

- 1. CAD信息提取
  - 1.1 对于数字CAD图纸的方法
    - 1.1.1 识别分类图层 【1】
    - 1.1.2 数字CAD图纸信息提取流程概述 【2】
    - 1.1.3 数字CAD图纸信息提取流程概述 【3】
  - 1.2 对于扫描CAD图纸
    - 1.2.1 微调LLM与CNN的二维图纸结构化信息提取流程概述 【4】
    - 1.2.2 基于图像处理的平面图矢量化与语义分割 【5】
    - 1.2.3 基于 OpenCV 轮廓分析的 2D 视图特征提取 【6】
  - 1.3 点云信息提取
- 2. BIM自动建模
  - 2.1 参数化自动建模
    - 2.1.1 基于dynamo visual 的BIM建模 【8】
    - 2.1.2 用Grasshopper 实现自动化 【2】
  - 2.2 LLM代理建模
    - 2.2.1 ReAct的 llm 代理建模 【4】
    - 2.2.2 基于协作LLM agent 与反思的 BIM 生成 【7】
      - 反馈循环
  - 2.3 IFC直接建模 【5】
- 引用

CAD2BIM整个过程分为两个阶段

1. CAD信息提取
2. BIM自动建模

# 1. CAD信息提取

---

目的：

CAD图纸根据数据形式分为两类，需要各自有对应的提取方法：

1. 数字CAD图纸 (Digital CAD Drawings):
  - 指使用CAD软件直接创建的电子文件，通常包含矢量图形数据。
  - 方法：通过规则和语义构建拆分 获得参数

## 2. 扫描CAD图纸 (Scanned CAD Drawings) 或 栅格化CAD图纸 (Rasterized CAD Drawings):

- 指将纸质CAD图纸扫描后得到的图像文件，本质上是位图或像素图。

### 1.1 对于数字CAD图纸的方法

#### 1.1.1 识别分类图层 【1】

为后续的 构建提取 提供了 遵循了良好的图层管理规范CAD图纸：即不同的结构构件类型（如柱、梁、板）、轴网、尺寸标注、文字注释等被放置在各自独立的、预定义的图层上。

用于在CAD图纸中构建图层属性，即使图层名称不固定，也能自动识别和分类图层（例如墙、门、窗）。主要步骤如下：

- 特征元素与相关元素提取：** 首先从需要分类的目标图层中找出最具代表性的**特征元素 (FE)**，比如墙体的线条、门的弧形、文字注释等。同时，也会提取与这些特征元素有拓扑关联的**相关元素 (RE)**，例如图层中的其他辅助图形。
- 条件判断与打分机制：** 接着，系统会根据预设的条件来判断这些元素：
  - **必要条件 (NC)：** 这是特征元素或相关元素必须满足的基本特征，如果不满足，则该图层不可能属于目标类别。
  - **充分条件 (SC)：** 虽然不是强制性的，但如果满足这些条件，则能增加该图层与目标类别匹配的可能性。在判断过程中，会先检查所有必要条件，只有通过这些基本判断后，才会进一步检验充分条件。每个特征元素的得分会根据其满足的充分条件数量来计算，最后累加得到该图层的总分。
- 图层分类决策：** 对所有待检测图层中的特征元素逐一计算得分，最终选择得分最高的图层作为对应的目标图层。例如，通过不同的判断策略，可以分别识别轴网文字、尺寸标注、门窗符号以及墙体等常见的图层。

#### 1.1.2 数字CAD图纸信息提取流程概述 【2】

该方法分为两个主要阶段：**阶段一：2D CAD图纸准备 (2D CAD Drawing Preparation)** 这个阶段是信息提取的基础，涉及对原始数字CAD文件进行预处理，以获得简化的、适合后续处理的2D几何输入。主要步骤包括：

- 清理 (Cleaning):** 移除图纸中对于BIM模型生成非必需的元素，例如：
  - 填充图案 (Hatches)

- 尺寸标注 (Dimensions)
  - 文字标签 (Text)
  - 图框 (Drawing borders)
2. **分层 (Layering):** 将代表不同建筑构件（如柱、墙、楼板、门窗等）的2D几何图形分配到各自独立的图层。这是后续自动化识别和处理不同类型构件的关键。
3. **重绘 (Redrawing):** 进行一些必要的、简单的几何编辑工作，以处理自动检测困难的情况。例如：
- 连接和闭合代表柱子、楼板轮廓的多段线。
  - 连接因门窗插入而被打断的墙体线段，使其成为连续的墙体表示。

**阶段二：3D BIM模型生成 (3D BIM Model Generation)** 这个阶段侧重于利用第一阶段准备好的CAD数据，自动化地提取信息并生成BIM构件。主要步骤和技术包括：

1. **读取CAD数据:** 将清理后的CAD图纸文件读入处理环境（论文中使用了Rhino）。)
2. **几何检索与参数提取:**
  - **基于图层检索:** 从对应的图层中检索定义不同构件的2D几何图形。
  - **几何分析与构件识别:**
    - **柱和楼板:** 扫描指定图层中的几何实体，通过几何分析（例如识别闭合多段线）检测构件类型，提取插入点、尺寸、形状等参数。
    - **墙:** 从墙图层提取线段，使用启发式规则（考虑线间距离、共线性等）将代表墙两侧的平行线进行配对，生成墙体参考线和厚度参数。
    - **幕墙:** 识别代表竖梃（mullions）的实体。由于竖梃可能被分割，通常需要**手动绘制一条辅助线**来定义幕墙的起止点并连接相关竖梃。然后分析得出竖梃间距模式（一系列距离值）和其他尺寸参数。
    - **门窗:** 通常以图块（block）形式存在。需要将其与所在的墙体进行配对（通过查找最近的墙体参考线）。提取插入点、宽度等参数。
  - **参数化:** 将提取的几何信息转化为生成BIM构件所需的参数（如插入点、参考线、尺寸、厚度、形状、模式等）。

### 1.1.3 数字CAD图纸信息提取流程概述【3】

通过读取指定图层，系统可以直接识别出属于特定构件类型的几何图形。

- **几何信息提取 (Geometric Information Extraction):**
  - **构件识别:** 在预定义的图层上识别代表柱（通常是矩形或多边形）、梁（通常是线段）、板（通常是闭合多边形）的基本几何形状。
  - **定位与拓扑:** 利用结构轴网图层（通常包含轴线和轴号）来确定构件的精确位置，并帮助定义梁的跨度和端点（假设梁与轴网对齐）。

- **语义信息提取 (Semantic Information Extraction):**

- 这是该方法区别于纯几何转换的关键。它专注于**提取CAD图纸中的文本信息**，并将其与对应的几何构件关联起来。
- **算法应用:** 使用特定的算法（如基于邻近度搜索、文本模式匹配）来查找和识别与几何构件相关的文本注释。
- **提取内容:**
  - **截面尺寸:** 识别并提取靠近梁、柱几何图形的尺寸标注文本（例如, "B=300, H=600" 或 "400x500"）。
  - **楼板厚度:** 在楼板区域内或附近查找并提取厚度标注文本（例如, "T=150"）。
  - **构件编号/标签:** 提取与构件关联的唯一标识符（例如, "KZ-1", "L-2"）。
  - **其他信息:** 理论上也可以提取材料、配筋等信息（如果它们以文本形式清晰标注在图纸中）。

## 1.2 对于扫描CAD图纸

### 1.2.1 微调LLM与CNN的二维图纸结构化信息提取流程概述【4】

- **数据准备与微调:** 收集并整理了超过3万张图纸，对 Llama 3.1 基础模型进行了微调，以显著提高对二维图纸的识别准确率。
- **预处理:** 对图纸进行缩放、裁剪和去噪处理，并转换为 LLM 可处理的数字图像格式。
- **特征提取:** 利用卷积神经网络（CNN）从处理后的图纸中提取关键特征（如线条形状、方向、长度及图形内的文本信息），形成嵌入编码。
- **LLM 处理:** 将提取的图像特征与相应的文本描述对齐进行训练，使 LLM 学习图像特征与文本描述的映射关系。模型利用其自然语言理解能力，结合上下文理解图纸信息。
- **参数提取:** 从二维图纸中提取建模所需的特征参数，如尺寸、形状、比例和注释内容。
- **结构化输出:** 将提取的信息以结构化的 JSON 格式输出，供后续的建模 Agent 使用。
- **优化:** 通过持续使用新数据进行微调和优化，防止模型过拟合，提高模型适应性，减少 AI 幻觉。

### 1.2.2 基于图像处理的平面图矢量化与语义分割【5】

- 首先对输入的扫描二维平面图（栅格图像）进行预处理，包括二值化（转为黑白图像）和去噪（去除小的斑点和干扰）。

- 接着，通过分析图像中的连通分量（识别大面积的黑色像素区域）并结合形状和尺寸约束来检测墙体。
- 检测到的墙体像素区域被矢量化，即用直线段来近似表示墙体轮廓。
- 然后，在墙体矢量线中检测中断或识别特定的图形符号（如平行线、圆弧）来定位门窗洞口。
- 最后，通过识别被墙体围合的白色区域（如使用泛洪填充或连通域分析）来进行房间分割，并将检测到的门窗与其相邻的房间关联起来。

### 1.2.3 基于 OpenCV 轮廓分析的 2D 视图特征提取 【6】

该方法主要使用 OpenCV 库处理输入的 2D 工程图纸的图像（可以是照片或扫描件），特别是处理其正交视图（如主视图、俯视图、侧视图）。

首先进行图像预处理：加载图像，转换为灰度图，应用高斯模糊以减少噪声。

使用 Canny 边缘检测算法 识别图像中的线条（边缘）。

接着，从边缘图像中查找轮廓 (Contours)，即连续的曲线或边界。

利用 Ramer-Douglas-Peucker 算法 (多边形逼近) 对找到的轮廓进行简化，将其近似为由直线段组成的多边形，从而获得关键的顶点 (Vertices) 坐标。

最终目标是从每个 2D 视图的轮廓多边形中提取出顶点坐标等几何特征。

## 1.3 点云信息提取

参数提取是在点云经过深度学习语义分割之后进行的，主要步骤如下：

1. 语义分割 (Semantic Segmentation): 使用深度学习模型（论文中可能具体说明了所用的网络架构，例如基于 PointNet++, RandLA-Net 或其他改进的网络）对输入的原始点云进行处理。为点云中的每个点分配一个语义标签，例如墙壁 (wall)、地板 (floor)、天花板 (ceiling)、门 (door)、窗户 (window)、家具 (furniture)、杂物 (clutter) 或其他结构/非结构元素。这是后续参数提取的基础。
2. 规则结构元素参数化 (Regular Structural Elements):
  - 平面检测: 对于被分割为墙壁、地板、天花板的点簇，通常使用 RANSAC (Random Sample Consensus) 或类似算法来拟合平面。参数提取: 从拟合的平面中提取关键参数，如平面的法向量、位置、边界（通常通过点云投影到平面后计算二维轮廓获得）。对于墙体，可能还会估算厚度（如果点云包含墙体两侧信息或基于先验知识）。
  - 开口元素参数化 (Openings - Doors/Windows): 在墙壁点云簇中，通过识别点云密度的中断或特定的几何形状（如矩形边界）来检测门和窗的位置。提取开口的参数，主要是它们在墙体上的位置、高度和宽度。这通常通过分析开口边界点的坐标来实现。

### 3. 不规则/非结构化元素处理 (Irregular/Unstructured Elements):

- **类别聚类与边界框提取**

针对每一个非结构化类别采用 DBSCAN 聚类得到对象的点云，然后计算其最小外接矩形作为初步的边界框。

- **边界框优化**

对初步边界框进行尺寸调整以更符合实际对象尺寸，剔除体积低于预设阈值（例如  $0.05, m^3$ ）的小簇，确保最后生成的 BIM 对象具有合理的尺寸和位置。

---

## 2. BIM自动建模

### 2.1 参数化自动建模

自动建模过程的关键在于从2D **CAD几何图形中提取生成BIM构件所需的参数**。 这些参数包括：

- 构件的位置（如插入点）
- 尺寸（如长、宽、高、厚度）
- 形状或轮廓（如柱截面、楼板边界）
- 参考线或路径（如墙体中心线）
- 模式（如幕墙竖挺间距列表）

#### 2.1.1 基于dynamo visual 的BIM建模【8】

对于结构化元素（墙，门，窗）基于使用Dynamo Visual编程设计的参数算法。它用可重复的操作代替了元素的手动创建，并提高了Revit软件中工作流的生产率。Dynamo算法工作流的视觉流程图如图8所示。该算法由连续组组成。每个组包含许多通过链接连接的节点。该算法从特定文件中导入输入数据，然后将数据分配给单独的坐标。之后，它应用了具有节点，链接和Python代码的适当编程的脚本，以创建墙壁，地板，天花板和开口的不同元素。该发电机算法不是手动创建BIM模型，而是有效地自动化了所有结构化元素的开发。 ![[Pasted image 20250408224929.png]]

非结构元素（家具） ![[Pasted image 20250408225110.png]] Autodesk Revit 和 Dynamo 自动化处理点云中识别出的非结构化元素（如家具、设备等）。它首先读取这些元素的 3D 边界框数据，提取其位置和尺寸。然后，在 Revit 中选择一个标准 BIM 库构件（例如通用模型），将其放置在计算出的位置，并调整其尺寸以匹配边界框。最后，

通过自定义 Dynamo 逻辑，将这些构件与其邻近的墙体或楼板连接起来，实现非结构化元素在 BIM 模型中的自动化、参数化和集成化表示。

### 2.1.2 用Grasshopper 实现自动化 【2】

流程：

- Grasshopper 节点被用来接收这些提取的几何参数和数值。
- 通过 Grasshopper 与 ArchiCAD 的连接插件（ACC），这些参数被传递给 ArchiCAD，用于设置并生成相应的BIM构件（如柱、墙、楼板、门窗、幕墙）。

**局限性亦与参数化相关：** 论文中提到的挑战，如幕墙复杂模式和转角的处理、门窗开启方向的精确控制，也反映了当前工具链在**参数化控制特定BIM属性方面的局限性**。例如，门窗的开启方向很难通过Grasshopper的参数直接精确控制，往往需要依赖默认设置或后续手动调整。

## 2.2 LLM代理建模

### 2.2.1 ReAct的 llm 代理建模 【4】

架构与流程: ReAct 范式: 采用基于 ReAct 范式的架构，形成“思考-行动-观察-思考”的闭环。 Parser 模块: 作为 LLM 输出和工具输入参数之间的桥梁，将 LLM 识别图纸后输出的原始字符串（包含详细规格和特征参数）转换为工具（如 Revit API）可用的输入参数。通过正则表达式等方法提取需要调用的工具及其参数信息。 Agent 学习与执行: 训练 LLM Agent 理解具体的建模需求（建筑类型、规模、功能需求等），学习如何使用 Revit 等 BIM 软件。 参数化建模: 训练好的 Agent 根据需求设置参数，调用 Revit API 函数编写驱动代码，在 Revit 中通过参数化建模创建墙体、楼板、柱子等基本构件。可以随时通过调整参数值修改构件尺寸和属性。 模型验证与优化: 完成建模后，Agent 会验证模型是否符合设计要求和规范，并根据反馈进行调整和优化。

### 2.2.2 基于协作LLM agent 与反思的 BIM 生成 【7】

该方法被称为 Text2BIM，其核心是一个基于大型语言模型 (LLM) 的多智能体 (Multi-Agent)协调 框架，旨在直接从自然语言文本指令生成 BIM 模型。 协作llm agent包括:

- 产品所有者(product owener): 将用户输入的简单模型说明增强，生成详细的要求文档
- 建筑师(Architect): 根据建筑知识制定文本建筑计划
- 程序员(programmer): 分析要求并写入建筑代码
- 审阅者(reviewer): 提供代码优化建议，以解决模型中确定问题

基于规则模型评估：

1. 几何冲突：门窗重叠，墙体交叉
2. 语义属性定义：组件信息完整性正确性，GUID唯一性等
3. 建筑常识检查：承重结构合理性

将检查的输出导出到BCF文本文件，交给审阅者(reviewer)提出改进建议，最后审阅者agent将建议反馈给程序员agent，不断迭代直至BCF无错误

## 2.3 IFC直接建模 【5】

该方法利用的矢量化信息（墙线、洞口位置、房间边界）。将二维的墙体矢量线沿垂直方向拉伸 (Extrusion) 到预设或推断的高度，生成三维墙体对象。根据识别出的二维洞口位置和类型，在相应的三维墙体上创建并放置三维门窗构件，通常具有默认的高度和竖向偏移。基于二维房间分割的边界多边形，创建楼板 (Slabs)（通常包括楼面和天花板）。利用三维墙体和楼板围合的区域，定义三维空间对象 (Spaces)，代表房间体积。最关键的是，为所有生成的三维几何体赋予语义信息，将它们分类为标准的 BIM 对象类型（如 IFC 标准中的 IfcWall, IfcDoor, IfcWindow, IfcSlab, IfcSpace），并建立它们之间的拓扑关系（如门窗属于哪个墙，墙围合哪个空间）。最终输出为具有丰富信息的 BIM 文件（IFC 格式）。

# 引用

1. An Automated Layer Classification Method for Converting CAD Drawings to 3D BIM Models
2. Automated Generation of BIM Models from 2D CAD Drawings
3. Semiautomatic Structural BIM-Model Generation Methodology Using CAD Construction Drawings
4. Research on the automatic BIM modeling method of substation based on LLM agent
5. Automatic reconstruction of 3D building models from scanned 2D floor plans (2015)
6. Photo2CAD: Automated 3D solid reconstruction from 2D drawings using OpenCV (2023)
7. Text2BIM: Generating Building Models Using a Large Language Model-based Multi-Agent Framework



## 8. Automated BIM generation for large-scale indoor complex environments based on deep learning (2024)