变分直接建模：面向计算机辅助设计中参数化建模与直接建模集成的框架

Qiang Zou[a]，*Hsi-Yung Feng[b]、Shuming Gao[a]

[a]*CAD*与计算机图形学国家重点实验室，浙江大学，杭州，*310027*，中国[b]机械工程系，英属哥伦比亚大学，温哥华，*BC*，*V6T 1Z4*，加拿大

arXiv:2301.02999v1 [cs.GR] 8 Jan 2023

摘要

基于特征的参数化建模是计算机辅助设计（CAD）领域事实上的标准。基于边界表示的直接建模是近年来发展起来的另一种 CAD 范式。两者各有优势与局限，因此在集成式 CAD 建模方案方面具有巨大的改进潜力。现有的大多数集成方法由工业界开发，通常将直接编辑视为伪特征，因此很难实现无缝集成。本文提出了一种实现无缝参数化/直接建模集成的替代方法，使参数化编辑和直接编辑能够以统一的方式协同工作。本文首先阐述了参数化/直接建模集成的基本问题与挑战，随后提出一个基于先检测后解决策略的框架，用于处理这些信息不一致问题。此外，本文还给出了可系统地检测并解决所有可能类型的信息不一致的算法，以实现该框架。借助这些算法，整个模型编辑过程中均可保持模型有效性，进而解决直接编辑与参数化编辑之间的差异。基于初步原型的一系列案例研究与对比结果表明，所提出方法是有效的。

关键词：计算机辅助设计、实体建模、参数化/直接建模集成、信息不一致、有效性维护、决策、智能 CAD

1. 引言

计算机辅助设计（CAD）一直是产品设计的主流工业实践。其主要用途之一是创建和编辑产品的计算机模型 [1, 2]。如此定义的 CAD 建模经历了三个主要发展阶段：实体建模、参数化建模和直接建模。实体建模始于 20 世纪70年代，当时有两种相互竞争的方法 [3]：构造实体几何（CSG）和边界表示（B-rep）。它们分别推动了 20 世纪80年代末的参数化建模和 21 世纪初的直接建模的发展。

CSG 方法将实体表示为基本形状通过正则布尔运算依次组合的结果[4–6]。参数化建模沿用了这种程序化工作方式，但引入特征来取代基本形状[7, 8]。这样一来，便能轻松将关联性[1] 嵌入形状中，从而带来自动变更传播、形状重用以及设计意图嵌入等好处[9]。然而，关联性也给CAD模型带来了复杂性和僵化性。最值得注意的是，必须在建模之前仔细规划参数化模型的构建历史，因为这会限制用户在模型建成后可以编辑的内容对现有参数化模型进行更改，需要深入了解其构建历史及其背后的数学原理。基于这些原因，用户往往觉得参数化建模难以学习和使用 [10, 11]。

边界表示法通过指定实体与空域之间的边界来表示实体。该边界以缝合在一起的一组面来存储 [5]。直接建模允许用户直接操纵这些边界面，以获得实体变体 [12, 13]。这种工作方式具有编辑灵活、更新快速和交互直观等优点。从原理上说，这是一种灵活的方法，因为它可以将模型变为任意形状。模型更新速度快，因为只需更新经过修改的边界面，而大部分边界面保持不变。交互直观意味着学习曲线平缓。然而，这些优势也付出了高昂的代价：关联性信息会丢失，参数驱动的自动变更传播也不再可用 [14]。

显然，参数化建模与直接建模各有优势和不足。因此，将二者结合起来，便能兼具其优点，即自动变更传播和高度的建模灵活性。然而，二者的集成并非易事。根本问题在于，对同一 CAD模型进行参数化/直接编辑时会导致信息不一致（第 3.1 节将对此进行详细说明）。如果 CAD 模型中的信息彼此不一致，就会产生无效模型，例如非实体模型和/或过约束模型。

迄今为止，工业界和学术界已提出五种集成策略

*通讯作者。电子邮箱： qiangzou@cad.zju.edu.cn (Qiang Zou)

[1]关联性是指 CAD模型的几何实体通过几何约束等方式相互关联 [9]。

（详见第2节）。在处理上述信息不一致这一根本问题时，这些策略始终试图将直接编辑转换为某些特征操作，而转换过程可能会造成信息损失，因此很难实现无缝集成。我们所说的无缝集成，是指集成后的模型能够提供与未集成的建模系统相同的编辑能力和建模行为。换言之，如果应用直接编辑，模型的行为就像边界表示模型；如果进行参数化编辑，模型的行为就像参数化模型。

本文介绍了我们在实现参数化/直接建模无缝集成方面所做的尝试。本文要研究的根本问题，是解决参数化/直接编辑所导致的 CAD 模型中的信息不一致。首要目标是系统地检测并解决所有可能类型的信息不一致，从而在所有建模操作中满足以下三项技术要求：模型有效性（为实体且约束良好）；模型形状连续变化；以及模型约束变化最小。第 3.2 节将详细讨论这三项要求的定义。

本文的主要贡献包括：

1. 确定参数化/直接集成所涉及的研究问题及其根本挑战；
2. 提出一种称为变分直接建模的框架，以系统地处理参数化/直接编辑后 CAD 模型中的任何信息不一致；3. 将检测和解决信息不一致这一高层任务表述为定义明确的技术问题，例如实体布尔运算和过约束最大化。

以下各节的安排如下：第2节回顾参数化建模、直接建模及二者的集成。第 3 节深入分析参数化/直接建模集成所面临的问题与挑战。第 4 节介绍本文提出的集成方法的总体框架，并在第 5 节中详细阐述。第 6 节给出应用示例，并与领先的商用 CAD 建模软件进行比较；第 7 节总结该方法的优势与局限。

2. 相关工作

参数化建模和直接建模都建立在实体建模之上。实体建模是一种计算机辅助设计建模方法，由当时在罗切斯特大学的 Requicha 和 Voelcker，以及当时在剑桥的 Braid 和 Lang 提出 [15, 16]。实体建模旨在改进先前的线框和曲面建模范式，从而解决这两种范式中出现的信息不完整问题[17]。这种方法非常适合记录已完成的设计，但缺乏设计修改所需的灵活性。

鉴于上述局限，参数化建模通过关联性（以特征和约束的形式）扩展了实体模型，使变更能够自动传播 [9, 18]。自动变更传播带来了设计意图嵌入和几何重用。这使计算机辅助设计从实例建模器转变为“电子主模型”建模器。然而，关联性也意味着复杂性和僵化。要使用参数化模型，需要充分了解其建模历史 [7]。如果所需变更超出了设计的参数化族范围，编辑参数化模型就很困难，甚至不可能，除非重建模型 [19]。由于模型再生成机制总是“从头开始”，更新参数化模型往往耗时。要在所有建模操作中保持模型特征的语义并非易事 [20]。这些因素都使参数化建模器难以学习和使用 [10, 11]。

直接建模是近期为解决上述参数化建模局限性，尤其是其僵化问题而提出的一种方案。它允许用户直接操作模型几何，而无须考虑模型是如何构建的，只需使用抓取、推压和拉动等简单操作 [21, 22]。尽管直接建模最初由工业界提出，但其源头可以追溯到学术界在20世纪80年代开发的局部操作 [23–25]。如今的计算机辅助设计系统中，一些广泛使用的局部操作包括圆角和偏移。直接建模所源自的局部操作是微调操作 [23]。它是直接建模中推拉操作的前身。其关键进步在于：微调不允许模型拓扑出现任何违规，而推拉操作则允许此类违规。通过这种放宽限制，直接建模实现了前所未有的建模灵活性。然而，这也付出了高昂代价：关联性被舍弃，直接建模也缺乏参数驱动的修改能力。

为使用户能够在同一个建模器中兼具直接建模和参数化建模功能，已有一些尝试。据作者所知，目前已有五种关于参数化建模与直接建模集成方法的报道，概述如下。

集成方法 I：伪特征 这种由工业界提出的方法，在基于历史记录的参数化建模器中模拟直接建模。用户指定的直接编辑只需作为伪特征添加到模型构建历史的末尾，原有历史则保持原样 [14, 26]。图 1a 展示了这一策略（该图根据 Siemens NX 的建模行为绘制）。这种方法很容易实现，因此已被大多数 CAD供应商采用。然而，简单地将直接编辑添加到模型历史的末尾，会产生复杂且参数设置存疑的建模历史 [26]。这可能会扰乱模型的参数化信息，并导致有意义的参数化控制丢失。图 1b 所示的示例说明了这一情况：更改尺寸 X 会导致面 F2 不可预测地丢失，进而干扰第二个伪特征处的模型再生成过程。解决这一问题的理想方法不是使用伪特征，而是将用户指定的直接编辑转换为对相关特征的适当重新定义，如图 1c 所示。本研究采用这一策略来解决该问题。

集成方法 II：模式切换 这一方法同样由工业界提出，允许用户在同一款 CAD 软件包中切换

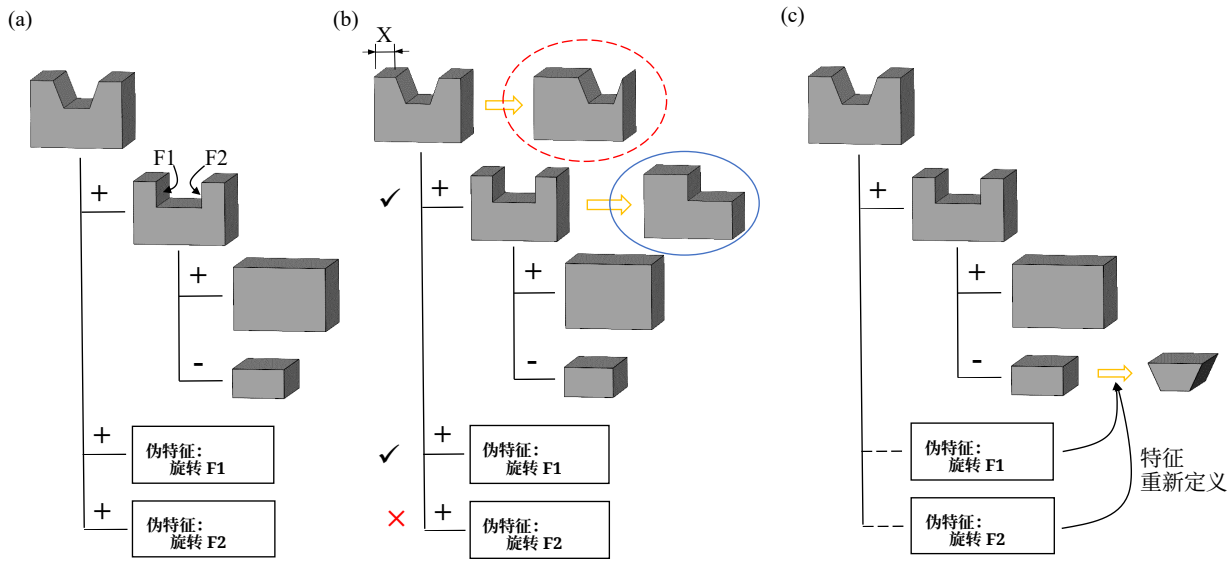


图 1：伪特征方法的局限性：(a) 模型的构建历史；(b) 增大 X 维度无法得到红色虚线圆圈中所示的预期模型，反而会因面 F2 丢失而导致历史重生成失败（如蓝色圆圈中的模型所示）；(c) 一种合理的集成策略是根据直接编辑重新定义槽特征，以避免出现 (b) 中的情况。

直接建模和参数化建模模式。其实现非常简单。从参数化模式切换到直接模式时，参数化模型会降级为边界表示模型，随后即可进行直接建模。不过，从直接模式切换回参数化模式时，它无法恢复任何参数 [27]。

集成方法 III：同步技术 这种方法同样由工业界提出，更具体地说，是由 Siemens NX 提出的。它可以看作是对上述模式切换方法的改进。从参数化模式切换到直接模式时，该方法不会将整个特征模型转换为边界表示模型，而是进行部分转换。部分转换的基本算法如下：将特征分为直接编辑特征和普通特征。只有直接编辑特征会被转换为边界表示模型。建模时，用户将一个普通特征移入直接编辑特征集，然后进行直接建模；但无论是否会造成设计意图丢失，该方法都要求将建模历史中位于被移动特征之前的所有普通特征一并移动 [28]。显然，这会造成不必要的参数化信息丢失。（请注意，同步技术曾一度也指三维变量建模 [29, 30],，但这种解释似乎并未在当前的 Siemens NX 软件包中实现。）

集成方法 IV：操作转换 这种方法由学术界和工业界（Autodesk）共同提出，通过调整模型构建历史中已有特征的参数和/或重新排列这些特征的顺序，将直接编辑转换为操作 [13, 14]。目前，转换是通过启发式方法完成的，例如 [13]。这种做法或许有所帮助，但无法彻底解决问题，因为并非所有直接编辑都能通过这些特征操作实现。更糟糕的是，对于给定的直接编辑（如果可以实现），特征参数的调整方式往往不唯一。如何生成详尽的参数调整选项列表，并从中做出稳健的决策，仍然是未知问题。

集成方法 V：约束式直接建模这一方法同样由工业界提出，它允许用户应用直接编辑，同时通过高效的几何约束求解，在后台保留模型的所有几何约束 [31]，[32]。所保留的几何约束由自动约束识别算法生成，例如，[33]。该方法的缺点在于，识别出的几何约束通常与原始设计意图有所不同。此外，该方法将直接建模降格为一种仅用于进行参数化修改的图形用户界面（GUI）工具。基于这一原因，尽管该方法也称为变分直接建模，但它本质上是传统变量建模的另一种形式，与直接建模关系不大，因此与本文的工作有显著差异。

上述综述表明，在理解和实现参数化建模与直接建模方面已取得显著进展，并且人们尝试了若干方向，旨在将二者集成。已报道的集成方法为本研究带来了诸多启发。尽管如此，这些方法仍处于发展初期；若要实现无缝参数化/直接建模集成，它们仍存在固有缺陷。本研究沿着这一研究方向展开，但采用了一种实现无缝参数化/直接建模集成的新方法。具体而言，我们将提出一个名为变分直接建模的附加模块，以在模型经历参数化/直接编辑时维持信息一致性。不同于采用间接方式工作的现有转换方法，该模块将以直接且系统的方式解决所有可能的信息不一致问题。

3. 参数化/直接集成的问题与挑战

如前所述，要实现无缝参数化/直接建模集成，模型在进行直接编辑时应如同边界表示模型，在进行参数化编辑时则应如同参数化模型。为实现这一目标，需要解决的根本问题是：参数化/直接编辑会导致模型中的拓扑、几何和约束之间出现信息不一致。解决此类信息不一致问题的根本挑战在于：通常存在许多决策选项，而如何在这些选项中进行决策需要有系统的方法。接下来的两个小节将阐释这两点。

3.1. 从集成到信息不一致

CAD模型中有三种基本信息：几何、拓扑和几何约束系统（GCS）。几何与拓扑共同决定CAD模型的实际形状；围绕它们的GCS则表示嵌入CAD模型中的关联性。参数化编辑通过GCS信息层进行，直接编辑则通过几何信息层进行。这里的主要问题是：在当前的模型表示方案中，编辑某个信息层时，所做的更改无法自动反映到其他信息层。因此，编辑前模型中三个信息层之间的一致性遭到破坏，继而在参数化/直接编辑后生成无效模型。

可能存在三种类型的信息不一致：

1. **几何-拓扑不一致（GTI）**进行直接编辑后，更改后的几何可能以两种方式与未更改的拓扑不一致：（1）拓扑中的某些连接不再能够形成，例如，推拉后，原本相连的两个平面变为平行；（2）形成额外的连接，如图2所示。无论哪种情况，都会导致无效边界面 [12], ，例如开放面、自相交面和非流形面。这些面会破坏编辑前模型的有效性。（关于B-rep实体模型有效性的正式定义，请参阅Mantyla的著作 [34] 或我们此前的工作 [12]。）

2. **形状关联不一致（SAI）** 在直接编辑后（并成功解决GTI），模型会呈现新的形状。该形状或其一部分将具有新的边界面和尺寸，导致模型 GCS 中的某些几何约束不再适用。以图3第二行中的模型为例。旋转推拉操作前，它是一个具有六个几何约束的长方体。操作后，它的顶面消失，右侧面变为倾斜面。因此，与这两个发生变化的面相关的所有几何约束都与新形状不兼容，需要进行约束更新。使用新的模型形状更新模型 GCS，将从模型 GCS 中删除一些现有约束，和/或添加新的约束，这可能会破坏模型的约束良好状态。仍以同一模型为例。根据新形状（即棱柱）进行约束更新后，第二个几何约束会取得新的参数值，而第三个约束则应被删除。因此，会生成一个约束不足的 GCS。图中的第一行和第三行展示了两种 GCS 更新情况：约束状态保持不变，以及约束状态分别从约束良好变为过度约束。

3.**形状关联不一致（SAI）B** 参数化编辑后，模型 GCS 的某些参数发生了变化，原始模型形状与 GCS 之间的一致性随之遭到破坏。

第三种不一致类型可通过现有研究成果加以处理，而前两种则需要新的输入。对于第三种类型，核心任务是根据给定的新参数值求解模型 GCS，然后相应地更新模型形状。几何约束求解是一个成熟的领域，已有许多有效算法可供使用 [26, 36]。值得注意的是，Gonzalez-Lluch 等人近期的一项研究 [37], 揭示，用户指定的 GCS 往往包含冗余约束，这可能会妨碍设计可重用性。因此，在处理 SAI （ B ）之前，有必要使用诸如 [37] 中所介绍的工具消除这类冗余。基于这一原因，本文不会聚焦于第三种类型。除非另有说明，下文中的 SAI 均专指 SAI（A）。

总而言之，直接编辑导致的几何变化可能与未改变的拓扑和约束产生不一致，进而生成无效的实体模型和/或 GCS。对于 GTI，其原因可能是丢失旧连接或插入新连接。这些问题表现为无效边界面。对于 SAI，问题源于约束过多或约束不足，无法限制模型几何的自由度（DOF）。这些问题表现为过约束（包括冗余约束）或欠约束。

3.2. 从信息不一致到决策

GTI 和 SAI 需要修正模型拓扑和 GCS，以适应几何形状的变化。解决信息不一致并非易事，因为通常存在许多决策选项。图4展示了一个 GTI 示例。在此示例中，对盲孔模型应用了旋转推拉直接编辑，其中蓝色面为推拉面，橙色面表示目标位置。如果不修正原始拓扑以适应该面的新位置，就会生成无效模型；请看第一行最右侧的模型，其中出现了自相交（表明模型无效）。即使只有一个无效边界面，也有多个选项可用于解决无效问题并生成有效边界环，如图中第二行所示的三个示例选项。然而，相邻面的决策选项彼此耦合，因为要使最终模型有效，已修正面上的任何边都应与相邻面共享，以满足流形条件，同时还必须确保相邻面边界良好。

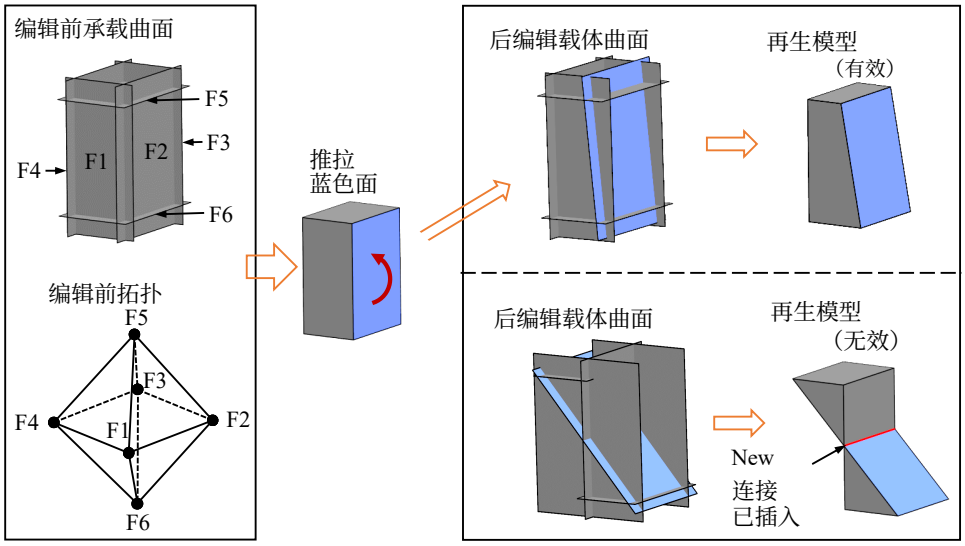


图2：几何-拓扑不一致的示例（蓝色面：推拉面；红色箭头：推拉方向）。模型再生成是指使用编辑后曲面和编辑前拓扑重新评估边界。

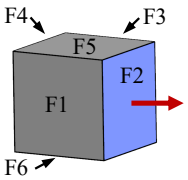
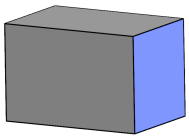
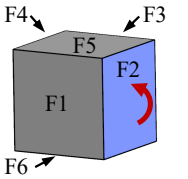
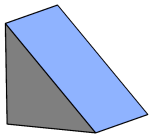
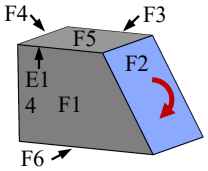
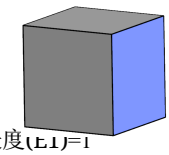
编辑前模型		编辑后实体模型	更新后的模型	状态变化
实体模型	GCS		GCS	
	1. 距离(面1,面3)=1 2. 距离(面2,面4)=1 3. 距离(面5,面6)=1 4. 垂直(面1,面5) 5. 垂直(面1,面4) 6. 垂直(面4,面5)		1. 距离(面1,面3)=1 2. 距离(面2,面4)=2 3. 距离(面5,面6)=1 4. 垂直(面1,面5) 5. 垂直(面1,面4) 6. 垂直(面4,面5)	Well To Well
	1. 距离(面1,面3)=1 2. 距离(面2,面4)=1 3. 距离(面5,面6)=1 4. 垂直(面1,面6) 5. 垂直(面1,面4) 6. 垂直(面4,面6)		1. 距离(面1,面3)=1 2. 角(面2,面4)=135° 3.（已移除） 4. 垂直(面1,面6) 5. 垂直(面1,面4) 6. 垂直(面4,面6)	Well To 下方
	1. 距离(面1,面3)=1 2. 距离(F5,面6)=1 3. 垂直(面1,面5) 4. 垂直(面1,面4) 5. 垂直(面4,面5) 6. 角(面2,面4)=150° 7. 角(面2,面5)=60° 8. 长度(E1)=1		1. 距离(面1,面3)=1 2. 距离(面5,面6)=1 3. 垂直(面1,面5) 4. 垂直(面1,面4) 5. 垂直(面4,面5) 6. 平行(面2,面4) 7. 垂直(面2,面5) 8. 长度（E1）=1	Well To Over

图3：模型 GCS 更新及约束状态变化结果（蓝色面：推拉面；红色箭头：推拉方向） [35]。请注意，“编辑前模型”下方的前两列被视为一列。

例如，如果选择三个示例选项中最上方的一个，就无法通过相邻面上的其他决策选项与之匹配，从而得到有效模型。因此，在针对各个无效边界面的大量有效决策选项中，只有少数组合能够使整个模型有效。大多数组合都会导致解析失败。因此，GTI 解析中的决策容易出错。

除了有效性问题之外，可预测结果的生成建模结果也颇具挑战。在众多消解选项中，有些会导致无效的建模结果；而在有效结果中，也只有一种符合设计者的意图。例如，图4中的三个示例选项中，中间的消解方案会得到一个总体有效但缺少一个孔的模型，而底部的方案则能得到令人满意的结果。由于涉及设计意图 [38]，始终获得预期结果并不容易。为使问题更易于处理，本研究将可预测性要求稍作放宽，

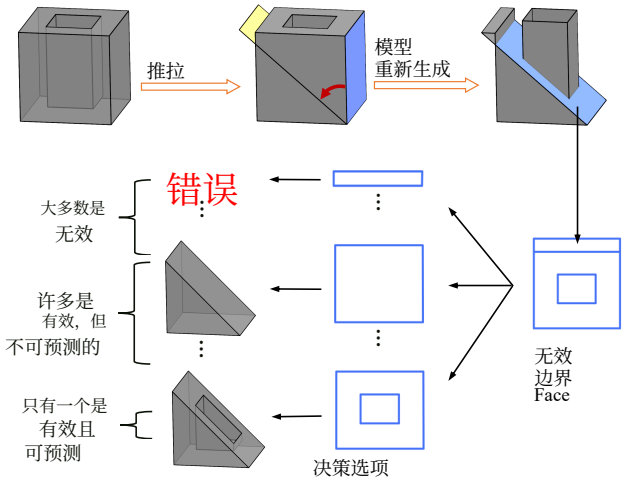


图4：不同 GTI 消解选项及其不同结果的示例。

转而采用连续性要求，即模型变化遵循连续的变化模式。这一放宽受到 Raghothama 和 Shapiro 的研究 [39]启发。他们指出，在计算机辅助设计中，连续性要求能够避免大部分（甚至全部）不可预测的建模行为。（请注意，放宽是指通过研究一个相近且更易求解的问题来处理难题。在我们的案例中，可预测性是难题，而连续性则是与之相近的问题。）总体而言，GTI 消解中决策的根本挑战，在于稳健地生成有效的建模结果，并确保模型变化连续。

SAI 消解也面临多种解决选项的情况。以图3底部的情形为例。更新后的模型几何约束系统中存在一个过约束部分：约束 5、6 和 7 之间形成了循环依赖。删除任何与该依赖无关的约束都无法消解过约束，并会导致消解失败。只有删除与该依赖相关的约束才能消解过约束，如表 1 所示。因此，稳健地生成相关约束（即有效的解决选项）是 SAI 消解面临的基本挑战之一。

即使解决方法成功避免了无效的决策选项，通常也仍会有多个有效的解决方案。表 1 中列出的选项就是这类情况的典型示例。由于同样涉及设计意图，因此很难在这些有效的解决方案之间做出决策。因此，至少目前看来，完全自动化的决策方法似乎无法实现。决策支持方案或许是更实际的选择。关键在于对有效的解决方案进行优先级排序，从而逐步向用户推荐。有效的优先级排序方案应能准确衡量移除或添加约束对模型形状的影响。然而，这个问题并不简单，因为移除或添加约束属于定性操作，而模型形状变化则是定量的，两者之间并无直接联系，

性质。

总之，参数化建模与直接建模集成的主要问题，是模型在进行直接编辑时可能出现的 GTI 和 SAI。为处理这些问题，通常存在多种解决选项，因此需要进行系统化决策。对于 GTI 消解，挑战在于稳健地生成有效的建模结果，并实现模型的连续变化。对于 SAI 消解，挑战在于稳健地生成有效的解决选项，以及有效地确定这些选项的优先级。

4. 所提出的方法：变分直接建模

基于上述问题分析，本文提出了一个参数化/直接建模集成框架，如图 5、7 和 8 所示。本节重点概述该框架的高层工作流。底层实现细节将在下一节中介绍。图 5 展示了最高层级的工作流。右侧虚线矩形中的四个模块构成了该框架的主要组成部分。该框架以模型变更信息（由用户编辑所导致）为输入，随后修正拓扑（针对 GTI）和约束（针对 SAI），以适应变更后的模型信息，最后输出一组一致的信息，用于更新正在编辑的模型。我们的方法需要从模型中提取的具体模型信息列于中间的虚线矩形中。修正过程主要分为两个步骤：检测和解决。检测步骤会检查是否存在信息不一致；如果存在，则将不一致的信息整理成便于后续解决步骤处理的形式。

*4.1. GTI*检测与消解

本工作没有将 GTI 检测与消解推迟到直接编辑结束后，而是采用迭代方法，重复执行以下两个步骤，直至直接编辑完成：（1）检测下一个发生 GTI 的点；（2）立即消解该点检测到的不一致。图6以图4中的模型为例，说明了这一方法。整个推拉操作过程中有三个发生 GTI 的临界点：第一个是蓝色面碰到孔的右侧面时；第二个是蓝色面碰到孔的左侧面时；最后一个是蓝色面碰到块的左侧面时。每当到达一个临界点，就立即消解由此形成的 GTI，并在该点生成一个中间模型（见图6下排的三个模型）。随后，将剩余的直接编辑应用于该中间模型，而不是原始模型。将本例中展示的步骤推广后，便得到图7所示的算法工作流程。图中将发生 GTI 的临界点简称为 GTIP（GTI 点）。

迭代方法的本质是将用户指定的直接编辑分解为多个依次进行的较小直接编辑。这样做的好处是，在 GTIP 处进行不一致消解会变得简单，因为有效的参考模型

表 1：图 3 [35] 中底部情况的候选解决方案。

已更新 模型 GCS	候选项 解答 1	候选项 解答 2	候选项 解答 3
1. 距离(面1,面3)=1	1. 距离(面1,面3)=1	1. 距离(面1,面3)=1	1. 距离(面1,面3)=1
2. 距离(面5,面6)=1	2. 距离(面5,面6)=1	2. 距离(面5,面6)=1	2. 距离(面5,面6)=1
3. 垂直(面1,面5)	3. 垂直(面1,面5)	3. 垂直(面1,面5)	3. 垂直(面1,面5)
4. 垂直(面1,面4)	4. 垂直(面1,面4)	4. 垂直(面1,面4)	4. 垂直(面1,面4)
5. 垂直(面4,面5)	5. 垂直(面4,面5)	5. **(已移除)**	5. 垂直(面4,面5)
6. 平行(面2,面4)	6. **(已移除)**	6. 平行(面2,面4)	6. 平行(面2,面4)
7. 垂直(面2,面5)	7. 垂直(面2,面5)	7. 垂直(面2,面5)	7. **(已移除)**
8. 长度(E1)=1	8. 长度(E1)=1	8. 长度(E1)=1	8. 长度(E1)=1

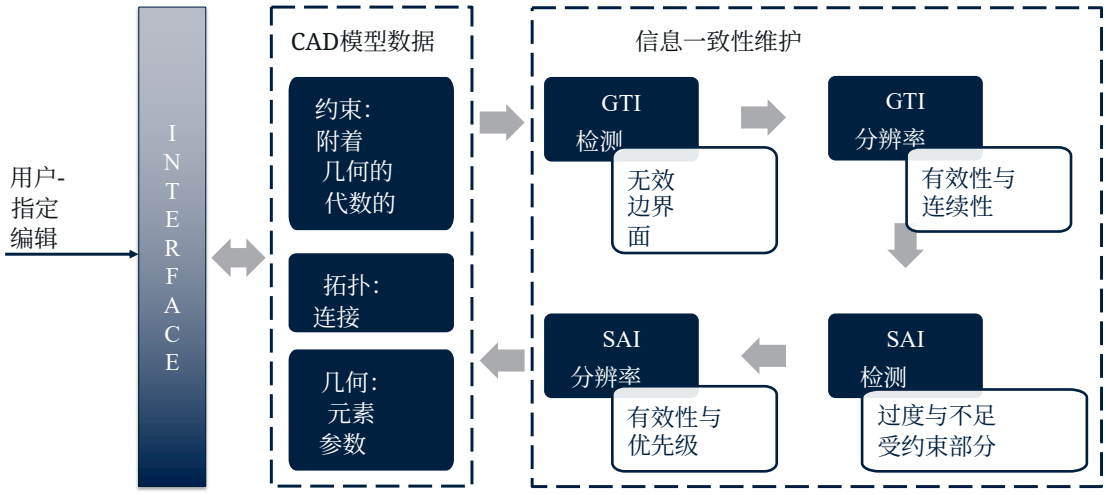


图5：变分直接建模中信息不一致性消解的总体框架。

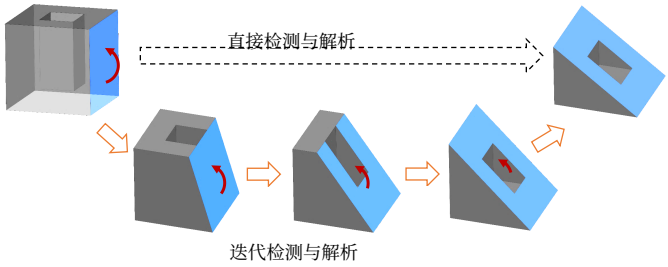


图6：GTI 检测与消解迭代过程示意图（蓝色面：推压面；红色箭头方向：旋转方向；箭头长度：旋转角度）。

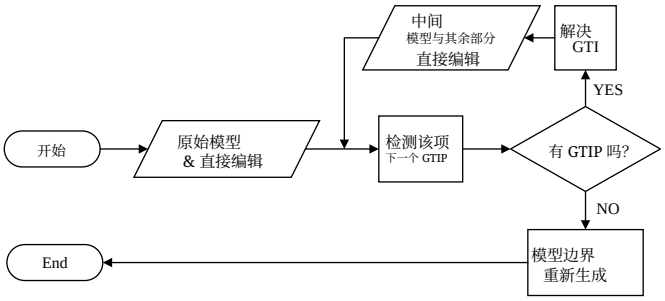


图7：GTI 检测与消解示意图。

就在附近。相比之下，直接编辑结束时的不一致消解并不具备这样的有利条件。（不过需要指出的是，尽管迭代方法适用于几乎所有直接建模操作，例如推压和面缩放，但它无法处理面删除操作，因为该操作会突然移除边界面，无法将其分解为更小的步骤。这可以视为所提出方法的一项严重局限。）

4.2. SAI 检测与解决

解决所有 GTI 后，下一项任务是执行 SAI 检测和决策模块。如前所述，SAI 表现为过度约束部分和欠约束部分。因此，SAI 检测模块的主要任务是找出这些部分，并将其作为输入传递给 SAI 决策模块，以生成决策选项并确定其优先级，从而便于决策。然而，仅仅找出这些部分并一并呈现，并不能让人了解各个不一致是如何形成的。在进入 SAI 决策模块之前，应先对它们进行解耦。解耦之所以重要，是因为与某个过度约束（或欠约束）部分的决策相关的约束，可能与其他部分无关。处理某个部分时，纳入任何无关约束只会使后续的 SAI 决策工作更加复杂。

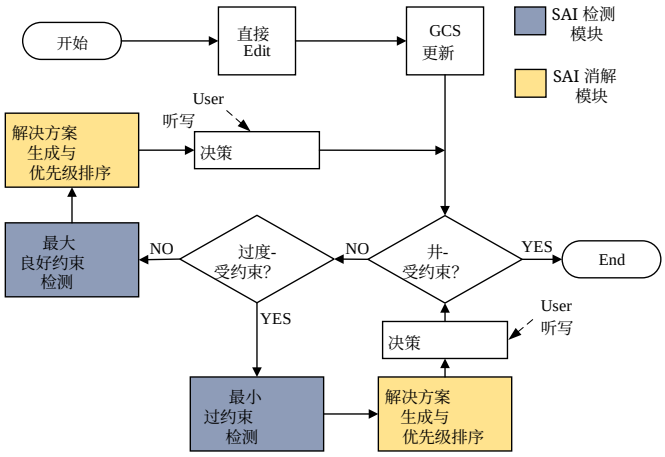


图 8：SAI 检测与解决示意图。

为实现解耦，应尽可能缩小过度约束部分的规模，并尽可能扩大约束良好部分的规模。过度约束部分本质上是一组彼此存在依赖关系的约束。若将其规模缩至最小，就无法再将其分解为更小的子部分，其中便只存在一个循环约束依赖。因此，不会纳入任何无关约束。过度约束部分中仅存在一个循环约束依赖还带来以下好处：移除任意一项约束都能打破循环依赖，如表 1 中的决策选项所示。换言之，用于消解最小过度约束部分的有效决策选项，就是该部分中的约束本身。

对于欠约束部分，可以提出类似的要求：应尽可能缩小其尺寸。这等同于最大化模型中的每个约束良好部分，因为模型的欠约束状态可以用其约束良好部分之间的自由度来表示。一旦约束良好部分达到最大，能够限制它们之间自由度（同时不为模型引入新的过约束）的约束就是可行的解决方案。在单个最大化的约束良好部分内部定义的任何约束，都是无效的解决方案。

基于上述讨论，我们概述了图8所示的 SAI 检测与解决工作流。该工作流从用户指定的直接编辑开始，然后根据新的模型形状执行 GCS 更新，再将模型发送到分析器（菱形框中的模块）以评估其约束状态。如果模型仍然约束良好，则无需采取任何措施；否则，工作流将进入两个不同的分支。在这两个分支中，首先提取信息不一致（两个检测模块），然后根据检测结果生成有效的解决方案选项并进行优先级排序（两个优先级排序模块），再将排序后的选项呈现给用户，由用户作出决定。此外，如果用户选择不手动处理这些不一致，还可以使用自动模式。在该模式下，将选择优先级列表中的首选项。

5. 变分直接建模的实现细节

在概述了该框架之后，本节将介绍可用于实现其组成模块的方法，以展示变分直接建模在实践中的可行性。上一节的讨论/公式推导将该框架的实现归纳为解决以下四个关键技术问题：

1. 下一步是检测 GTIP，以实现图 5 中的 GTI 检测模块；2. 在 GTIP 处进行 GTI 消解，以实现图 5 中的 GTI 消解模块；3. 检测最小过约束和最大良好约束，以实现图 5 中的 SAI 检测模块；4. 进行约束优先级排序，以实现图 5 中的 SAI 解决模块。

接下来，我们将介绍当前可用的算法如何解决这些关键问题、能够解决到何种程度、可以提出哪些新的改进以补充现有方法，以及哪些部分仍然存在问题。基本而言，利用现有方法并进行一些细微改进，就可以实现一个初步原型，用于展示该框架的可行性；但若要使该框架实现完整功能，仍需进一步开发，尤其是针对第四个技术问题。我们希望这些开放领域能够吸引本领域更多的研究工作，最终实现参数化建模与直接建模的无缝集成。还需要指出的是，本文的重点和贡献主要在于参数化建模/直接建模集成的理论基础，以及变分直接建模框架，正如第 3 节和第 4 节所述。用于解决上述问题的算法仅代表可行的方法，并不一定意味着它们是最佳或唯一的方案。

5.1. 下一步 *GTIP* 检测

由于直接建模在计算机辅助设计领域还是一个相对较新的概念，与 GTIP 检测相关的研究工作并不多。现有方法 [12, 40–43] 一贯采用启发式策略，检查给定的建模操作是否会导致 GTI，以及 GTI 何时发生。其基本思路是：首先将 GTIP 与边界面的退化构型联系起来（例如，两个原本相交的平面变为平行），然后检查所生成的退化构型是否确实可能发生。

采用上述策略，Lipp 等人 [40] 设计了一套完整的启发式方法，用于预测推拉操作期间何时会出现 GTI，但仅适用于由平面且无孔边界面构成的多边形网格模型。遗憾的是，这类模型只占实体模型的一小部分。在我们先前的工作 [12], 中，我们推导出了另一套启发式方法，将 Lipp 的工作扩展到由线性、二次曲面及带孔边界面构成的模型（但不涉及自由曲面）。然而，所提出的启发式方法仍仅限于由相邻面之间形成的局部几何与拓扑不变量。对于全局 GTI，例如一个边界面

穿入另一个边界面，两种方法 [12, 40]均存在局限。

在参数化建模领域，也有一些与 GTIP 检测相关的研究 [41, 42]。这些研究旨在计算模型在参数化编辑过程中拓扑发生变化的临界点。这与本文考虑的问题略有不同，但这些临界点的概念在理论上相似。 [41, 42]中介绍的方法也运用启发式方法，根据可能出现的退化面构型，预测参数化修改过程中的拓扑变化点。不过，在检查退化面构型之前，这些方法会先识别受建模操作影响的边界面，以缩短检查时间（称为剔除思想）。

在本研究中，我们结合了 [12] 中提出的启发式方法和 [41, 42] 中介绍的思想，实现了 GTIP 检测模块。我们并未使用剔除思想来加速 GTIP 检测，而是用它来处理全局 GTI。具体而言，在应用 [12], 中提出的启发式方法之前，我们先进行边界再生成，以确定模型中哪些边界面变为无效。这类边界面即为直接编辑过程中受影响的面。显然，由此得到的受影响面集合包含了直接编辑过程中与全局 GTI（如果存在）相关的边界面。

确定受影响的边界面后，我们直接对其应用 [12] 中的启发式方法，生成一份完整的退化构型列表；随后检查这些候选项是否可能在特定的直接编辑中实际发生，从而缩小候选项集合。最后，我们选取剩余候选项中距离最近的一项，作为所需的下一个 GTIP。上述步骤已汇总于算法 1 中。该算法不会遗漏任何 GTIP，因为 [12] 中的启发式方法是以系统化的方式构建的。推导过程简述如下。正如第 3 节中所述，GTI 有两种来源：插入新连接和丢失旧连接。这两种来源与曲面类型的所有可能组合，共同产生 GTI 可能出现的三种方式；这些方式总共可转化为以下四种退化构型：曲面-曲面相交、曲面-面碰撞、曲面-曲面相切，以及面-工作空间碰撞，如图 9 所示。这里，我们区分了简称边界面的面，以及作为基础的曲面.....

界面。

这里需要指出一条实践经验。针对所有一般退化构型进行穷举检查，对于一般情况而言非常耗时，但在直接建模的语境下，这是可以接受的，因为直接编辑属于局部修改，而检查中涉及的边界面通常非常少。

5.2. GTIP 处的 GTI 消解

GTI 消解旨在修改 GTIP 前的拓扑，以适应 GTIP 后的几何，同时确保建模结果有效且模型变化连续。这项任务同样鲜有研究成果[12, 40, 43]。Lipp 等人 [40]提出通过一组启发式方法直接修改模型拓扑，以消解检测到的所有 GTI。尽管这些启发式方法在多边形

算法 1 下一处 GTIP 检测

输入：M, E – 模型和所应用的直接编辑
输出：t – 下一个 GTIP

$F \leftarrow \emptyset$ // 用于存储受影响的面
$\mathcal{C} \leftarrow \emptyset$ // 用于存储退化构型
\ $M \leftarrow$ 执行边界重新生成(M, E)
$F \leftarrow$获取边界不良的面(M)
对每个 $f_1, f_2 \in F$ 执行 // 使用启发式方法
    $C \leftarrow$ 获取退化配置(f_1, f_2)
结束循环
对每个 $c \in C$ 执行
    如果 是否发生(c) =为 TRUE 则
        $t \leftarrow$ 最小值$(t,$ 获取发生时间$(c))$
    e 结束 if
结束 for
返回 t

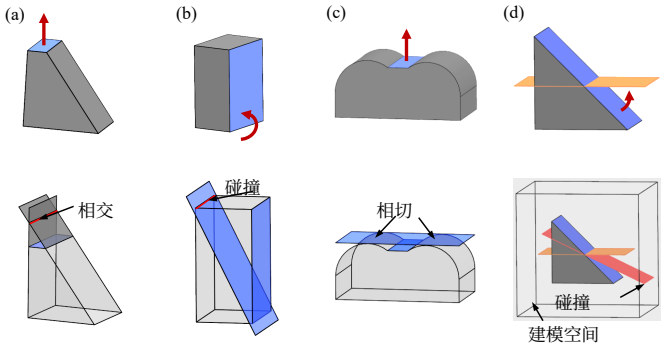


图 9：GTIP 检测中退化构型的示意图（蓝色面：推拉面；弯曲箭头：旋转推拉；直线箭头：平移推拉） [12]。

网格模型上的鲁棒性已得到验证，但如何将其扩展到由线性与二次曲面构成的模型，仍然未知。与他们直接修改拓扑的方法不同，我们先前的工作 [12, 43] 提出了一种间接方法，将此处的任务转化为对模型体积执行布尔运算。图 10 展示了这一转换过程，其依据是图 6 中的第一个 GTIP。通过从原始模型中减去楔形模型，即可直接得到该 GTIP 所需的中间模型。与直接修改拓扑相比，采用布尔运算的一个重要优势是能够自动保证生成的模型有效 [4]。此外，无需深入处理模型底层的拓扑与几何数据，否则容易出错。本工作采用了这一方法来实现 GTI 消解模块。

使用基于布尔运算的方法也很容易实现模型的连续变化。我们需要做的是，在构造要从原始模型中减去或添加到原始模型中的体积时加入连续性约束。这些体积称为辅助模型。具体的连续性约束是[12]：将辅助模型限制在编辑后的表面从当前点扫掠至下一个 GTIP 所经过的体积范围内，

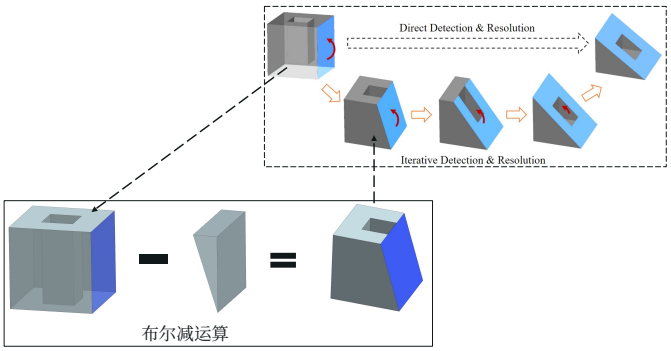


图 10：基于布尔运算的 GTI 消解示意图（直立的子图与图 6 相同）。

并由其相邻表面围成。为此，我们开发了一种四步构造算法，如图 11 所示，概述如下：

- 第一步是选取相关的相邻边界面，并将其延伸至覆盖直接编辑的整个扫掠范围。
- 第二步和第三步协同作用，求出由第一步生成的相邻边界面与直接编辑的边界面所围成的体积。以此方式生成的体积有两种可能的类型：如果直接编辑向模型添加材料，则该体积为加法式；如果移除材料，则该体积为减法式，如图 11 所示。
- 最后一步对原始模型与前几步生成的辅助模型执行实际的布尔运算。

需要指出的是，上述步骤仅是对该算法的简要介绍。显然，该算法涉及的技术细节和特殊情况远多于图 11 所示的简单案例。有关如何处理这些问题的更多细节，请参阅 [12, 43]，以获取完整版本的算法。

5.3. 最小过约束与最大约束良好部分检测

此处的检测任务包括评估模型的约束状态；若模型未达到约束良好状态，还需提取模型中的最小过约束部分和最大充分约束部分。这是几何约束求解领域中得到广泛研究的课题，详尽综述见 [26, 44]。所提出的方法可分为以下四类。

基于求解的方法 这类方法通过数值方法（如同伦延拓）或符号方法（如格罗布纳基）直接求解模型，以分析模型的约束状态 [26]。由于耗时很长，这类方法在当今的计算机辅助设计系统中很少使用。

基于逻辑 这类方法将公理化的概念应用于 GCS。它们的工作方式依赖于一组几何定理和推导规则 [45]。如果一个模型 GCS 可以从这些定理和规则中通过逻辑推导得出，则其约束良好；如果存在额外约束，则约束过度；否则，则约束不足。尽管在数学上优雅，但这类方法面临一个问题：必须拥有一套完备的几何定理和推导规则，才能涵盖常用的几何约束。

基于图的方法 这类方法在文献中最为常见。其基本思路是先将模型 GCS 转换为图，再分析该图以获取约束状态信息，而不是使用原始 GCS。基于这一思路，形成了两条发展路线。第一条路线试图从给定的约束图中识别与已知形状相对应的子图模式。该路线由 Owen [46]开创，并在模式库规模方面于 [47–49] 得到显著改进。第二条路线则比较模型几何的自由度与模型 GCS 的限制度。该方法最初由 Bardord [50] 和 Serrano [51], 提出，并在 [52–54]中作了详细阐述。2001 年，这两条路线被统一到 Hoffmann 等人提出的框架中 [32]。此后，相关研究仍取得了一些进展，例如 [44], 中讨论的工作，但其基础始终未变。基于图的方法已在工业界得到广泛应用。然而，这类方法无法处理存在约束依赖关系的 GCS（最简单的结构性依赖关系除外） [55, 56]。这是因为模型 GCS 一旦转换为图，保留下来的就只有组合信息，而所有几何信息都会丢失。这一局限使基于图的方法不适用于本文所研究的 SAI 检测问题。（需要指出的是，基于图的方法相关论文数量庞大，受篇幅限制，上述讨论仅涵盖部分重要研究。）

基于扰动的方法 为克服基于图的方法的局限性，Michelucci 等人 [56]提出了见证配置法（WCM）。该方法研究约束方程变量受到无穷小扰动时约束方程的行为，并通过约束方程对应的雅可比矩阵描述这种行为。不同的约束状态具有不同的扰动行为 [57]。尽管该方法在处理约束依赖关系方面效果显著，但由于采用了贪心算法，所提出的算法难以实现最小过约束和最大充分约束。为此，我们在先前的工作中 [35, 58]提出了一系列基于 WCM 的算法，用于在模型 GCS 中自动提取最小过约束部分和最大充分约束部分。

考虑到 SAI 通常存在约束依赖关系，在实现 SAI 检测模块时，我们采用 WCM 来完成约束状态评估任务。对于最小过约束和最大充分约束部分的提取任务，则采用了 [35, 58] 中开发的算法。下面简要介绍所采用的方法。

使用 WCM，模型的约束状态可通过其约束方程的雅可比矩阵 [56] 来表征。如果雅可比矩阵存在线性相关行，则 GCS 处于过度约束状态。雅可比矩阵的零空间表明模型的自由度；其中

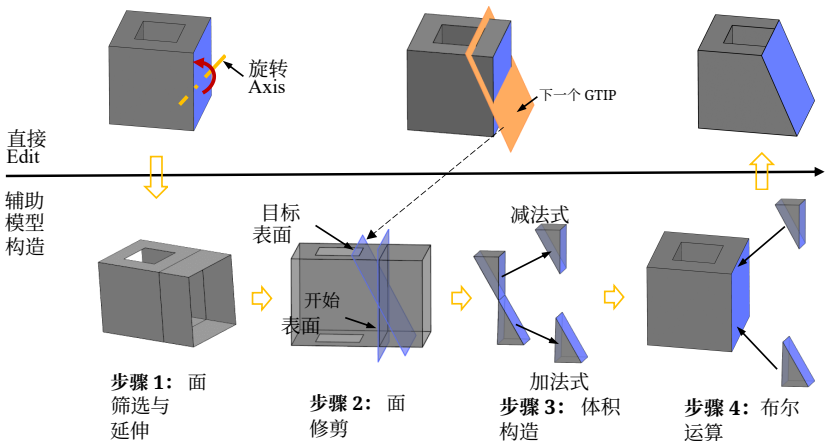


图 11：辅助模型构造所用流程的示意图。

如果模型的自由度多于六个典型刚体变换，则模型处于欠约束状态。[57, 58]。根据线性代数，我们知道，雅可比矩阵 J 中的相关行可通过线性方程组 $J^T \cdot x = 0$的解体现。零空间 J^T中的向量 x 代表一个过约束部分，而 x 中的非零元素则表明该部分涉及的约束。因此，最小化该部分的规模等价于要求向量 x的非零元素数量最少。接着，我们可以通过求解以下优化问题，得到规模最小的过约束部分：

$$\min_x \|x\|_0 \quad \text{s.t.} \quad J^T x = 0,\ x \neq 0$$

其中，$\|\cdot\|_0$ 是 ℓ_0 范数，其数学含义是统计向量中的非零元素个数。因此，最小过约束检测问题被转化为稀疏恢复（又称压缩感知）问题。这是图像处理领域中一个经过充分研究的问题 [59]，可以使用[60]中所述的松弛方法进行数值求解。通过一系列数学运算，最大良约束检测问题也可以转化为类似的稀疏恢复问题。详细推导请参阅 [35]的第 4.3 节。

*5.4. SAI*解决方案优先级排序

根据图8所概述的工作流，在获得最小的过度约束部分和最大的约束良好部分之后，下一步是生成解决方案，然后确定其优先级。在这些部分已经被最小化/最大化的情况下，生成有效的解决方案并不困难（如第4.2节所述）。因此，剩下的关键任务就是确定所生成解决方案的优先级。

解决方案采用几何约束的形式，这些约束会从模型中移除或添加到模型中。对它们进行优先级排序，就是将其排列成一定顺序；该顺序由一种二元比较运算决定，此运算接受两个约束作为参数，并判断哪一个应排在前面。与几何约束求解这一主题不同，针对几何约束优先级排序的研究工作要少得多。现有方法大致可分为定性方法和定量方法。定性方法根据一组关于约束类型的启发式方法来确定约束的优先级 [61–64]。在某些应用场景中，基于类型的优先级排序可能很有效，因为某些类型的约束承载着更多工程知识，且比其他类型更常见。然而，这些方法无法进一步区分相同类型几何约束的优先级。定量方法则可以解决这一问题。文献中提出了一种基于偏差的度量，用于确定几何约束的优先级 [65, 66]。本研究采用了类似思路来实现 SAI 决策模块。

在本研究中，我们结合定性和定量策略来完成约束优先级排序任务。这形成了一个由粗略比较和精细比较组成的两级比较方案。粗略比较负责保留类型可能承载更多设计意图的约束。例如，平行关系可能比一般的角度约束承载更多设计意图。本研究采用的具体类型优先级见表2，其依据是来自 [64, 65]的实验数据。

精细比较处理类型相同的几何约束。我们基于敏感性的概念对它们进行比较。对不同约束进行参数变化通常会导致不同的模型形状变化，这可以用敏感性的概念进行数学刻画（即模型形状变化相对于参数变化的变化率）。如果某个约束具有较高的敏感性值，那么对其进行微小的参数变化就会导致模型形状发生较大变化，从而可能引发剧烈且不可预测的形状变化。在这种情况下，模型被称为具有较差的约束方案 [67]。因此，我们使用敏感性来量化约束对模型形状的影响。采用这一选择后，可以预期将选择使模型具有最小变化率的解决方案，从而在后续参数化编辑中实现最小的模型变化。

需要指出的是，上述约束优先级排序方法仍采用启发式方法。因此，它可能存在

表2：基于约束类型的粗略优先级排序。

几何约束 实体	Type	优先级（1：高，5：低）
面 面	平行/垂直方向，位置之间的距离，相等的尺寸参数	1
	方向之间的一般夹角	2
Face Edge	平行/垂直方向，位置之间的距离	2
	方向之间的一般夹角	4
面-顶点距离（位置之间）		5
边-边	相等的角度/长度参数，平行/垂直方向，位置之间的距离	3
	方向之间的一般夹角	5
边-顶点距离的位置之间的距离		5
顶点间距离的位置之间的距离		5

泛化问题。目前，如图8所示，我们通过纳入用户指导（如有必要）来缓解这一问题。然而，提示用户选择合适的结果可能会影响其工作流。因此，我们非常需要一种能够以自动且智能的方式运行的约束优先级排序方法。这正是当前最先进水平仍显不足、亟待进一步发展的领域。

6. 建模示例

6.1. 原型实现

为展示所提出框架的有效性，我们使用 C++ 在 Apple Macintosh 环境（2.4 GHz Intel Core i5，8G 内存）下开发了一个初步原型建模器。该软件的架构与开源几何处理和渲染框架 Open-Flipper（3.0 版）类似。建模器的图形用户界面如图 12 所示，实现了图 5 中概述的框架接口模块。该界面使用 QT 库（5.7 版）实现。框架的其他四个模块，即 GTI 检测、GTI 消解、SAI 检测和 SAI 消解，均基于几何建模内核 Open CASCADE（7.0 版）开发。所有数值求解/优化任务均使用 C++ 库 Eigen（3.2.9 版）和 MATLAB 完成。

要编辑模型，用户点击直接建模工具箱中的按钮（红框 1），随后弹出图形模型操纵手柄（红框 2），供用户进行交互式修改。要查看模型的几何约束，用户点击左侧边栏中的约束处理按钮，随后弹出图 12 所示的中间面板，其中红框 3 列出了所有约束。要查看模型中是否存在 SAI，用户点击红框 4 中的“分析/消解”按钮，随后计算机显示消解这些 SAI 所需的全部信息（红框 5 和 6），包括自由度、约束依赖关系，以及消解建议。如果用户希望由计算机完成所有工作，则应勾选红框 4 中的两个“自动”选项。

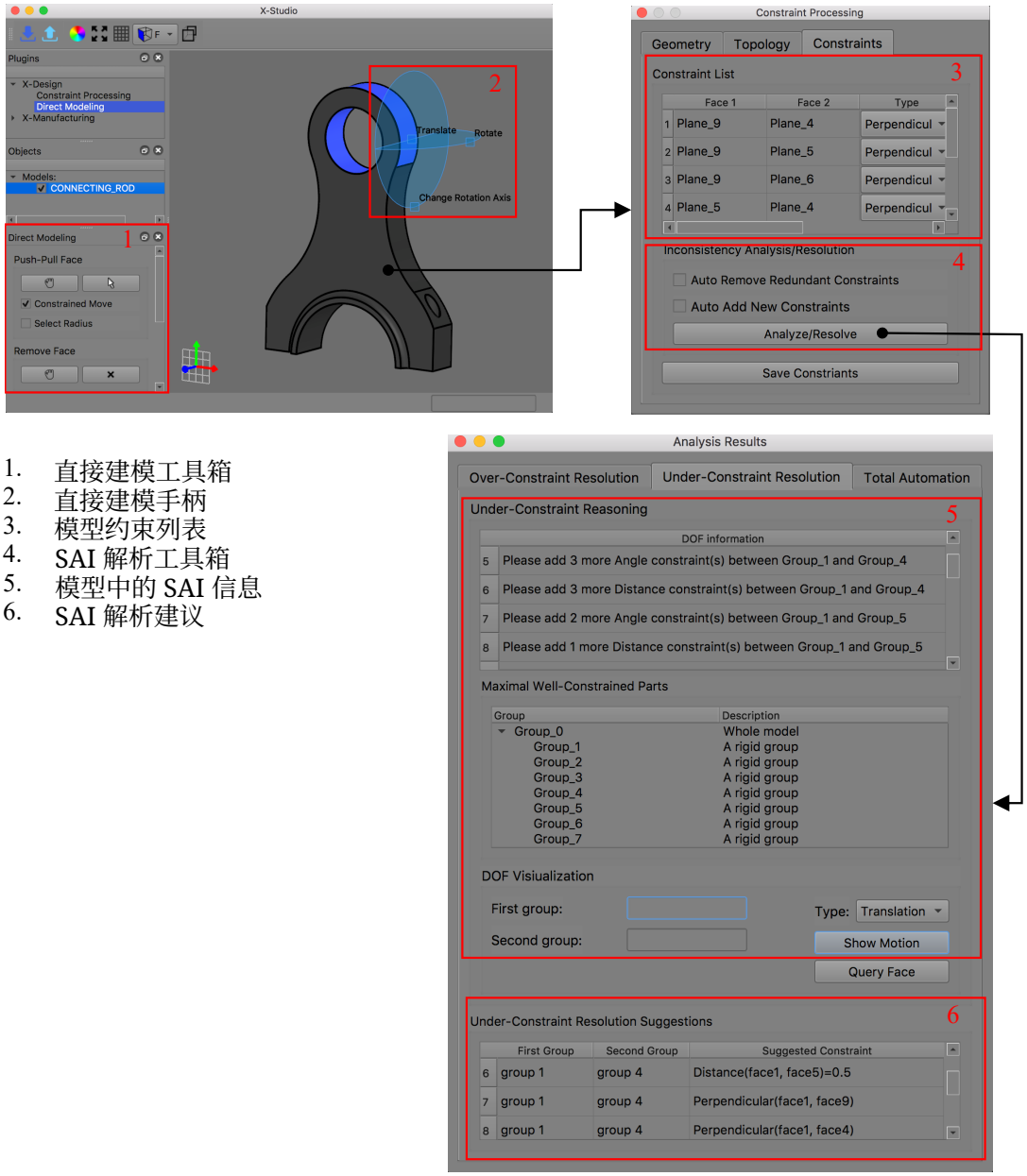
6.2. 示例与比较

基于原型建模器，我们使用从真实数据和模拟数据中选取的示例，对所提出的检测与消解机制（即图 7 和图 8 中的模块）的实用性进行了测试。图 13 展示了部分测试示例。蓝色面表示编辑后的面，红色箭头表示推拉方向，红色虚线表示旋转轴。例如，在示例 1a 中，蓝色面逆时针旋转了 160 度。表 3 根据图 13 所示示例，总结了与领先 CAD 软件的比较结果。图 14 还展示了 Siemens NX 和 Ansys SpaceClaim 的一些失败案例。通常，NX 会将相关面标成红色，以提示模型更新失败；SpaceClaim 则会将相关面标成橙色（如中间的示例），或生成随机但错误的形状（如左侧的示例）。

此外，我们还在一组规模更大的数据集上，对 Siemens NX、Ansys SpaceClaim 和我们的方法进行了更定量的比较。该数据集包含 20 个 CAD模型，其中包括图 13 中已展示的模型，以及图 15 所示的另外 12 个 CAD模型。测试时，对每个测试模型随机执行 10 次直接编辑，然后测量各方法的成功率。对于这 $200 = 20 \times 10$ 次随机直接编辑，其中一半特意设置为不会导致被编辑模型发生任何拓扑变化。最终成功率如下：对于不会导致拓扑变化的 100 次直接编辑，三种方法/工具的成功率非常接近：Siemens NX（94%）、Ansys SpaceClaim（96%）和我们的方法（96%）。对于其余涉及拓扑变化的 100 次直接编辑，成功率则存在明显差异：Siemens NX（76%）、Ansys SpaceClaim（84%）和我们的方法（93%）。这证实了所提出的方法在处理信息不一致方面具有较强的鲁棒性。

图 16 展示了一个示例，其中所有模块（即前文介绍的 GTI 检测和 GTI 消解模块）被组装在一起协同工作。该示例涉及一个齿轮箱安装零件模型（取自 GrabCAD 零件库 https://grabcad.com/library）。所应用的直接编辑相当全面，包括单面编辑、多面编辑、平移编辑类型和旋转编辑类型。在每个子图中，蓝色文字和箭头标示了后续直接编辑要修改的面，红色椭圆圈出的部分则展示了编辑结果。从图 13、14 和 16 中的所有示例与比较可以看出，所提出的方法显著提高了直接建模的鲁棒性。

图17、18和19展示了一个整合 GTI 检测、GTI 消解、SAI 检测和 SAI 消解这四个模块的示例。该示例同样基于从 GrabCAD 库下载的模型。图17a展示了旋转顶面进行直接编辑后更新的模型形状。图17b列出了根据新模型形状得到的更新后的模型几何约束系统。图18展示了在更新后的模型几何约束系统中检测到的最大充分约束部分。总共找到了7个部分。对于每个部分，



1. 直接建模工具箱
2. 直接建模手柄
3. 模型约束列表
4. SAI 解析工具箱
5. 模型中的 SAI 信息
6. SAI 解析建议

图 12：原型建模器的图形用户界面。

其所属边界面以深色显示，不属于该部分的面则设为透明。图的右下方还列出了每个部分的面列表。图19a—19d展示了重新消解所有 GTI 和 SAI 后，参数化编辑下的建模行为。如图所示，两个顶部特征之间的对称设计意图得以成功保持，符合预期。对一个特征所做的更改也会反映到另一个特征上。

7. 结论与未来工作

现代 CAD系统有两种主要的建模方式：参数化建模和直接建模。参数化/直接建模集成是一种结合两者互补优势的方式，正成为计算机辅助设计的重要发展方向。本文详细回顾了学术界和工业界公开报道的该方向相关方法。回顾结果表明，这些方法仍处于早期发展阶段，而参数化/直接建模的无缝集成仍是一个尚未解决的问题。随后，本文针对这一问题提出了一种替代方案，称为变分直接建模。首先，通过问题分析确定了参数化/直接建模集成中的根本问题与挑战。随后，提出了一个框架来应对这些挑战。最后，给出了实现该框架各组成模块的有效算法。

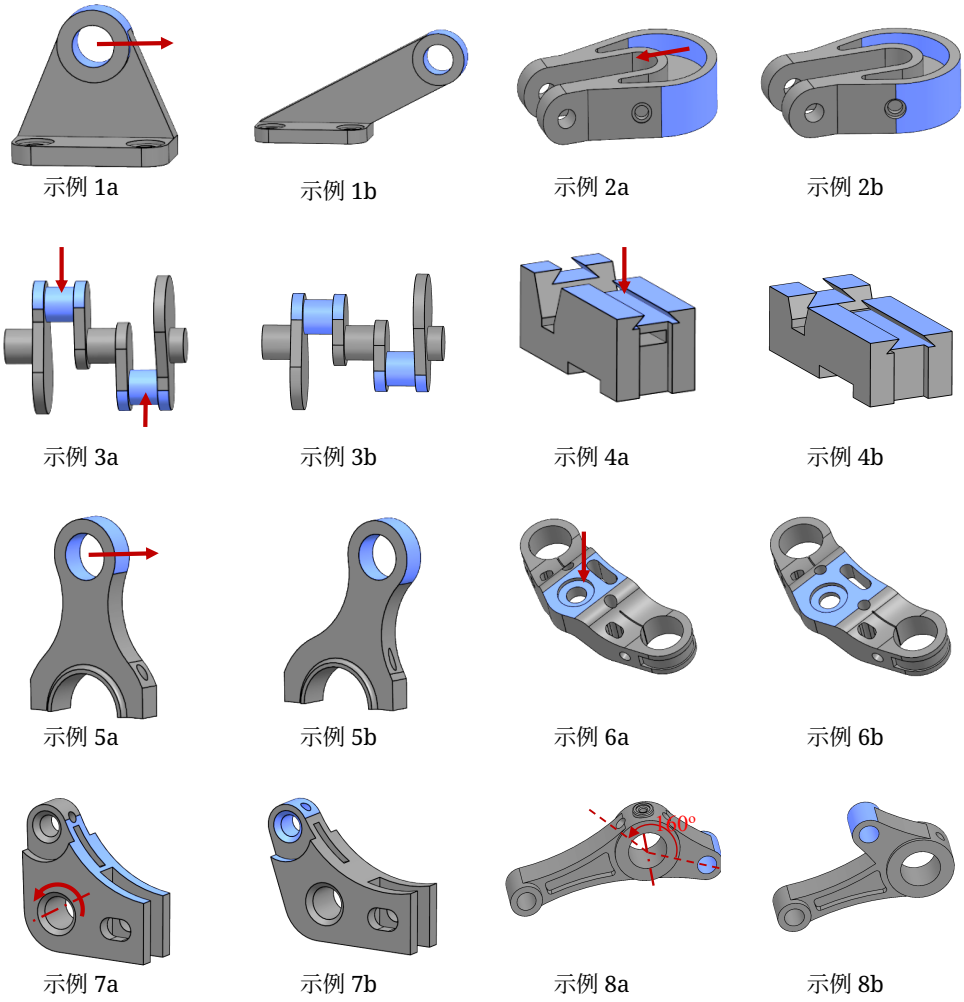


图 13：GTI检测与消解示例（a：原始模型及应用的直接编辑；b：生成的模型）。

表 3：与领先 CAD 软件及其失败/成功统计的比较。

	测试模型（编号请参见图 13）								总体成功率
	1	2	3	4	5	6	7	8	
Siemens NX	✓	✓	×	×	×	×	✓	×	3/8
SpaceClaim	✓	×	×	×	×	×	✓	×	2/8
PTC Creo	✓	✓	×	×	×	×	×	×	2/8
Autodesk Inventor	✓	✓	×	×	×	×	×	×	2/8
SolidWorks	✓	×	✓	×	×	×	×	×	2/8
所提出的方法	✓	✓	✓	✓	✓	✓	✓	✓	8/8

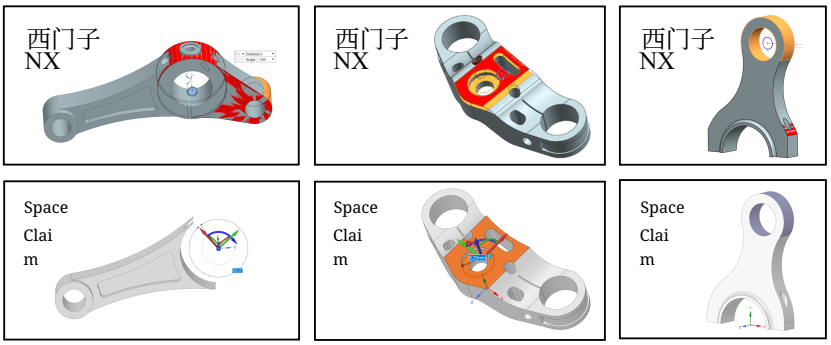


图14：两个最先进的直接建模软件 Siemens NX 和 SpaceClaim 针对图 13 中示例给出的失败案例。

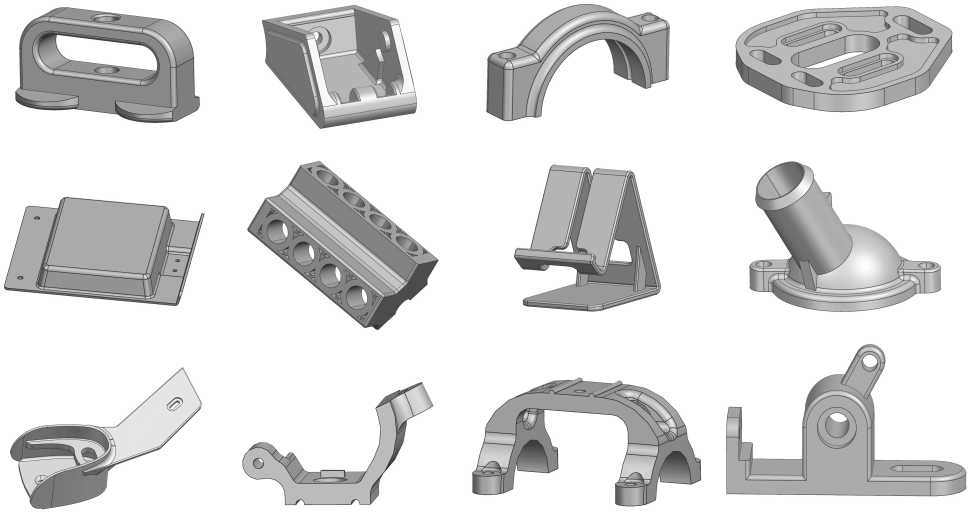


图15：与图 13 中的模型一起，用于对 Siemens NX、Ansys SpaceClaim 和我们的方法进行定量比较实验。

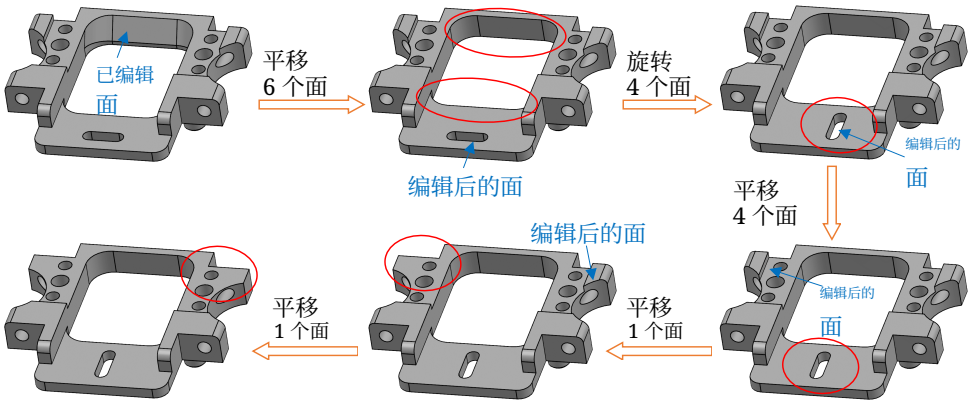
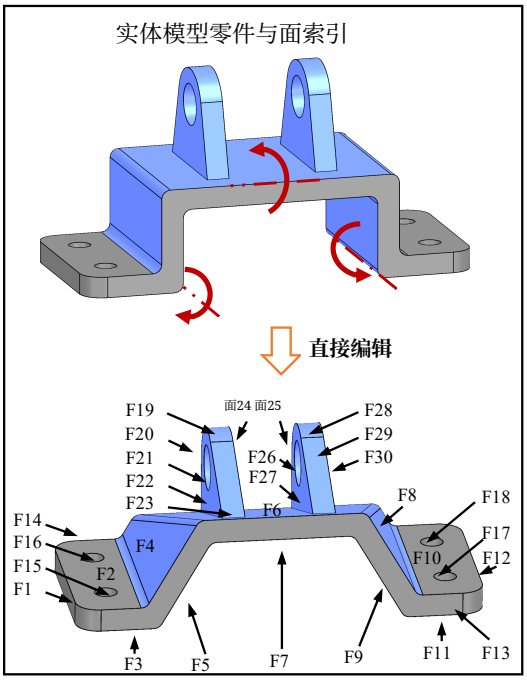


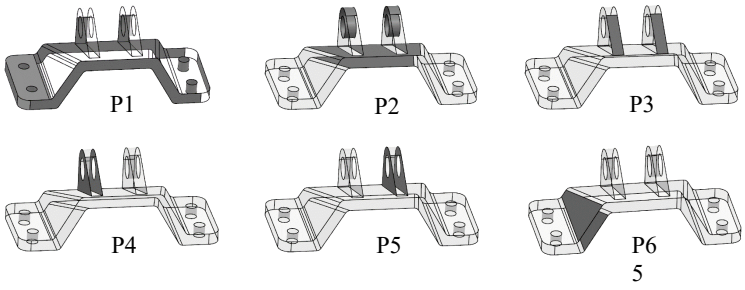
图16：一系列　对齿轮箱支架模型进行的直接编辑（圆圈标示出已　　更改的零件） [12]。



模型关联性零件与约束更新

更新后的模型几何约束系统	
C1. Dis(面2,面3)=10	C21. Ang(面23,F13)=60°
C2. Dis(面4,面5)=10	C22. Ang(面4,面22) =30°
C3. Dis(面6,面7)=10	C23. Dis(面22,面24)=10
C4. Dis(面8,面9)=10	C24. Dis(F28,面6)=10.62
C5. Dis(面10,面11)=10	C25. Dis(面28,面13)=57.96
C6. Ang(面1,面4)=30°	C26. 同轴（面28，面26）
C7. Ang(面1,面8)=150°	C27. 相切（面28，面29）
C8. Dis(面1,F12)=180	C28. 相切（面28，面25）
C9. Dis(面13,面14)=80	C29. 平行（面25，面14）
C10. Ang(面3 ,面6)=160°	C30. 角度（面29，面13）=60°
C11. Ang(面6,面11)=160°	C31. 角度（面8，面30）=30°
C12. Per(面1,面3)	C32. Dis(面27,面30)=10
C13. Per(面1,面2)	C33. Dis(F15,面1)=10
C14. Per(面2,面3)	C34. Dis(F15,F13)=20
C15. Dis(面19,面6)=10.62	C35. Dis(F16,面1)=10
C16. Dis(面19,F13)=57.96	C36. Dis(面1 6,面1 4)=20
C17. Coaxial(面19,面21)	C37. Dis(F17,F12)=10
C18. Tan(面19,面20)	C38. Dis(F17,F13)=20
C19. Tan(面19,面23)	C39. Dis(F18,F12)=10
C20. Par(面20,F14)	C40. Dis(F18,F14)=20

图17：左：发动机支架模型及对其应用的直接编辑；右：更新后的模型 GCS（Dis：距离，Ang：角，Per：垂直，Par：平行，Tan：相切）。



P1={面1,面2,面3,面12-面18,面20,面25}
P2={面6,面7,面19,面21,面26,面28}
P3={面23,面29} P4={面22,面24}
P5={面27,面30} P6={面4,面5}
P7={面8,面9} P8={面10,面11}

图18：检测到的发动机支架模型的最大充分约束部分（深色面：各部分所属的面；透明面：不属于这些部分的面；面列表：所属面的索引，如图17所示）

.

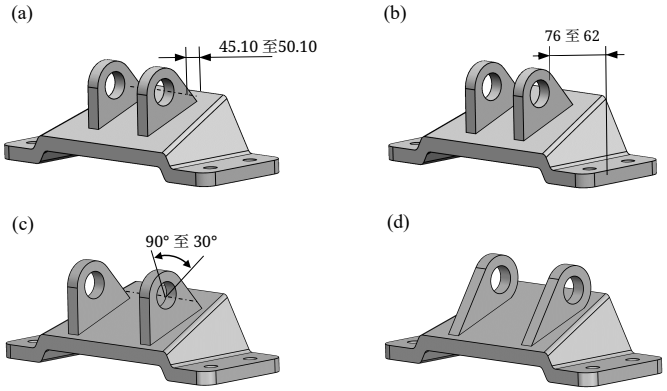


图19：分辨率处理后对模型进行的一系列参数化编辑（(a) 中编辑的建模结果如 (b) 所示，(b)、(c) 和 (d) 中的编辑结果也依此类推）。

简而言之，参数化/直接建模集成的根本问题，是处理模型在经历参数化/直接编辑时，几何、拓扑和约束信息之间的不一致。我们识别出两类关键的信息不一致：GTI 和 SAI。GTI 和 SAI 的主要问题在于，通常存在许多解决方案。根本挑战在于，如何在这些方案中进行系统性决策，从而满足以下三项要求：模型有效性（为实体且约束良好）、连续形状变化以及约束变更最少。所提出的算法之所以可行，是因为它们将这些要求转化为定义明确的实体建模操作（例如布尔运算）、已知的优化问题（例如稀疏恢复）以及有用的工程概念（例如敏感性）。

所提出的方法与现有方法的主要区别在于，它直接处理模型中的信息不一致，而现有方法则试图通过将直接编辑转换为某些特征操作来间接处理这些不一致，这会限制参数化建模能力或直接建模能力。由于采用了直接策略，所提出的方法有望实现参数化建模与直接建模的无缝集成。不过需要指出的是，这并不意味着该方法以目前的形式已经能够提供完整的解决方案。无缝参数化/直接建模集成是一个极具挑战性的问题，需要在鲁棒实体建模、快速几何约束求解、鲁棒拓扑命名以及智能约束识别等方面取得重大进展。本研究只是朝着无缝参数化/直接建模集成迈出的又一步，仍有大量工作有待完成。明确集成所面临的根本挑战，并为其提供可行的框架，是本研究的主要贡献。通过识别这些挑战，作者希望能够吸引更多研究人员投身于参数化/直接建模集成这一十分重要的研究课题。

这项工作有几个重要的改进方向，包括：

- 从实际应用的角度来看，向用户展示 SAI 解决方案建议时，应提供更直观、更具交互性的工具。要求普通CAD用户理解过约束和欠约束部分的数学复杂性及细微之处是不合理的，尤其是在零件尺寸较大时。

- 可利用人工智能算法改进 SAI 消解中选项的优先级排序方法，从而提供更智能的计算机辅助，并减少 SAI 消解过程中对人工干预的需求。当前的方法在提示用户协助解决不一致问题时，可能会中断设计过程，进而影响设计人员的工作流程。如果通过智能决策加以改进，使自动消解能够取得令人满意的效果，至少大幅减少对用户协助的需求，就能显著缓解中断问题。不过需要指出的是，在计算机辅助设计领域，完全自动化的 SAI 消解方法似乎并不可行，因为工程设计中存在主观因素。

- 另一项值得探索的改进是提高 GTI 消解的效率。目前，GTI 消解通过布尔运算实现，而这类运算的计算开销很大。处理复杂模型时，计算负荷可能会很高。平行计算和局部布尔运算（例如 Rossignac 和 Voelcker [68] 提出的方法）或许有助于解决这一问题。

- 在变分直接建模中保持特征语义也是一个非常重要的改进方向。特别是，可以利用相关信息智能地辅助用户，判断正在进行的建模操作是否会使模型的特征语义失效，以及何时会失效，例如，盲孔无法改为通孔。如果用户确实破坏了这些语义，那么这一方向的下一步工作就是自动保持特征的语义；这是一个与语义特征建模中所探讨的问题类似的问题 [20]。因此，那里提出的方法非常有帮助。

- 此外，将自由曲面纳入当前框架也非常值得关注。这不仅会带来新的挑战性问题（例如保持几何连续性）和研究机遇，还会提升这项工作的实际可用性，因为许多机械零件至少部分由自由曲面构成。将本研究扩展到处理其他操作（例如扫掠）也非常值得关注。这可以显著提高所提出框架的适用性和实用价值。

由此开发的参数化/直接集成方法，进一步提升了操作实体模型的方式，使其更加直观、智能。直接建模使设计人员能够直观地与实体模型交互，而参数化建模则负责将设计意图嵌入实体模型中

这两项优势推动实体建模进一步迈向设计早期的概念阶段。

致谢

本研究得到中国国家自然科学基金（编号：62102355）、浙江省自然科学基金（编号：LQ22F020012）、浙江省重点研发计划（编号：2022C01025）以及英属哥伦比亚大学博士奖学金的资助。

参考文献

[1] L. Li, Y. Zheng, M. Yang, J. Leng, Z. Cheng, Y. Xie, P. Jiang, Y. Ma, 特征建模方法综述：历史演变与最新发展，《机器人与计算机集成制造》61 (2020) 101851。[2] N. S. Sapidis，空间约束的几何建模：目标、方法与实体建模要求，Computing 79 (2) (2007) 337–352。[3] C. M. Hoffmann，基于约束的计算机辅助设计，技术报告，普渡大学（2005）。[4] A. A. Requicha, H. B. Voelcker，实体建模中的布尔运算：边界求值与合并算法，《IEEE学报》73 (1) (1985) 30–44。[5] A. A. Requicha, H. B. Voelcker，实体建模：历史回顾与当代评述，《IEEE计算机图形学与应用》2 (02) (1982) 9–24。[6] A. A. Requicha, H. B. Voelcker，实体建模：当前状况与研究方向，《IEEE计算机图形学与应用》3 (7) (1983) 25–37。[7] J. D. Camba, M. Contero, P. Company，参数化CAD建模：设计可重用性策略分析，《计算机辅助设计》74 (2016) 18–31。[8] C. Gonz´alez-Lluch, P. Company, M. Contero, J. D. Camba, R. Plumed，三维CAD模型质量保证与测试工具综述，《计算机辅助设计》83 (2017) 64–79。[9] J. J. Shah，参数化CAD设计：构造技术的分类与比较，载于：国际几何建模研讨会，施普林格，1998年，第53–68页。[10] M. A. El Hani, L. Rivest, R. Maranzana，产品开发中的产品数据复用：从业者视角，载于：IFIP产品生命周期管理国际会议，施普林格，2012年，第243–256页。[11] J. Monedero，参数化设计：综述与若干实践经验，Automation in Construction 9 (4) (2000) 369–377。[12] Q. Zou, H.-Y. Feng，实体CAD模型的推拉式直接建模，《工程软件进展》127 (2019) 59–69。[13] X. Qin, Z. Tang, S. Gao，直接建模操作后特征模型的自动更新，《计算机辅助设计与应用》18 (2021) 170–85。[14] J. Fu, X. Chen, S. Gao，基于单元模型的特征模型与直接编辑自动同步，《计算机辅助设计与应用》14 (5) (2017) 680–692。[15] I. C. Braid，由多个面围成的实体合成，《ACM通讯》18 (4) (1975) 209–216。[16] H. B. Voelcker, A. A. Requicha，机械零件与加工过程的几何建模，《计算机》10 (12) (1977) 48–57。[17] V. Shapiro，实体建模，载于：《计算机辅助几何设计手册》，North-Holland，2002年，第473–518页。[18] S. Kyratzi, P. Azariadis，三维参数化模型的集成设计意图，《计算机辅助设计》（2021）103198。[19] S. Raghothama, V. Shapiro，零件族的拓扑框架，载于：第七届ACM实体建模与应用专题研讨会论文集，ACM，2002年，第1–12页。[20] R. Bidarra, W. F. Bronsvoort，语义特征建模，《计算机辅助设计》32 (3) (2000) 201–225。[21] S. Tornincasa, F. D. Monaco，CAD的未来与演进，载于：第14届国际研究／专家会议论文集，2010年，第11–18页。[22] H. Ault、A. Phillips，《直接建模：CAD 中的简易变更》，载于：第70届 ASEE EDGD 年中会议论文集，2016年，第99–106页。[23] A. R. Grayer，《几何建模的替代方法》，《计算机辅助设计》12（4）（1980）189–192。[24] J. R. Rossignac，《基于特征的编辑与实体模型查询中的问题》，《计算机与图形学》14（2）（1990）149–172。[25] I. Stroud、P. C. Xirouchakis，《CAGD——计算机辅助墓碑设计》，《工程软件进展》37（5）（2006）277–286。[26] B. Bettig、C. M Hoffmann，《参数化计算机辅助设计中的几何约束求解》，《工程计算与信息科学杂志》11（2）（2011）。[27] A. Nag、T. D. Gallagher、J. J. Dunne，《将计算机辅助设计模型中的选定特征转换为直接编辑特征的方法与系统》，美国专利 9,117,308（2015）。[28] J. Chad、H. David，《同步技术：工程组织的两全其美》，技术报告，Aberdeen Group，波士顿（2008）。[29] V. C. Lin、D. C. Gossard、R. A. Light，《计算机辅助设计中的变分几何》，ACM SIGGRAPH 15（3）（1981）171–177。[30] J. C. Chung、T.-S. Hwang、C.-T. Wu、Y. Jiang、J.-Y. Wang、Y. Bai、H. Zou，《集成机械设计自动化框架》，《计算机辅助设计》32（5-6）（2000）355–365。[31] D. Ushakov，《变分直接建模：如何在无历史记录 CAD 中保留设计意图》，技术报告，LEDAS Ltd（2008）。[32] C. M. Hoffman、A. Lomonosov、M. Sitharam，《几何约束系统的分解方案，第一部分：CAD 的性能指标》，《符号计算杂志》31（4）（2001）367–408。[33] F. Cordier、H. Seo、M. Melkemi、N. S. Sapidis，《从草图推断镜面对称三维形状》，《计算机辅助设计》45（2）（2013）301–311。[34] M. Mantyla，《关于欧拉算子建模空间的注记》，《计算机视觉、图形学与图像处理》26（1）（1984）45–60。[35] Q. Zou、H.-Y. Feng，《CAD 模型直接建模中信息不一致性消解的决策支持方法》，《高级工程信息学》44（2020）101087。[36] H. Hu、M. Kleiner、J.-P. Pernot，《几何方程组中的过约束检测与消解》，《计算机辅助设计》90（2017）84–94。[37] C. Gonz´alez-Lluch、R. Plumed、D. P´erez-L´opez、P. Company、M. Con-tero、J. D. Camba，《一种用于改善参数化建模中设计重用的约束冗余消除策略》，《工业计算机》129（2021）103460。[38] J. D. Camba、M. Contero，《评估几何设计意图标注对参数化模型修改活动的影响》，《工业计算机》71（2015）35–45。[39] S. Raghothama、V. Shapiro，《参数化实体建模中的边界表示变形》，《ACM 图形学汇刊》（TOG）17（4）（1998）259–286。[40] M. Lipp、P. Wonka、P. M¨uller，《Pushpull++》，《ACM 图形学汇刊》33（4）（2014）1–9。[41] H. A. Van der Meiden、W. F. Bronsvoort，《参数化模型中的拓扑变化跟踪》，《计算机辅助几何设计》27（3）（2010）281–293。[42] M. Hidalgo、R. Joan-Arinyo，《构造几何约束求解中的参数范围计算：实现与正确性证明》，《计算机辅助设计》44（7）（2012）709–720。[43] Q. Zou、H.-Y. Feng，《一种基于几何—拓扑不一致性跟踪的二次曲面边界表示模型鲁棒直接建模方法》，《计算机辅助工程》（2021）1–16。[44] H. Hu、M. Kleiner、J.-P. Pernot、C. Zhang、Y. Huang、Q. Zhao、S. Yeung，《几何过约束检测：综述》，《工程计算方法档案》28（7）（2021）4331–4355。[45] J.-F. Dufourd、P. Mathis、P. Schreck，《通过组装已求解子图形进行几何构造》，《人工智能》99（1）（1998）73–119。[46] J. C. Owen，《由尺寸约束进行几何的代数求解》，载于：第一届 ACM 实体建模基础与 CAD/CAM 应用研讨会论文集，1991年，第397–407页。[47] W. Bouma、I. Fudos、C. Hoffmann、J. Cai、R. Paige，《几何约束求解器》，《计算机辅助设计》27（6）（1995）487–501。[48] I. Fudos、C. M. Hoffmann，《求解几何约束系统的图构造方法》，《ACM 图形学汇刊》（TOG）16（2）（1997）179–216。[49] X.-S. Gao、C. M. Hoffmann、W.-Q. Yang，《通过轨迹交点求解空间基本几何约束构型》，载于：

第七届 ACM 实体建模与应用研讨会论文集，2002年，第95–104页。[50] L. A. Barford，《基于语言的通用实体模型约束图形编辑器》，博士学位论文，康奈尔大学（1987）。[51] D. Serrano，《概念设计中的约束管理》，博士学位论文，麻省理工学院（1987）。[52] S. Ait-Aoudia、R. Jegou、D. Michelucci，《约束系统的约简》，载于：Compugraphics 论文集，1993年，第83—92页。[53] R. S. Latham、A. E. Middleditch，《连通性分析：处理几何约束的工具》，《计算机辅助设计》28（11）（1996）917–928。[54] C. M. Hoffmann、A. Lomonosov、M. Sitharam，《寻找约束图的可求解子集》，载于：《约束编程原理与实践国际会议》，施普林格，1997年，第463–477页。[55] C. M. Hoffmann、M. Sitharam、B. Yuan，《使约束求解器更易用：过约束问题》，《计算机辅助设计》36（4）（2004）377–399。[56] D. Michelucci、S. Foufou，《几何约束求解：见证构型方法》，《计算机辅助设计》38（4）（2006）284–299。[57] S. E. Thierry、P. Schreck、D. Michelucci、C. F¨unfzig、J.-D. G´enevaux，《见证法在刻画欠约束、过约束和约束良好几何约束系统中的扩展》，《计算机辅助设计》43（10）（2011）1234–1249。[58] Q. Zou、H.-Y. Feng，《利用几何扰动进行直接建模的变分边界表示模型分析》，《计算设计与工程学报》6（4）（2019）606–616。[59] M. Fornasier、H. Rauhut，《压缩感知》，《成像数学方法手册》（2015）187–229。[60] S. Osher、W. Yin，《通过 l1 和 L1 优化进行稀疏恢复》，技术报告，加利福尼亚大学洛杉矶分校（2014）。[61] S. Murugappan、S. Sellamani、K. Ramani，《利用建议实现徒手草图美化》，载于：第6届 Eurographics 基于草图的界面与建模研讨会论文集，2009年，第69–76页。[62] B. Mills、F. Langbein、A. Marshall、R. Martin，《估计几何规则性的频率，以用于简单机械部件的逆向工程》，提交至《形状建模国际期刊》（2001）。[63] M. L. Mart´ınez、J. F´elez，《用于正确标注尺寸零件和过度标注尺寸零件的约束求解器》，《计算机辅助设计》37（13）（2005）1353–1369。[64] H. Zou、Y. Lee，《基于约束的二维草图重建三维多面体模型的美化与尺寸标注》，《计算机辅助设计》39（11）（2007）1025–1036。[65] F. C. Langbein、A. D. Marshall、R. R. Martin，《为逆向工程几何模型的美化选择一致约束》，《计算机辅助设计》36（3）（2004）261–278。[66] Y. Li、X. Wu、Y. Chrysathou、A. Sharf、D. Cohen-Or、N. J. Mitra，《Glob-fit：通过发现全局关系一致地拟合基元》，《ACM 图形学汇刊》30（4）（2011）52:1–52:12。[67] R. Hillyard、I. Braid，《计算机辅助机械设计中的尺寸与公差分析》，《计算机辅助设计》10（3）（1978）161–166。[68] J. R. Rossignac、H. B. Voelcker，《CSG 中的活动区域：用于加速边界求值、冗余消除、干涉检测和着色算法》，《ACM 图形学汇刊》（TOG）8（1）（1988）51–87。