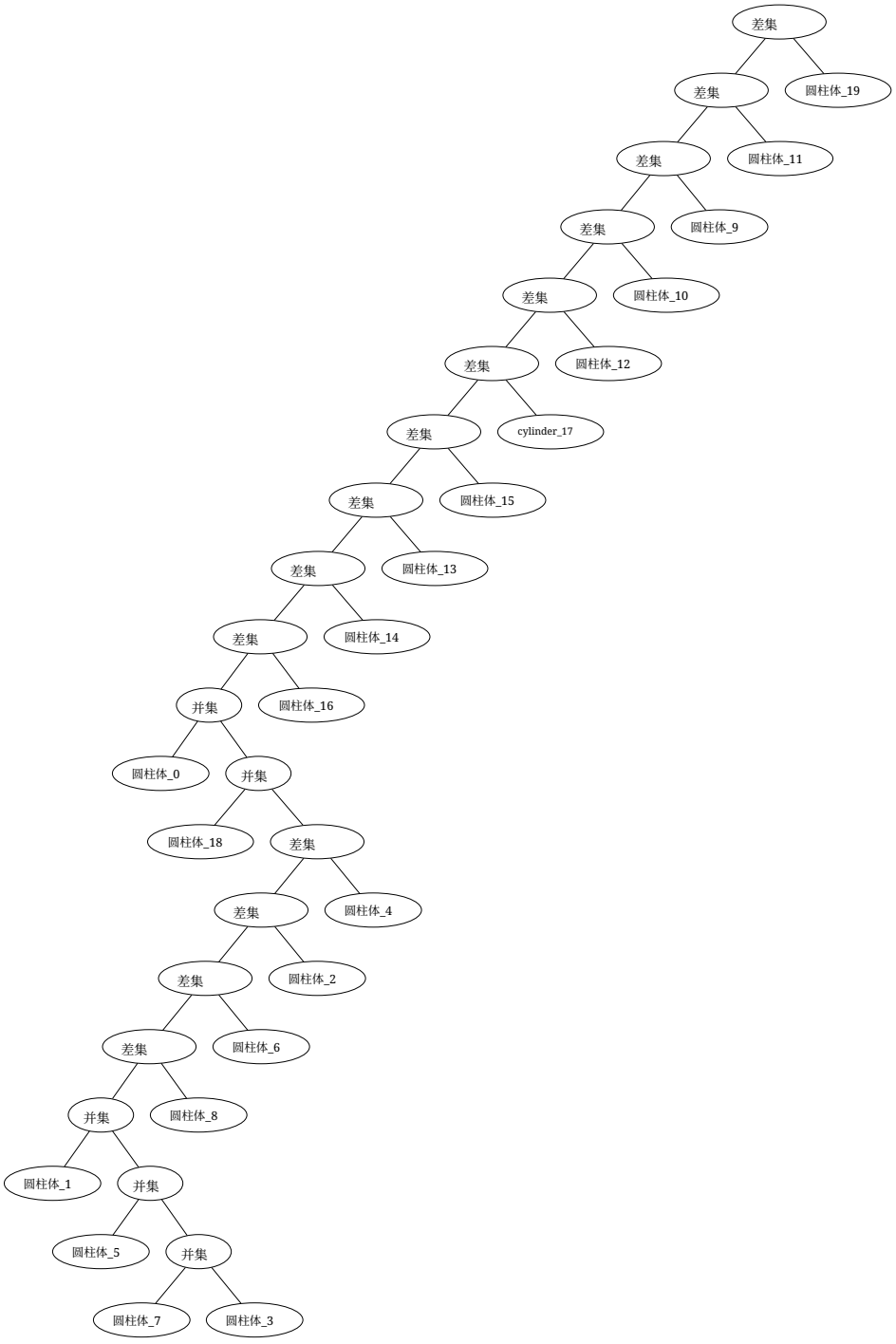


图14：CAD模型及其构造实体几何树示例。



优化后与图 14 对应的 CSG树。图 15：

空间邻近性优化 Andrews 提出，通过从每个基本乘积单元中移除空间上距离较远的基元，来简化 CSG表达式 [And13], ，因为这些距离较远的基元与给定的基本乘积单元无关。这与可编辑性 问题有关，也与这样一个事实有关：编辑给定 CSG模型的子树，只应对形状进行局部修改。当通过进化方法（第4.2.2节）生成CSG表达式时，可以在目标函数中添加一项，以确保布尔运算的两个操作数在空间上相连。例如，可以考虑以下项：

$$f(\Phi) = \frac{P(\Phi)}{\#|\Phi|},$$

其中 #|Φ| 表示 CSG树 Φ中的节点数，而 P 递归定义如下：

- $P(\Phi) = 1$ 若 Φ 是一个基元，
- $P(\Phi) = P(\Phi_1) + P(\Phi_2) + \delta(\Phi_1, \Phi_2)$，当 Φ由子树 Φ_1和 Φ_2构成时；若Φ_1 和 Φ_2 相交，则 $\delta(\Phi_1, \Phi_2) = 1$；否则为 0。

该技术已用于 [FRF+20]。

5 讨论与结论

5.1 程序与高层表示

我们在第4.2.3节中已经看到，CSG表达式可以被视为用一种简单语言编写的程序，而恢复CSG模型可以理解为程序综合的一个例子。更一般地说，我们可以考虑生成表示实体的计算机程序这一问题 [GBL+21]，例如探讨高层表示 [ENP+19]。

用于恢复CSG表达式的程序综合技术仅限于将简单的几何基元（立方体、圆柱体等）与布尔运算（并集、交集、差集）结合起来。然而，这类表达式非常基础，缺少高层结构，例如函数（用于将共同功能分组）或循环，因此难以编辑这些CSG模型。该研究 [NWA+20] 采用一种称为等式饱和的技术 [TSTL09], ，提出通过映射与折叠算子扩展综合技术。他们认为，生成的CSG模型（增加了函数和循环）更易于编辑和操作。

在 [TLS+19] 中，引入了一种用于表示三维形状的领域特定语言（DSL）。该语言支持基本基元、**并集**（或 **连接**）运算符以及 **for** 循环。引入 **for** 循环后，便能高效地表示重复结构，其方式类似于 [NWA+20]。这种方法的一个有趣之处在于采用了自监督学习：神经程序生成器生成表示三维形状的程序，神经程序执行器则根据该程序生成三维形状。

将形状分解为子部件的想法实际上与将计算机程序分解为函数等较小单元的过程非常相似。深度生成模型中的技术可以恢复描述形状各部分如何组成整体的装配结构 [LNX20, JBX+20]。这些方法通常依赖于训练自编码器。编码器为给定的点云或三角形网格生成潜在编码，而解码器则根据该潜在编码生成一个以简化的领域特定语言表达的模型，该语言仅包含有限集合的几何基元。通过在语言中引入宏，并在大型数据集上学习这些宏，可以提取重复模式，从而帮助实现程序化建模 [JCG+21]。关于深度生成建模的最新综述，请参阅 [CRXZ19]。 [JBX+20、JCG+21] 中用于学习生成模型的 DSL 示例展示于图16中，其对应的几何模型显示在右侧。训练是在相对简单的形状集合上进行的，因此相应的语言也可以保持相对简单（长方体基元，以及仿射变换和对称化操作）。

```
bbox = Cuboid(1.845, 0.53, 0.944, True)
cube0 = Cuboid(1.845, 0.045, 0.944, True)
cube1 = Cuboid(0.077, 0.084, 0.777, True)
cube2 = Cuboid(1.547, 0.073, 0.039, True)
attach(cube0, bbox, 0.5, 1.0, 0.5, 0.5,
       1.0, 0.5)
squeeze(cube1, bbox, cube0, bot, 0.063, 0.5)
attach(cube2, cube0, 0.5, 1.0, 0.5, 0.503,
       0.0, 0.115)
reflect(cube1, X)
reflect(cube2, Z)
```

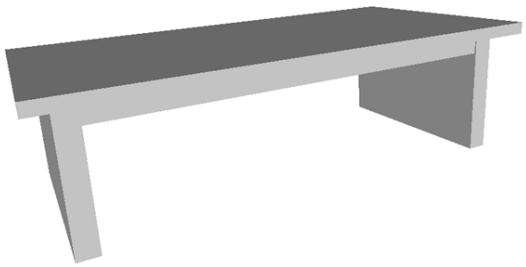


图16：一个 e　来自 [JBX+20、JCG+21] 的 DSL 示例（左）及其对应的三维模型（右)　　).

通过组装零件来构建模型，是计算机辅助设计中的常见理念。 [WJC+21] 中描述的方法及相关的 AutoMate 系统 [JHC+21] 能够学习将零件组装在一起以形成连接部件。目标模型以 B-rep模型表示。

然而，典型模型通常由一系列草图、拉伸和布尔运算构成。 [WPL+21]描述了一种包含草图和拉伸操作的简单语言，其目标是以简洁的程序形式学习 CAD 程序。在这项工作中，序列式 CAD 程序被视为可供学习的马尔可夫决策过程。这项工作还引入了一个以该语言表达的人类设计序列数据集，可用于训练模型。

为了学习生成与给定边界表示相对应的一系列草图、拉伸和布尔运算，Xu 等人提出使用一种称为区域图的新型几何表示 [XPC+21]。一个区域对应于由边界表示面派生的一组实体区域。以区域为节点构建图，并在具有几何邻接关系的节点之间建立边。生成模型对应于在给定区域图所允许的拉伸空间中进行搜索。相关方法 DeepCAD [WXZ21]，将 CAD 模型生成为一系列操作，用于定义二维草图、计算其拉伸，并通过布尔运算将其组合起来。CAD生成网络基于 *Transformer* [VSP+17]。本文未直接讨论将点云转换为 CAD 模型，但[UCS+21]对此进行了处理。给定输入三维点云后，会生成一组拉伸圆柱体，其中一个拉伸圆柱体由一个二维草图和一个拉伸操作组成。

5.2 挑战

展望未来，主要挑战之一是将目标表示从构造实体几何表示扩展到更高层表示：在构造实体几何表示中，实体通过对基元执行布尔运算来构建；而更高层表示则涉及更高层基元，例如由草图和拉伸得到的基元，以及圆角、倒角、斜角、线性拉伸或扫掠等操作。上一节讨论了近期朝这一方向发展的部分研究成果。

CAD 模型可以理解为用领域特定语言 [GBL+21] 编写的程序。从非结构化数据中恢复 CAD 模型的问题，因而可视为多模态学习的一个实例，这是人工智能 [ACD+23] 领域中日益受到关注的课题。

5.3 结语

本文综述了从非结构化数据中恢复 CSG表示的现有方法，并介绍了分割以及将基元拟合到数据等相关问题的求解技术。我们从 CAD 和实体建模领域中将多面体和边界表示转换为.....的技术讲起。

构造实体几何。我们考察了来自不同领域的技术，例如程序综合、进化方法（遗传算法、遗传编程）或基于深度学习的方法。最后，我们讨论了一些当前的研究方向，例如高层表示的恢复，以及用于生成表示实体模型的计算机程序的技术，同时也讨论了一些潜在的现有挑战。

参考文献

[ACD+23] Cem Akkus、Luyang Chu、Vladana Djakovic、Steffen Jauch-Walser、Philipp Koch、Giacomo Loss、Christopher Marquardt、Marco Moldovan、Nadja Sauter、Maximilian Schneider、Rickmer Schulte、Karol Urbanczyk、Jann Goschenhofer、Christian Heumann、Rasmus Hvingelby、Daniel Schalk 和 Matthias Aßenmacher。多模态深度学习，2023年。[And13] James Andrews。用户引导的逆向三维建模。博士论文，加州大学伯克利分校 EECS 系，2013年5月。[Aut] Autodesk。Fusion 360：基于云的三维 CAD、CAM、CAE 和 PCB 软件。https://www.autodesk.co.uk/products/fusion-360/overview。[BBCV21] Michael M. Bronstein、Joan Bruna、Taco Cohen 和 Petar Veli˘ckovi´c。几何深度学习：网格、群、图、测地线与规范。https://arxiv.org/abs/2104.13478，2021年。[BBL+17] Michael M. Bronstein、Joan Bruna、Yann LeCun、Arthur Szlam 和 Pierre Vandergheynst。几何深度学习：超越欧几里得数据。*IEEE*信号处理杂志，34(4):18–42，2017年7月。[BC04] Suzanne F. Buchele 和 Richard H. Crawford。基于隐式边界表示构建三维半空间构造实体几何树。计算机辅助设计，36(11):1063–1073，2004年。实体建模理论与应用。[BDS+18] Gavin Barill、Neil Dickson、Ryan Schmidt、David I.W. Levin 和 Alec Jacobson。面片集合与点云的快速绕数。*ACM*图形学汇刊，2018年。[BK04] Yuri Boykov 和 Vladimir Kolmogorov。用于视觉领域能量最小化的最小割/最大流算法实验比较。*IEEE*模式分析与机器智能汇刊，26(9):1124–1137，2004年。[BMV01] P´al Benko、Ralph R. Martin 和 Tamas Varady。边界表示模型逆向工程算法。计算机辅助设计，33(11):839–851，2001年。[Buc99] Suzanne Buchele。基于代数边界表示的三维二叉空间分割树和构造实体几何树构建。博士论文，德克萨斯大学奥斯汀分校，1999年。[CBC+01] J. C. Carr、R. K. Beatson、J. B. Cherrie、T. J. Mitchell、W. R. Fright、B. C. McCallum 和 T. R. Evans。利用径向基函数重建和表示三维物体。载于第*28*届计算机图形学与交互技术年度会议论文集，SIGGRAPH '01，第67–76页。计算机协会，2001年。[CFG+15] Angel X. Chang、Thomas Funkhouser、Leonidas Guibas、Pat Hanrahan、Qixing Huang、Zimo Li、Silvio Savarese、Manolis Savva、Shuran Song、Hao Su、Jianxiong Xiao、Li Yi 和 Fisher Yu。ShapeNet：信息丰富的三维模型库，2015年。[Che92] Tsu-Wang Chen。基于构造实体几何的范围图像物体识别。博士论文，美国西北大学，1992年。UMI 订单号 GAX92-29889。

[CRXZ19] Siddhartha Chaudhuri、Daniel Ritchie、Kai Xu 和 Hao (Richard) Zhang。《三维结构生成模型》。载于 Wenzel Jakob 和 Enrico Puppo 编，《style id='3'>Eurographics 2019 - 教程》。Eurographics协会，2019 年。[CTZ20] Zhiqin Chen、Andrea Tagliasacchi 和 Hao Zhang。BSP-net：通过二叉空间分割生成紧凑网格，2020 年。[CvMBB14] Kyunghyun Cho、Bart van Merrienboer、Dzmitry Bahdanau 和 Yoshua Bengio。关于神经机器翻译的性质：编码器-解码器方法。https://arxiv.org/ abs/1409.1259，2014 年。[Dey91] Tamal Dey。《style id='11'>三维多面体的分解》。博士论文，普渡大学，1991 年。[DGHS88] David Dobkin、Leonidas Guibas、John Hershberger 和 Jack Snoeyink。寻找简单多边形的 CSG表示的高效算法。《style id='15'>SIGGRAPH计算机图形学》，22(4)：31–40，1988 年。[DGY+20] Boyang Deng、Kyle Genova、Soroosh Yazdani、Sofien Bouaziz、Geoffrey Hinton 和 Andrea Tagliasacchi。CvxNet：可学习的凸分解，2020 年。[DIP+18] Tao Du、Jeevana Priya Inala、Yewen Pu、Andrew Spielberg、Adriana Schulz、Daniela Rus、Armando Solar-Lezama 和 Wojciech Matusik。InverseCSG：自动将三维模型转换为 CSG树。《style id='21'>《ACM图形学汇刊》》，37(6)，2018 年 12 月。[ENP+19] Kevin Ellis、Maxwell Nye、Ye wen Pu、Felix Sosa、Josh Tenenbaum 和 Armando Solar-Lezama。编写、执行、评估：使用 REPL 的程序综合，2019 年。[FB81] Martin A. Fischler 和 Robert C. Bolles。随机采样一致性：用于模型拟合并应用于图像分析和自动制图的一种范式。《style id='27'>《ACM通讯》》，24(6)：381–395，1981 年 6 月。[FDSL22] John Feser、Isil Dillig 和 Armando Solar-Lezama。用于逆向CSG的度量程序合成，2022 年。[FFGLP19] Markus Friedrich、Pierre-Alain Fayolle、Thomas Gabor 和 Claudia Linnhoff-Popien。优化进化式CSG树提取。载于《style id='33'>遗传与进化计算会议论文集》，GECCO 2019，第 1183–1191 页。计算机协会，2019 年。[FFPF18] Markus Friedrich、Sebastian Feld、Thomy Phan 和 Pierre-Alain Fayolle。通过图划分加速进化式构造树提取。载于《style id='37'> WSCG国际计算机图形学、可视化与计算机视觉会议论文集》，2018 年。[FIFLP20] Markus Friedrich、Steffen Illium、Pierre-Alain Fayolle 和 Claudia Linnhoff-Popien。将实体基元分割并拟合到三维点云的混合方法。载于第*15*届计算机视觉、成像与计算机图形学理论及应用国际联合会议论文集——第*1*卷：*GRAPP*》，第 38–48 页。INSTICC，SciTePress，2020 年。[FIFLP22] Markus Friedrich、Steffen Illium、Pierre-Alain Fayolle 和 Claudia Linnhoff-Popien。使用混合方法从三维点云和网格中提取 CSG树。载于 Kadi Bouatouch、A. Augusto de Sousa、Manuela Chessa、Alexis Paljic、Andreas Kerren、Christophe Hurter、Giovanni Maria Farinella、Petia Radeva 和 Jose Braz 编，《style id='45'>计算机视觉、成像与计算机图形学理论及应用》，第 53–79 页。施普林格国际出版，2022 年。[FP16] Pierre-Alain Fayolle 和 Alexander Pasko。从三维点云中提取对象构造树的进化方法。《style id='49'>计算机辅助设计》，74：1–17，2016 年。

[FPK+08] Pierre-Alain Fayolle、Alexander Pasko、Elena Kartasheva、Christophe Rosenberger 和 Chris- tian Toinard。《体积模型构建自动化》，第214–238页。Springer Berlin Heidelberg，柏林，海德堡，2008年。[FRF+20] Markus Friedrich、Christoph Roch、Sebastian Feld、Carsten Hahn 和 Pierre-Alain Fayolle。用于优化构造实体几何树的灵活流程。《计算机科学研究札记》，2020。[FS09] Philippe Flajolet 和 Robert Sedgewick。《解析组合学》。剑桥大学出版社，美国，第1版，2009年。[GBL+21] Yaroslav Ganin、Sergey Bartunov、Yujia Li、Ethan Keller 和 Stefano Saliceti。作为语言的计算机辅助设计，2021年。[GCS+20] Kyle Genova、Forrester Cole、Avneesh Sud、Aaron Sarna 和 Thomas Funkhouser。用于三维形状的局部深度隐式函数。载于《IEEE/CVF计算机视觉与模式识别会议论文集》，第4857–4866页，2020。[GCV+19] Kyle Genova、Forrester Cole、Daniel Vlasic、Aaron Sarna、William T. Freeman 和 Thomas Funkhouser。使用结构化隐式函数学习形状模板，2019。[GKOM18] Paul Guerrero、Yanir Kleiman、Maks Ovsjanikov 和 Niloy J. Mitra。PCPNet：从原始点云学习局部形状属性。《计算机图形学论坛》，37(2)：75–85，2018。[GL98] Fred Glover 和 Manuel Laguna。《禁忌搜索》，第2093–2229页。Springer US，波士顿，马萨诸塞州，1998年。[GPS17] Sumit Gulwani、Alex Polozov 和 Rishabh Singh。《程序综合》，第4卷。NOW，2017年8月。[HDD+92] Hugues Hoppe、Tony DeRose、Tom Duchamp、John McDonald 和 Werner Stuetzle。从无序点进行曲面重建。《SIGGRAPH计算机图形学》，26(2)：71–78，1992年7月。[HZ03] Richard Hartley 和 Andrew Zisserman。《计算机视觉中的多视图几何》。剑桥大学出版社，2003年。[JBX+20] R. Kenny Jones、Theresa Barton、Xianghao Xu、Kai Wang、Ellen Jiang、Paul Guerrero、Niloy J. Mitra 和 Daniel Ritchie。ShapeAssembly。《ACM图形学汇刊》，39(6)：1–20，2020年11月。[JCG+21] R. Kenny Jones、David Charatan、Paul Guerrero、Niloy J. Mitra 和 Daniel Ritchie。ShapeMod。《ACM图形学汇刊》，40(4)：1–16，2021年8月。[JHC+21] Benjamin Jones、Dalton Hildreth、Duowen Chen、Ilya Baran、Vladimir G. Kim 和 Adriana Schulz。AutoMate：用于CAD装配体自动配合的数据集与学习方法，2021年。[KBH06] Michael Kazhdan、Matthew Bolitho 和 Hugues Hoppe。泊松表面重建。载于《第四届Eurographics几何处理研讨会论文集》，SGP '06，第61–70页，2006年。[KMJ+19] Sebastian Koch、Albert Matveev、Zhongshi Jiang、Francis Williams、Alexey Artemov、Evgeny Burnaev、Marc Alexa、Denis Zorin 和 Daniele Panozzo。ABC：用于几何深度学习的大型CAD模型数据集。载于《2019年IEEE/CVF计算机视觉与模式识别会议》（CVPR），第9593–9603页，2019年。[Koz92] John R. Koza。《遗传编程：通过自然选择进行计算机编程》。麻省理工学院出版社，剑桥，马萨诸塞州，美国，1992年。

[KYZB19] Adrien Kaiser、Jose Alonso Ybanez Zepeda 和 Tamy Boubekeur。针对采集的三维数据的简单几何基元检测方法综述。《计算机图形学论坛》，38(1)：167–196，2019。[KZK20] Kacper Kania、Maciej Zieba 和 Tomasz Kajdanowicz。UCSG-Net：无监督发现构造实体几何树，2020。[LC88] Wei-Chung Lin 和 Tsu-Wang Chen。基于构造实体几何的范围图像对象识别。载于*[1988《] 9*国际模式识别会议论文集》，第1卷，第99–103页，1988年。[LNX20] Jun Li、Chengjie Niu 和 Kai Xu。学习面向结构感知形状合成的部件生成与装配，2020。[LP02] Sean Luke 和 Liviu Panait。字典序简约压力。载于《第四届遗传与进化计算年会论文集》，GECCO'02，第829–836页，美国加利福尼亚州旧金山，2002年。Morgan Kaufmann出版社。[LSC+21] Eric-Tuan Le、Minhyuk Sung、Duygu Ceylan、Radomir Mech、Tamy Boubekeur 和 Niloy J. Mitra。CPFN：用于高分辨率点云的级联基元拟合网络，2021年。[LSD+19] Lingxiao Li、Minhyuk Sung、Anastasia Dubrovina、Li Yi 和 Leonidas J. Guibas。对三维点云进行几何基元的监督拟合。载于CVPR，第2652–2660页。计算机视觉基金会/IEEE，2019年。[LWC+11] Yangyan Li、Xiaokun Wu、Yiorgos Chrysathou、Andrei Sharf、Daniel Cohen-Or 和 Niloy J. Mitra。Globfit：通过发现全局关系实现基元的一致拟合。《ACM图形学汇刊》，30(4)，2011年7月。[LXR+22] Xueyi Liu、Xiaomeng Xu、Anyi Rao、Chuang Gan 和 Li Yi。AutoGPart：用于通用三维部件分割的中间监督搜索。https://arxiv.org/abs/2203.06558，2022。[MJ56] Edward J McCluskey Jr. 布尔函数的最小化。《贝尔系统技术期刊》，35(6)：1417–1444，1956年。[MLM01] David Marshall、Gabor Lukacs 和 Ralph Martin。存在几何退化时对范围数据中的基元进行稳健分割。《IEEE模式分析与机器智能汇刊》，23(3)：304–314，2001年。[Mon95] David J. Montana。强类型遗传编程。《进化计算》，3(2)：199–230，1995年6月。[Nan21] Chandrakana Nandi。《计算制造的编程语言工具与技术》。博士论文，华盛顿大学，2021年。[NCGT17] Chandrakana Nandi、Anat Caspi、Dan Grossman 和 Zachary Tatlock。用于3D打印的编程语言工具与技术。载于Benjamin S. Lerner、Rastislav Bod´ík 和 Shriram Krishnamurthi（编），《第二届编程语言进展峰会》（SNAPL 2017），Leibniz信息学国际论文集（LIPIcs）第71卷，第10:1–10:12页，德国达格施图尔，2017年。Schloss Dagstuhl–Leibniz信息学中心。[NWA+20] Chandrakana Nandi、Max Willsey、Adam Anderson、James R. Wilcox、Eva Darulova、Dan Grossman 和 Zachary Tatlock。通过等式饱和与逆变换综合结构化CAD模型。第41届ACM SIGPLAN编程语言设计与实现会议论文集，2020年6月。

[NWP+18] Chandrakana Nandi、James R. Wilcox、Pavel Panchekha、Taylor Blau、Dan Grossman 和 Zachary Tatlock。用于编译与反编译计算机辅助设计的函数式编程。《ACM程序设计语言汇刊》，2(ICFP)，2018年7月。[OBA+03] Yutaka Ohtake、Alexander Belyaev、Marc Alexa、Greg Turk 和 Hans-Peter Seidel。多层单位分解隐式表示。《ACM图形学汇刊》，22(3):463–470，2003年7月。[O'R82] Joseph O'Rourke。多边形分解与开关函数最小化。《计算机图形学与图像处理》，18(4):382–391，1982年。[Pet86] D P Peterson。边界到构造实体几何的映射：聚焦二维问题。《计算机辅助设计》，18(1):3–14，1986年2月。[PKGF21] Despoina Paschalidou、Angelos Katharopoulos、Andreas Geiger 和 Sanja Fidler。神经部件：利用可逆神经网络学习表达力强的三维形状抽象，2021年。[PTC] PTC。OnShape。https://www.onshape.com/en/。[PUG19] Despoina Paschalidou、Ali Osman Ulusoy 和 Andreas Geiger。重新审视超二次曲面：学习超越长方体的三维形状解析，2019年。[PY90] Michael S. Paterson 和 F. Frances Yao。用于隐藏面消除和实体建模的高效二叉空间分割。《离散与计算几何》，5:485–503，1990年。[Qui52] W. V. Quine。真值函数化简问题。《美国数学月刊》，59(8):521–531，1952年。[QYSG17] Charles R. Qi、Li Yi、Hao Su 和 Leonidas J. Guibas。PointNet++：在度量空间的点集上进行深度层次特征学习。https://arxiv.org/abs/1706.02413，2017年。[Rab06] Tahir Rabbani。利用点云和图像自动重建工业设施。博士论文，代尔夫特理工大学，2006年。[Req80] Aristides Requicha。刚性实体的表示：理论、方法与系统。《ACM计算调查》，12(4):437–464，1980年12月。[Ros22] Jarek Rossignac。IBNC：用于多面体的集成边界与自然CSG（综述、简化及既有技术整合）。《计算机辅助设计》，150:103296，2022年。[RRQ+21] Chiara Romanengo、Andrea Raffo、Yifan Qie、Nabil Anwer 和 Bianca Falcidieno。Fit4CAD：用于拟合CAD对象中简单几何基元的点云基准。《计算机与图形学》，2021年10月。[RZC+21] Daxuan Ren、Jianmin Zheng、Jianfei Cai、Jiatong Li、Haiyong Jiang、Zhongang Cai、Junzhe Zhang、Liang Pan、Mingyuan Zhang、Haiyu Zhao 和 Shuai Yi。CSG-Stump：一种便于学习、用于可解释形状解析的类CSG表示。载于IEEE/CVF国际计算机视觉会议（ICCV）论文集，第12478–12487页，2021年10月。[SBK22] Martin J. A. Schuetz、J. Kyle Brubaker 和 Helmut G. Katzgraber。利用受物理启发的图神经网络进行组合优化。《自然·机器智能》，4(4):367–377，2022年4月。[SC09] Sara Silva 和 Ernesto Costa。遗传编程中控制膨胀的动态限制，以及对过去和当前膨胀理论的回顾。《遗传编程与可进化机器》，10(2):141–179，2009年6月。

[SFV+05] Sara Silva、Pierre-Alain Fayolle、Johann Vincent、Guillaume Pauron、Christophe Rosenberger 和 Christian Toinard。用于形状建模与拟合的进化计算方法。载于《人工智能进展》，第144–155页。Springer Berlin Heidelberg，2005年。[SGL+18] Gopal Sharma、Rishabh Goyal、Difan Liu、Evangelos Kalogerakis 和 Subhransu Maji。CS-GNet：用于构造实体几何的神经形状解析器。载于2018年IEEE/CVF计算机视觉与模式识别会议，第5515–5523页，2018年。[SGL+20] Gopal Sharma、Rishabh Goyal、Difan Liu、Evangelos Kalogerakis 和 Subhransu Maji。用于构造实体几何的神经形状解析器。IEEE模式分析与机器智能汇刊，第1–1页，2020年。[Sha91] Vadim Shapiro。r函数理论及其应用：入门介绍。技术报告，康奈尔大学，1991年。[Sha01] Vadim Shapiro。曲线多边形的凸缺陷树算法。《国际计算几何与应用杂志》，11(02):215–238，2001年。[SL08] Armando Solar-Lezama。基于草图的程序合成。博士论文，加州大学伯克利分校，美国，2008年。AAI3353225。[SLM+20] Gopal Sharma、Difan Liu、Subhransu Maji、Evangelos Kalogerakis、Siddhartha Chaudhuri 和 Rado m´ır M˘ech。Parsenet：用于三维点云的参数曲面拟合网络，2020年。[SMB+20] Vincent Sitzmann、Julien N. P. Martel、Alexander W. Bergman、David B. Lindell 和 Gordon Wetzstein。采用周期激活函数的隐式神经表示，2020年。[SV91a] Vadim Shapiro 和 Donald L. Vossler。CSG表示的构造与优化。《计算机辅助设计》，23(11):4–20，1991年。[SV91b] Vadim Shapiro 和 Donald L. Vossler。二维实体的高效CSG表示。《机械设计杂志》，113(3):292–305，1991年。[SV93] Vadim Shapiro 和 Donald L. Vossler。边界到CSG转换的分离。《ACM图形学汇刊》，12(1):35–55，1993年1月。[SWK07] R. Schnabel、R. Wahl 和 R. Klein。用于点云形状检测的高效RANSAC。《计算机图形学论坛》，26(2):214–226，2007年。[Thi] Thingiverse。https://www.thingiverse.com/。[Til84] Robert B. Tilove。用于构造实体几何的空对象检测算法。《ACM通讯》，27(7):684–694，1984年7月。[TLS+19] Yonglong Tian、Andrew Luo、Xingyuan Sun、Kevin Ellis、William T. Freeman、Joshua B. Tenenbaum 和 Jiajun Wu。学习推断并执行三维形状程序。载于《国际学习表征会议》，2019年。[TSG+18] Shubham Tulsiani、Hao Su、Leonidas J. Guibas、Alexei A. Efros 和 Jitendra Malik。通过组装体积基元学习形状抽象，2018年。[TSTL09] Ross Tate、Michael Stepp、Zachary Tatlock 和 Sorin Lerner。等式饱和：一种新的优化方法。《SIGPLAN通知》，44(1):264–276，2009年1月。[UCS+21] Mikaela Angelina Uy、Yen-yu Chang、Minhyuk Sung、Purvi Goel、Joseph Lambourne、Tolga Birdal 和 Leonidas Guibas。Point2cyl：从点云逆向工程三维对象，生成拉伸圆柱体。arXiv预印本 arXiv:2112.09329，2021年。

[VFT07] Tam´as V´arady、Michael A. Facello 和 Zsolt Ter´ek。数字形状重建中曲面结构的自动提取。《计算机辅助设计》，39(5):379–388，2007。[VMC97] Tam´as V´arady、Ralph R. Martin 和 Jordan Cox。几何模型的逆向工程——简介。《计算机辅助设计》，29(4):255–268，1997。[VSP+17] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N. Gomez、Lukasz Kaiser 和 Illia Polosukhin。注意力就是你所需要的一切，2017。[WJC+21] Karl D. D. Willis、Pradeep Kumar Jayaraman、Hang Chu、Yunsheng Tian、Yifei Li、Daniele Grandi、Aditya Sanghi、Linh Tran、Joseph G. Lambourne、Armando Solar-Lezama 和 Woj- ciech Matusik。Joinable：学习参数化 CAD 连接部件的自底向上组装，2021年。[WPL+21] Karl D. D. Willis、Yewen Pu、Jieliang Luo、Hang Chu、Tao Du、Joseph G. Lambourne、Armando Solar-Lezama 和 Wojciech Matusik。Fusion 360 资源库：一个用于根据人工设计序列以程序方式构建 CAD 模型的数据集和环境。《ACM图形学汇刊》，40(4)，2021年7月。[WSK+15] Zhirong Wu、Shuran Song、Aditya Khosla、Fisher Yu、Linguang Zhang、Xiaoou Tang 和 Jianxiong Xiao。3D ShapeNets：体积形状的深度表征，2015。[WSL+18] Yue Wang、Yongbin Sun、Ziwei Liu、Sanjay E. Sarma、Michael M. Bronstein 和 Justin M. Solomon。用于点云学习的动态图卷积神经网络。https://arxiv.org/abs/1801.07829，2018。[WXW18] Q. Wu、K. Xu 和 J. Wang。根据3D原始点云构建3D构造实体几何模型。《计算机图形学论坛》，37(5):221–232，2018。[WXZ21] Rundi Wu、Chang Xiao 和 Changxi Zheng。DeepCAD：用于计算机辅助设计模型的深度生成网络。载于 IEEE/CVF计算机视觉国际会议论文集，第6772–6782页，2021。[XF12] Jianxiong Xiao 和 Yasutaka Furukawa。重建世界各地的博物馆。载于 Andrew Fitzgibbon、Svetlana Lazebnik、Pietro Perona、Yoichi Sato 和 Cordelia Schmid（编），《计算机视觉——ECCV 2012》，第668–681页，柏林、海德堡，2012。Springer Berlin Heidelberg。[XF14] Jianxiong Xiao 和 Yasutaka Furukawa。重建世界各地的博物馆。《国际计算机视觉期刊》，110(3):243–258，2014年12月。[XPC+21] Xianghao Xu、Wenzhe Peng、Chin-Yi Cheng、Karl D. D. Willis 和 Daniel Ritchie。使用区域图推断 CAD 建模序列。载于2021年IEEE/CVF计算机视觉与模式识别会议（CVPR），第6058–6066页，美国加利福尼亚州洛斯阿拉米托斯，2021年6月。IEEE计算机学会。[YC21] Kaizhi Yang 和 Xuejin Chen。通过点云联合分割进行长方体形状抽象的无监督学习。《ACM图形学汇刊》，40(4):1–11，2021年8月。[YCL+22] Fenggen Yu、Zhiqin Chen、Manyi Li、Aditya Sanghi、Hooman Shayani、Ali Mahdavi-Amiri 和 Hao Zhang。CAPRI-Net：通过自适应基元组装学习紧凑型 CAD 形状。载于 IEEE/CVF计算机视觉与模式识别会议（CVPR）论文集，第11768–11778页，2022年6月。[YYM+21] Siming Yan、Zhenpei Yang、Chongyang Ma、Haibin Huang、Etienne Vouga 和 Qixing Huang。HPNet：使用混合表征进行深度基元分割。https://arxiv.org/abs/ 2105.10620，2021。[ZJ16] Qingnan Zhou 和 Alec Jacobson。Thingi10K：一个包含10,000个3D打印模型的数据集，2016。