

北京師範大學

本科生毕业论文（设计）

毕业论文（设计）题目：

基于形状语义和增强学习的建筑立面点云解译与重建的方法

院 (系): 地理学与遥感科学学院

专 业: 地理信息系统

学 号: 201011172008

学 生 姓 名: 徐翔

指 导 教 师: 张立强

指导教师职称: 教授

指导教师单位: 地理学与遥感科学学院

2014年5月
北京师范大学学士学位论文（设计）诚信承诺书

本人郑重声明：所呈交的学士学位论文（设计），是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

本人签名： 年 月 日

北京师范大学学士学位论文（设计）使用授权的说明

本人完全了解北京师范大学有关收集、保留和使用学士学位论文（设计）的规定，即：本科生在校攻读学位期间论文（设计）工作的知识产权单位属北京师范大学。学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许学位论文（设计）被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。保密的学位论文在解密后遵守此规定。

本论文（是、否）保密论文。
保密论文在_____年解密后适用本授权书。

本人签名： 年 月 日

基于形状语义和增强学习的建筑立面点云解译与重建的方法

摘要

建筑立面建模是城市建模的重要组成部分。本文基于形状语义和增强学习优化策略，提出了一种建筑立面激光点云解译和重建的方法。首先基于数据驱动对点云进行建筑学语义上的预分类，接着设计了可以灵活应用于各种建筑立面风格的二分形状语义。最后将建筑立面的最佳形状语义求解的问题建模为马尔科夫决策过程，采用增强学习算法结合支持向量机预分类的结果优化求解，实现对建筑立面的解译和三维重建。实验表明，本文提出的方法不仅能够较为精确的对建筑立面进行精细建模，而且能够适应于各种风格建筑立面的解译，对于建筑立面点云缺失的情况也有很好的鲁棒性。

关键词：形状语义，增强学习，建筑立面建模，激光雷达点云

Building Facade Point Cloud Parsing and Reconstruction with Shape

Grammar and Reinforcement Learning

ABSTRACT

Building facade modelling is an important part of city reconstruction. This paper proposes a method to building facade point cloud reconstruction with shape grammar and reinforcement learning. We start by designing metrics for support vector machine classifying of building facade point cloud. After that, we introduce a flexible shape grammar to express the structural of building facade. In order to obtain the optimal grammar of a building facade, Base on the result of pre-classify, we model this problem as Markov Decision Process and solve it with reinforcement learning. Finally, the 3D model of building facade is reconstructed. The experiments shows that our method can model the refined structures of building facade and can apply to facades of different style. Our algorithm also performs well even when the point cloud is seriously incomplete.

KEY WORDS: shape grammar, reinforcement learning, building facade modelling, Lidar point cloud

目录

基于形状语义和增强学习的建筑立面点云解译与重建的方法	1
摘要	1
第一章 引言	1
第二章 基于数据驱动的点云分类	4
2.1 预处理	4
2.1.1 噪声去除	4
2.1.2 坐标系转换	4
2.2 点云特征提取	5
2.2.1 曲度	6
2.2.2 密度	6
2.2.3 深度	6
2.3 支持向量机分类	7
第三章 过程建模	9
3.1 二分形状语义	9
3.2 建筑立面的表示	10
3.3 立面语义的求解思路	11
第四章 增强学习	13
4.1 马尔科夫决策过程	13
4.2 动态规划	15
4.3 蒙特卡罗方法	16
4.4 增强学习算法	16

4.5 半马尔科夫决策过程.....	17
第五章 城市建筑立面三维建模.....	18
5.1 最优形状语义的求解.....	18
5.2 建筑立面深度信息的恢复.....	19
第六章 实验验证	20
6.1 数据集.....	20
6.2 建筑立面结构分类结果.....	20
6.3 建筑立面解译及重建结果.....	21
6.4 算法效率.....	24
第七章 结 论	26
参考文献	27
附 录 1 几种建筑的二分形状语义	29
附 录 2 算法伪代码	30
致 谢	32

图表目录

表 1 仅包含窗户和墙面的二分形语义.....	10
表 2 实验的效率分析.....	24
图 1 算法概述.....	2
图 2 利用表 1 的二分形状语义对一个简单的建筑立面进行剖分得到的剖分树	11
图 3 马尔科夫决策过程各要素之间的联系.....	13
图 4 某点云缺失严重的建筑立面分类结果。 (a) 原始点云 (b) 分类后的 结果, 红色的为窗户, 灰色的为墙面.....	21
图 5 Cathedral 教堂分类结果。 (a)原始点云; (b)分类后的结果。总共有四 类, 其中灰色的为墙面, 粉红色的为门, 黄色的为窗户, 红色的为屋 顶, 靛青色的为立柱。	21
图 6 五道口华清嘉园分类结果。 (a)原始点云; (b)分类后的结果。总共有四 类, 其中灰色的为墙面, 蓝色的为屋顶, 黄色的为窗户外的小阳台, 红色的为窗户, 粉红色的为突出的大阳台。	21
图 7 地质大学图书馆的一个立面建模过程。 (a) 原始点云。 (b) 形状语义 解译结果。 (c) 恢复深度建模结果。	22
图 8 北师大某建筑建模立面建模过程。 (a) 原始点云。 (b) 形状语义解译 结果。 (c) 恢复深度建模结果。	22
图 9 北师大科技楼立面建模过程。 (a) 原始点云。 (b) 形状语义解译结果。 (c) 恢复深度建模结果。	22

图 10 点云严重缺失的立面建模过程。 (a) 原始点云。 (b) 形状语义解译结果。 (c) 恢复深度建模结果。 23

图 11 包含阳台的建筑立面建模过程。 (a) 原始点云。 (b) 形状语义解译结果。 (c) 恢复深度建模结果。 23

图 12 cathedral 教堂建模过程。 (a) 原始点云。 (b) 形状语义解译结果。 (c) 恢复深度建模结果。 24

第一章 引言

研究的背景和意义

随着 GIS 服务领域的不断拓展（如微软 Virtual Earth 和 Google Earth 的发展），导航、城市漫游、城乡规划、遗迹保护、风险分析等应用领域迫切需要三维数字城市的建模和自动重建。而建筑立面的重建是数字城市建模的重要组成部分。传统的建模方法主要是基于人工或是半自动的重建：首先从 2 维 GIS 数据生成建筑的轮廓，然后简单地给轮廓赋予高程属性得到粗糙的模型。精细的模型可以将粗糙的模型导入一些商业的三维建模软件（比如 3DStudio Max）中处理得到。最后再利用近距离拍摄得到的图片对模型的每一面添加纹理。由于城市里的建筑数量庞大，且建筑形状复杂，人工建立城市模型需要建模者要有一定的经验，并耗费大量时间。目前存在一些半自动生成建筑模型的方法^[1]，但效果都不令人满意。

目前的建筑立面建模方法主要是从二维的图像中来提取建筑的精细结构。近年来，随着激光雷达技术的发展，人们已经可以通过车载激光雷达或是机载激光雷达对地面进行快速的扫描从而获得城市模型的点云。与传统的图片相比，激光雷达点云是建筑模型的直接三维表示，包含了恢复建筑结构特征的几乎所有特征。一些研究显示了激光雷达数据可以高效地应用于城市的重建中^[2-4]。

本文将探索如何从激光雷达点云中提取建筑精细的结构特征，从而为建筑的自动化三维重建提供基础。目前关于建筑结构特征提取的方法可以分为数据驱动（自底向上）和模型驱动（自顶向下）两大类。数据驱动的方法利用建筑图像或是激光雷达点云自身的信息，构造相应的特征属性自底向上对建筑进行分类和识别。Becker^[5]等提出一种单元分解的方法来重建建筑立面，然而他们的方法需要一个额外的粗糙的三维模型作为输入，并且他们的方法只考虑了建筑墙面的窗户结构，因此无法提取精细复杂的建筑墙面；Ning^[6]等也提出了恢复粗糙建筑三位模型结构特征的方法。为了显示更复杂和精细的建筑结构，目前的研究成果大多需要大量的人工交互^[7,8]。为了利用建筑的对称性，一些特殊设计的方法被用来从照片中提出建筑的精细结构^[9-12]。

数据驱动的方法虽然很精确，但是却无法很好显示建筑的语义特征（比如，建筑由门、窗、阳台、屋顶等建筑学意义上的物体组成）。为此人们提出了形状语义的概念用来自顶向下对建筑进行建筑学语义上的建模。最早使用形状语义探索城市建模的是 Parish 等^[13]和 Wonka 等^[14]的工作。前者显示了如何使用体模型以及渲染的包含精细结构的墙面生成大规模的城市模型。后者展示了如何为单个建筑生成几何学意义上的墙面结构。Muller 等^[15]最早将过程建模的方法应用到形状语义对城市建模中来，提出并实现了 CGA shape，使用该软件，可以根据简单的规则快速地生成城市的三维模型。该实现目前已经集成到了 ESRI

公司的 CityEngine 软件中。使用形状语义进行建筑墙面结构的提取在过去几年已经得到了广泛的关注^[16-21]。

采用形状语义的难点在于定义适合建筑的语义以及如何高效地求解最优语义。Olivier Teboul^[21]等在之前工作^[20]的基础上定义了适合建筑立面建模的二分形状语义，并采用增强学习的方法从二维图像中提取建筑的建筑学语义。但是为了减少增强学习的状态量，他们采取了状态集聚的方法，只能对高度对称的建筑进行建模，无法对于层之间不对称的建筑。为了实现对不规则建筑进行建模，Zhang Hao^[22]等人引入分层的概念，并采用最大化对称度的方法递归地完成不规则建筑的解译；Riemenschneider^[23]基于不规则格网设计了更复杂的形状语义对建筑剖分。目前大部分方法^[20-23]均是对建筑图像进行建模，本文将研究如何设计形状语义对建筑的点云立面进行解译并进行高效地求解，以实现对不规则且存在点云缺失的立面进行建模。

研究内容

本文将研究如何高效的对建筑立面点云进行建模，恢复其建筑学意义上的特征（比如窗户，墙面，阳台等）。进而为建筑的三维重建提供基础。具体的研究内容分为以下三点：

- 1 基于数据驱动的点云分类，即设计合适的指标对建筑立面点云进行分类。对点云进行预分类是之后进行建筑学语义上解译的基础。在这一步中，将研究如何高效的计算某个点云区域属于特定的建筑结构（窗户、墙面、阳台、门等结构）的概率。
- 2 研究建筑立面解译的方法。将探索过程建模领域在建筑表达中的应用，设计一个对尽可能多的类型的建筑具有普适性的形状语义来表示建筑的结构。这一步是建筑立面解译的核心，形状语义的灵活与否决定了可适用的建筑范围有多大。
- 3 基于模型驱动的建筑立面建模。对于一个特定的建筑点云，在对其进行基于数据驱动的预分类之后，研究如何快速推测出其对应的形状语义。得到建筑立面对应的形状语义后，恢复相应的深度信息，即可重建立面的三维模型。

研究思路及技术路线

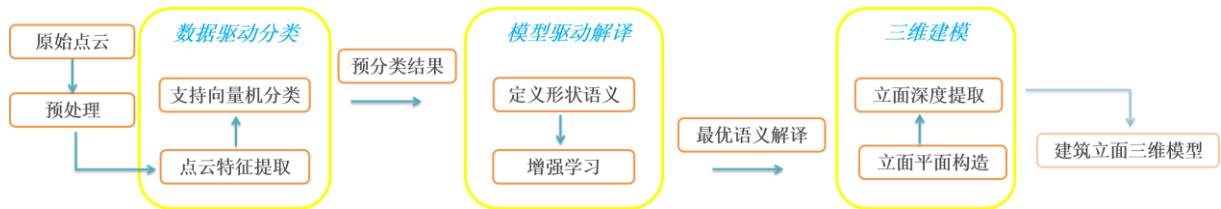


图 1 算法概述

图 1 显示了本文提出的算法流程。首先对原始的建筑立面点云数据进行预处理，在这

一步中，主要是消除奇异点、提取出建筑立面的主平面，并对建筑点云转换坐标系使得立面的主平面对齐到直角坐标系中的 X-Y 平面，以方便之后的处理。

接着，我们对点云进行基于数据驱动的预分类。预分类采用网格化的方法对预处理后的点云进行划分，并为网格定义了三种之后用于分类的属性（即网格内点的密度，平均深度以及平均曲度），之后便选取训练样本采用支持向量机进行分类。分类之后我们便得到了每个网格属于某种建筑结构的概率。这个概率在之后将会用于指导增强学习。

第三步，采用基于模型驱动的方法来对建筑立面进行解译。在这一步中，我们首先要定义适用于输入建筑的形状语义，该形状语义定义生成建筑的规则；在这之后，增强学习算法结合预分类的结果求解出建筑立面最佳的形状语义解译。

最后，根据形状语义解译，我们构造出立面的平面构造，然而为了还原三维立面，还缺少深度信息。此时我们根据解译结果从点云中提取各个建筑结构的深度值，最终完成对建筑立面的三维建模。

在下文的编排中，第二章将详细介绍基于数据驱动的点云预分类；第三章介绍过程建模及其在建筑解译中的应用。第四章介绍将用来求解最佳形状语义的增强学习算法。第五章介绍城市立面的建模。在这一章中介绍如何利用增强学习算法求解立面的最佳形状语义解译，并根据该语义从原始点云中提取深度信息，从而完成对建筑立面的三维建模。第六章给出了对建筑立面点云的处理结果，进行了实验验证。最后，第七章总结本文的工作。

第二章 基于数据驱动的点云分类

本文的目标要对建筑立面这样一个连续的空间进行建模，因此传统的基于点的分类方法并不适用，为此，我们首先将建筑立面齐到 X-Y 平面，接着对建筑立面主进行网格的离散化，每个三维的小网格可以用 $G(x, y, z)$ 来表示，其中 x, y, z 表示网格在立面 X, Y, Z 方向上的索引。由于建筑面上的结构体（如门、窗、阳台等）在立面的不同深度分布，且分布在 X-Y 平面的不同位置，因此 x, y 索引相同，但是 z 索引不同的网格表示的是同一种建筑结构，为了方便下文采用二维的形状语义解译，将 x, y 索引相同，但是 z 索引不同的网格合并成一个网格，记为 $G(x, y)$ 。实际上这种做法将点云的分类转化成了二维，大大简化了问题的复杂性，使得在第三章中我们得以使用二维形状语义对其解译，在解译完毕之后，我们在第四章设计了巧妙的方法来得到每种建筑结构的深度值，从而完成了从二维到三维的恢复。

点云分类的目标是得到建筑立面的网格 $G(x, y)$ 属于某种建筑结构的概率分布，其中 1 为建筑结构的类别。为此，对每个网格设计相应的指标，最后采用支持向量机的分类方法对每个网格进行分类，进而得到该概率分布，为下文的基于形状语义的建模提供基础。

2.1 预处理

在本文中，建筑立面剖分系统的输入是带有噪声以及有数据缺失的建筑立面激光雷达点云。预处理的过程包括基于统计的方法移除噪声以及转换坐标系。

2.1.1 噪声去除

点云中噪声的存在对于后面计算曲度会有较大的影响，因此，采用统计方法去除点云噪声^[24]。对于输入点云的每一点，计算它到邻近点的距离。假设没有噪声的点到其邻近点的距离分布是正态分布，计算该点到邻近点距离的均值 μ 和标准差 σ 。那些距离落在 $\mu \pm \alpha \cdot \sigma$ 之外的点则被视为噪声而移除，其中 α 是和选取的邻近点的数目相近的一个值。本文实验中， $\alpha = 1.0$ ，邻近点的数目为 20。

2.1.2 坐标系转换

为了方便后续的处理，把建筑立面所在的平面转换到 X-Y 平面中去，即，建筑的水平方向与 X 轴平行，竖直方向与 Y 轴平行。这样一来建筑立面点云的 z 坐标就代表了建筑结构的深度信息，并且建筑的水平和竖直方向与坐标轴平行，以便于下文对建筑立面进行网格剖分。

建筑立面的坐标系变换陈述为如下问题，给定建筑点云当前所处的基 $M_2 = \{\alpha_1, \alpha_2, \alpha_3\}$

以及目标坐标系的基 $M_2 = \{\beta_1, \beta_2, \beta_3\}$ ，求 M_1 到 M_2 的过渡矩阵。其中， $\beta_1 = \{1, 0, 0\}, \beta_2 = \{0, 1, 0\}, \beta_3 = \{0, 0, 1\}$ 。而要得到两组基之间的过渡矩阵，只需要给出目标坐标系的基中的每个向量 $(\beta_1, \beta_2, \beta_3)$ 在当前所处基下的坐标 (p_1, p_2, p_3) ，依次以这些坐标为列向量组成矩阵：

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} \quad (2.1)$$

P 即是从 M_1 到 M_2 的过渡矩阵。

目前的任务是计算 $(\beta_1, \beta_2, \beta_3)$ 在当前所处基下的坐标 (p_1, p_2, p_3) 。由前述知，建筑的水平方向对其到 X 轴，建筑的竖直方向对齐到 Y 轴，因此，基向量 β_1 在当前基下对应的坐标 p_1 应该是建筑水平方向的向量，基向量 β_2 对应的坐标 p_2 则是建筑竖直方向的向量，基向量 β_3 对应的坐标 p_3 则是建筑平面的法向量。

我们采用随机抽样一致模型（RANdom Sample Consensus, RANSAC）^[25] 来得到建筑立面的平面参数，继而得到其归一化的法向量，即为 p_3 。对于 p_2 ，使用建筑墙面边缘线所处的向量表示，而 p_1 则可以由 p_2 和 p_3 的外积求得。得到过渡矩阵 P 后，对原始点云中的点 X 应用变换 P 即可得到在新坐标系下的坐标 X' ：

$$X' = PX \quad (2.2)$$

2.2 点云特征提取

为了有效地识别建筑立面的结构（如门、窗户、阳台），设计了基于网格的点云特征提取方法。首先以一定的分辨率对建筑立面进行剖分，使得建筑立面由一系列三维的小网格组成。在本文中，网格的大小设置为 0.1 米。这样对点云分类的任务就变成了对这些网格的分类。本文设计了三种网格的属性，用来通过支持向量机对建筑立面点云中的结构单元分类，下面分别介绍。

2.2.1 曲度

曲度反映了建筑局部结构的变化情况，对于提取建筑结构的变化有很大作用。定义网格内的曲度特征是网格内所有点的曲度的平均值。计算某一点的曲度近似为计算该点和邻近的 N 个点近似组成的平面的曲度，二者可以通过分析邻近点的协相关矩阵的特征向量和特征矩阵得到^[24, 26]。

对于一点 p_i ，它的协相关矩阵为：

$$C = \frac{1}{k} \sum_{i=1}^k (p_i - \bar{p}) \cdot (p_i - \bar{p})^T, C \cdot \vec{v}_j = \lambda_j \cdot \vec{v}_j, j \in \{0, 1, 2\} \quad (2.3)$$

其中，k 是 p_i 的邻近点的数目， \bar{p} 代表邻近点的三维中心坐标， λ_j 是协相关矩阵的第 j 个特征值， \vec{v}_j 是第 j 个特征向量。

点 p_i 的曲度为：

$$\gamma_{p_i} = \lambda_0 / (\lambda_0 + \lambda_1 + \lambda_2) \quad (2.4)$$

假设网格 G 中有 n 个点，那么该网格的曲度则表示为：

$$\gamma_G = \frac{1}{n} \sum_{i=1}^n \gamma_{p_i} \quad (2.5)$$

2.2.2 密度

除了曲度外，网格内点的密度变化同样能反映不同的建筑结构，另外，对于密度还能够明确地反映数据缺失或是背景网格。

设网格 G 中有 n 个点，点数最多网格内的点数为 N，则该网格的密度表示为：

$$d_G = n / N \quad (2.6)$$

2.2.3 深度

由于建筑表面各个结构通常位于不同的平面，通过深度值可以较容易识别出不同的建筑立面结构。我们已经将建筑平面转换到了 X-Y 平面，因此点云的 z 坐标就表示了深度的信息。

假设网格 G 中有 n 个点，那么该网格的深度表示为：

$$s_G = \frac{1}{n} \sum_{i=1}^n z_i \quad (2.7)$$

其中 z_i 为点 i 的 z 坐标。

以上三种特征将作为网格的特征向量被下一节将要介绍的支持向量机的分类使用。

2.3 支持向量机分类

由于建筑立面的建筑学意义上的结构(如窗户、阳台、门、屋顶等)是从人类认知角度去定义的，采用非监督的方法往往很难得到满意的结果。比如，我们无法控制非监督分类得到的分类数；非监督分类之后，很难判断得到的子类属于哪种建筑结构。为此，本文采用了支持向量机^[27]的方法进行分类。

支持向量机是近年来十分流行的一个分类方法，使用它分类主要包括训练和预测两部分。首先选择训练集，训练集里的每一个样本都有一个类别标签及多个属性值。支持向量机的目标就是在训练集的基础上产生一个模型，后使用这个模型对测试集里的数据进行预测，以得到测试集中样本的类别估计。用数学的语言来描述就是：

给定一个训练集 $(x_i, y_i), i = 1, \dots, l, x_i \in R^n$ 表示样本的特征向量， $y \in \{-1, 1\}^l$ 表示样本的类别向量，支持向量机求解以下正定规划的最优解：

$$\begin{aligned} & \min_{w, b, \xi} \frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i \\ & \text{s.t. } y_i (w^T \varphi(x_i) + b) \geq 1 - \xi \\ & \quad \xi \geq 0 \end{aligned} \tag{2.8}$$

支持向量机的原理在于把特征向量的 x_i 使用预定义的函数 φ 映射到高维的空间，然后在这个高维的空间中找到一个线性的平面，使得各个类别与这个平面的距离最大。我们把 $K(x_i, x_j) \equiv \varphi(x_i)^T \varphi(x_j)$ 称为核函数。一些常用的核函数如下：

线性：

$$K(x_i, x_j) = x_i^T x_j \tag{2.9}$$

多项式：

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0 \tag{2.10}$$

半径基函数：

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0 \tag{2.11}$$

这里的 γ, r, d 都是核参数。

我们采用了 libsvm 库来求解这一优化问题，训练向量的特征包括三个，分别是曲度、密度和深度。我们选择了半径基函数作为核函数，并设定 $\gamma=1$ 。允许用户定义墙面由哪几

种建筑结构组成，并让用户动态地选取各个类别的样本，在这之后，计算 2.2 小节定义的三种网格属性，并分别归化到[-1, 1]之间，对样本集进行训练，最后进行预测。注意到分类是在 X-Y 平面上进行的，即网格(x,y)表示的是所有 x, y 索引相同，但是 z 索引不同的网格的平均。

图 4、5、6 显示了采用本文提出的算法对建筑立面进行分类得到的结果。可以看出，我们的方法可以很好地对点云进行初步的分类。在分类完成之后，我们便可以得到了某个网格属于特定建筑结构的概率：

$$p(l|x, y) = \begin{cases} 1, & \text{网格}(x, y) \text{ 的类别为 } l \\ 0, & \text{网格}(x, y) \text{ 为其它类别} \end{cases} \quad (2.12)$$

该概率分布是下面章节中基于模型的建模的目标函数的重要组成部分。

第三章 过程建模

过程建模是一种动态的描述复杂的结构化几何实体的方法。一些形状在外观上可能很复杂，但是通过一些基本的操作可以一步步地以一种简单的方式生成。分形、植物和建筑是这方面典型的例子：它们可以被视为由一种简单形状被其它的简单形状所不断递归替代的结果。尽管在这个递归替代的过程中，生成的形状越来越复杂、需要的操作也越来越多，但是在每一步中我们始终是在对简单形状进行操作。

对于过程建模的形式定义称为形状语义。由于本文是对建筑的立面进行建模，由前述知我们把建筑立面看作平面，因此，本章只介绍二维的二分形状语义^[21]，并详细讨论该形状语义在建筑立面建模中的应用。

3.1 二分形状语义

二分形状语义的基本概念是一个带标签的矩形，可以用 5 元组来表示 (c, x, y, w, h) ，其中 c 表示形状的标签（类别）， (x, y, w, h) 定义了矩形在坐标系中的位置和大小。一般写成 $c(x, y, w, h)$ 的形式。下面介绍二分形状语义的其它要素。

形状：一个形状 S 是一系列不相交的带标签的矩形的集合： $S = \{s_1, s_2, \dots, s_n\}$ 。

语义规则：一个语义规则可以在水平或是垂直方向上将一个带标签的矩形拆分为两个不相交的带标签矩形。用如下的记号表示将一个标签为 A 的矩形沿着水平方向(horizontal)剖分为标签为 B 和 C 的矩形：

$$A(x, y, w, h) \xrightarrow{a} B(x, y, a, h) + C(x+a, y, w-a, h) \quad (3.1)$$

其中，其中 \rightarrow 表示沿着水平方向剖分 (\downarrow 表示沿着垂直方向剖分)； $a \leq w$ ；如果 $a = w$ ，那么 C 符号就是空的。给定类别为 A 的形状语义， B 和 C 就可以唯一确定。为了简便，上述式子可以使用下式替换：

$$A \rightarrow B(a)C \quad (3.2)$$

在我们定义的规则中， C 符号不为空的情况下， B 和 C 至少有一个是非终端元素。

二分形状语义：一个二分形状语义 G 是一个 4 元组 (N, T, R, ω) 其中 N 是非终端形状语义的集合，如果一个带标号的矩形 $c(x, y, w, h)$ 出现在一个规则的左边，那么它就是非终端形状语义； T 是终端语义的集合，如果一个带标号的矩形 $c(x, y, w, h)$ 可以不能出现在一个规则的左边，那么它就是终端形状语义； ω 是一个特殊的非终端形状语义，称为初始形状，剖分总是从初始形状开始进行。 R 是一个有限的剖分规则的集合。

二分形状语义产生形状的过程：根据具二分形状语义生成一个形状 S 是一个递归的过程：首先从初始形状 $\{\omega\}$ 开始，在生成过程的每一步，从 S 中选择一个非终端形状 s_i ，从 R 中选择一个可以应用于该形状的规则 r_i ，将 r_i 应用于 s_i 之后， s_i 被剖分为两个子形状，用这

两个子形状取代 s_i , 继续上述操作, 直至 S 只包含终端形状为止。

可以用一个根节点为 ω 的剖分树来表示形状 S 。在生成 S 的过程中, 每一步都位于剖分树的一个结点之上, 该结点的两个子形状被添加到树中作为它的左右孩子。在生成结束的时候, 剖分树的叶子结点就表示了所有的终端形状, 而中间的结点则表示了非终端的形状。

语言: 二分形状语义的语言 $L(G)$ 是一个语义 G 的所有可能的生成方式。

3.2 建筑立面的表示

大部分建筑立面呈现规则的几何形状, 且建筑立面上的结构元素(窗户, 阳台等)的轮廓一般可以用矩形表达, 加之各个元素在建筑表面呈现网格对称分布, 因此建筑立面特别适合用二分语义进行剖分。

建筑立面由结构化的几何体组成, 在定义了形状语义之后进行剖分生成的过程是一致的, 但是不同类型建筑立面差别巨大, 不同风格的建筑立面的二分形状语义是迥然不同的。

表 1 仅包含窗户和墙面的二分形状语义

初始形状	Facade
终端形状	wall, window
非终端形状	facadeWall, facadeFloor, floorWall, floorWindow
剖分规则	facade \downarrow wall + facadeFloor
	facadeWall \downarrow wall + facadeFloor
	facadeFloor \downarrow floorWall + facadeWall
	floorWall \rightarrow wall + floorWindow
	floorWindow \rightarrow window + floorWall

为了展示二分形状语义对建筑立面的表示, 表 1 给出了只包含窗户和墙面的建筑立面的二分形状语义。该语义的初始形状为 facade, 即为建筑立面本身。终端形状只有 wall 和 window, 而非终端形状包括 facadeWall, facadeFloor 以及 floorWall 和 floorWindow。其中 facadeWall 和 facadeFloor 是多层的结构, 前者的最下面一层是只包含墙面的楼层, 后者的最后一层是墙面和窗户交互分布的楼层。而 floorWall 和 floorWindow 是单层的结构, 二者都是墙面和窗户交互分布的楼层, 但是 floorWall 的第一个结构是墙面, 而 floorWindow 的第一个结构是窗户。之所以这么设计非终端形状是为了能够递归地处理多层以及多个窗户和墙面交互分布的情况。如剖分规则 facadeWall \downarrow wall + facadeFloor 所示, facadeWall 可以沿着竖直方向剖分为 Wall 和 facadeFloor, 而规则 facadeFloor \downarrow floorWall + facadeWall 定义 facadeFloor 又可以剖分为 floorWall 和 facadeWall, 只要不断递归执行这两个规则便可

以处理建筑立面上多层分布的情形。同理规则 $\text{floorWall} \rightarrow \text{wall} + \text{floorWindow}$ 和规则 $\text{floorWindow} \rightarrow \text{window} + \text{floorWall}$ 可以处理同一层上窗户和墙面交替分布的情形。图 2 显示了利用该形状语义对一个简单的多层建筑进行剖分得到的剖分树。

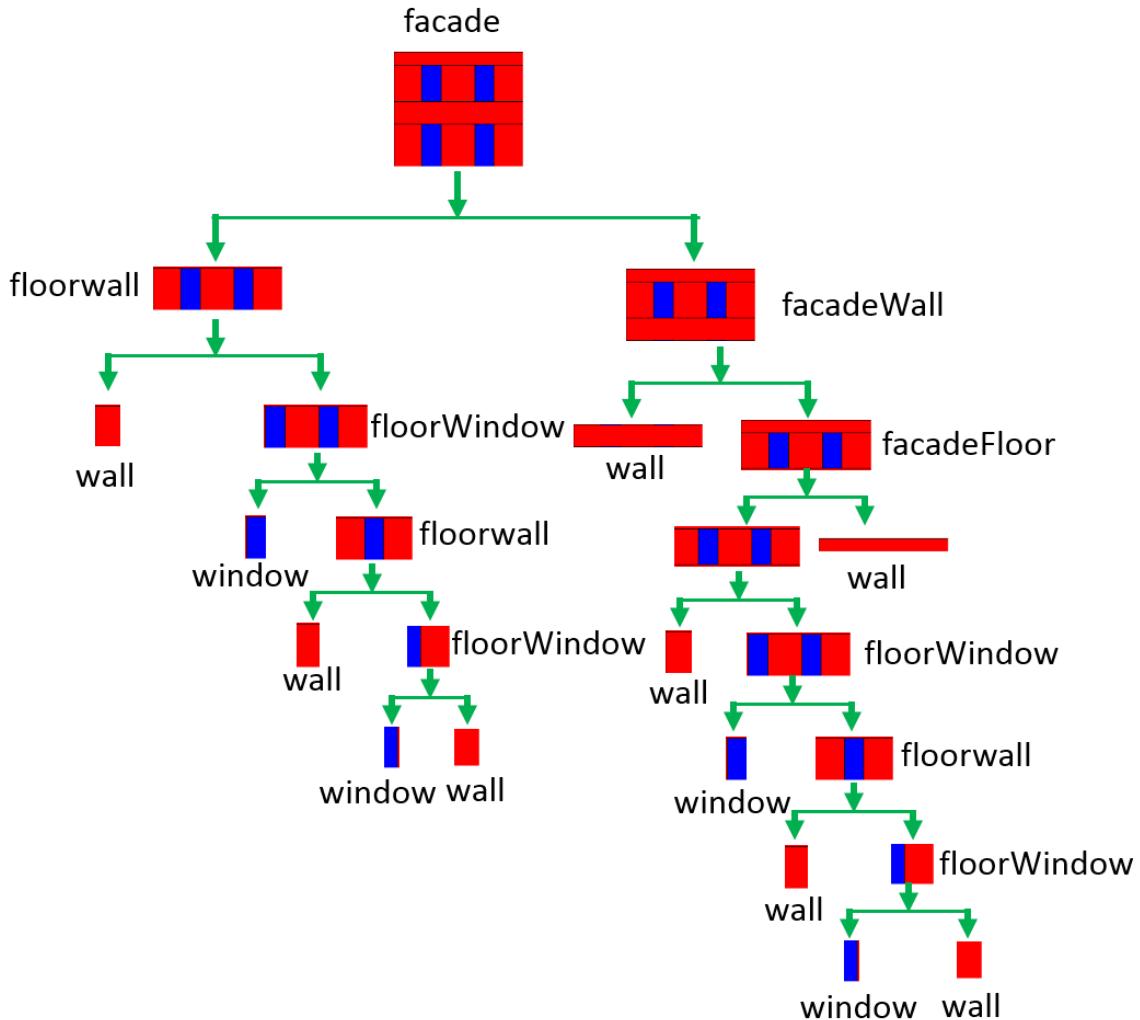


图 2 利用表 1 的二分形状语义对一个简单的建筑立面进行剖分得到的剖分树

对于其它类型的建筑，同样可以设计相应的形状语义对其解译。附录一、附录二给出了具有阳台结构以及教堂的建筑立面的二分形状语义。它们产生的建筑立面可以参见图 11 和图 12。

3.3 立面语义的求解思路

上一节论述了用二分形状语义表达建筑立面结构。只要得到了建筑的剖分树（即二分形状语义规则的应用序列），就可以重建建筑立面。本文目标便是从给定的建筑立面点云中得到表达墙面结构的最佳剖分树。用数学语言描述如下：

给定建筑立面点云 O ，二分形状语义 $G = (N, T, R, \omega)$ ，设 $L(G)$ 为二分形状语义的所

有可能的剖分树（剖分方式）。如何寻找到一个最佳的剖分语言 $s \in L(G)$ ，使得如下的目标函数最大：

$$\arg \max_s \sum_{x,y \in facade} p(s(x,y) | x, y) \quad (3.3)$$

其中， $s(x, y)$ 表示剖分 s 得到的网格 (x, y) 的建筑结构类别； $p(s(x, y) | x, y)$ 即为上一章得到的网格 (x, y) 处建筑类别为 $s(x, y)$ 的概率。从上式可以看出，评价一个剖分的好坏在于它剖分得到的终端形状的类别与我们预分类的结果是否一致。本文采用了逐网格进行累加的方式对建筑立面上的所有网格进行判断，看剖分的结果与预分类的结果是否一样，若一致则进行累加 1，否则累加 0，最终的累加值则反映了该剖分的好坏。最大化式(3.3)便可以得到建筑立面的最佳剖分。

从二分语义的剖分过程来看，可以把求解 (3.3) 式看成是一个马尔科夫决策过程。下一章将详细介绍马尔科夫决策过程，并采用增强学习方法来求解建筑立面的最优形状语义。

第四章 增强学习

这一章将介绍马尔科夫决策模型，以及如何利用增强学习算法来求解。在 4.1 小节，首先介绍马尔科夫决策过程，该过程定义了增强学习模型；接着在 4.2 小节介绍增强学习算法。

4.1 马尔科夫决策过程

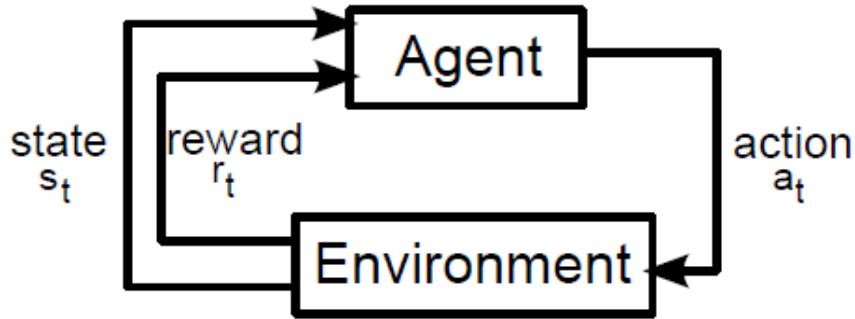


图 3 马尔科夫决策过程各要素之间的联系

定义： 马尔科夫决策过程 (MDP) 由状态集 S , 行为集 A , 以及状态之间的转移概率 P 和在某个状态下执行行为的期望回报 R 组成。该过程的各个组成部分的交互如图 3 所示：在时间 t , 环境(Environment)发送一个状态信号 $s_t \in S$ 给 Agent。根据状态 s_t , agent 做出行为(action) $a_t \in A(s_t)$ 。这个行为反过来影响了环境, 使其状态变为 s_{t+1} , 同时得到一个即时的回报(reward) r_{t+1} 。记 Agent 在状态 s 采取行为 a 使得状态变为 s' 的概率为 $P_{ss'}^a$, 得到的期望回报为 $R_{ss'}^a$, 那么有：

$$P_{ss'}^a = p(s_{t+1} = s' | s_t = s, a_t = a) \quad (4.1)$$

$$R_{ss'}^a = E[r_{t+1} | s_t = s, a_t = a, s_{t+1} = s'] \quad (4.2)$$

马尔科夫决策过程必须满足马尔科夫特性，即状态间的转移概率和即时回报只取决于当前状态和在当前状态下选择的行为，而与过去的状态无关。正是马尔科夫特性使得我们可以使用单步的递归式来表示相邻状态间的关系。

目标函数： Agent 的目标是最大化其长期的回报值。一个马尔科夫决策过程可以是有限步的，也可以是无限的。在有限步长的 MDP 中，我们称从开始的状态到最后的状态的

序列为一个情节 (episode)。Agent 的目标函数定义如下：

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \quad (4.3)$$

其中 $0 < \gamma \leq 1$ 成为折扣因子。对于有限 MDP，设置 $\gamma = 1$ 。

策略：策略 π 用来描述 agent 的行为，解一个 MDP 问题实际上就是寻找一个最优策略。策略 π 表示的是在每个状态 $s \in S$ 下采取行为 $a \in A(s)$ 的概率：

$$\pi(s, a) = p(a | s) \quad (4.4)$$

正如我们之前讨论的，agent 的目标是选择行为使得其长期的回报值最大，因此实际上 agent 是在寻找一个最优的策略来最大化其回报值。为此，要找到一种方法来比较不同策略的优劣。值函数和贝尔曼方程正是用来对策略进行评价的。

值函数和贝尔曼方程：值函数 $V^\pi(s)$ 表示了 agent 从状态 s 开始，遵循策略 π 可以取得的长期回报：

$$V^\pi(s) = E_\pi[R_t | s_t = s] = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\right] \quad (4.5)$$

类似地，可以定义一个行为-值方程 $Q^\pi(s, a)$ 用来表示 agent 从状态 s 开始，采取行为 a ，并遵循策略 π 可以取得的长期回报：

$$Q^\pi(s, a) = E_\pi[R_t | s_t = s, a_t = a] \quad (4.6)$$

通常把上述两个方程称为 V 方程和 Q 方程。显然 V 方程和 Q 方程之间有明显的联系，V 方程可以看成是在状态 s 根据概率 $\pi(s, a)$ 选择行为 a 的 Q 方程的加权和，而 Q 方程可以看成是先在状态 s 下采取行为 a 得到的回报值和 V 方程在下一状态的值之和：

$$V^\pi(s) = \sum_a \pi(s, a) Q^\pi(s, a) \quad (4.7)$$

$$Q^\pi(s, a) = \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma V^\pi(s')] \quad (4.8)$$

结合上面两个式子，得到著名的贝尔曼等式：

$$V^\pi(s) = \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a + \gamma V^\pi(s') \quad (4.9)$$

这个等式是线性的，显示了当前状态 s 和下一状态 s' 之间的联系。对于一个有着 n 个状态的 MDP 过程，只要我们得到所有的 n 个等式，就可以求解出策略 π 。然而在实际上，状态的数量通常十分巨大，因此使用数值迭代的方法来进行策略的评价。这将在下一小节进行介绍。

最优策略：之前我们讨论了如何定量评价策略 π ，因此现在可以比较两个策略的优劣。

我们称策略 π 比策略 π' 更优，如果满足下式：

$$\forall s, V^\pi(s) \geq V^{\pi'}(s) \quad (4.10)$$

因为 MDP 过程满足马尔科夫特性，因此至少存在一个策略会优于其它策略。我们称之为最优策略，用 π^* 表示，那么有：

$$V^*(s) = V^{\pi^*}(s) = \max_{\pi} V^\pi(s), \forall s \quad (4.11)$$

$$Q^\pi(s, a) = Q^{\pi^*}(s, a) = \max_{\pi} Q^\pi(s, a), \forall s, a \quad (4.12)$$

同理，可以得到 V^* 和 Q^π 之间的联系：

$$V^*(s) = \max_a Q^\pi(s, a) \quad (4.13)$$

$$Q^\pi(s, a) = \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma V^*(s')] \quad (4.14)$$

结合上面两个等式，可以得到贝尔曼最优等式：

$$V^*(s) = \max_a \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma V^*(s')] \quad (4.15)$$

$$Q^*(s, a) = \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma \max_{a'} Q^*(s', a')] \quad (4.16)$$

接下去的问题是如何计算出 MDP 的最优解。目前主要有三种类型的方法来求解马尔科夫决策过程：动态规划算法直接使用贝尔曼最优方程组进行迭代。蒙特卡罗方法采用抽样的方法来得到状态集的一个完全的变化轨迹之后才进行迭代。而在它们之间的增强学习算法将蒙特卡罗的抽样方法和动态规划的单步迭代结合到了一起。这三种方法都使用了策略迭代的方式求解最优策略，即从一个初始策略 π_0 开始，不断循环下面两步直到达到特定的阈值为止：1) 根据 V 方程或 Q 方程进行策略评价；2) 根据当前估计的 V 方程或 Q 方程进行策略变优。

4. 2 动态规划

在求解 MDP 的算法中，动态规划是一种保证可以收敛的算法。它使用值迭代的方法来进行策略演化。根据贝尔曼最优方程，当前状态下的最优 V 值可以由下一状态的最优 V 值估计。

$$V_{k+s}(s) = \max_a \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma V_k(s')] \quad (4.17)$$

值迭代的思想是对状态空间进行遍历，不断更新 $V_k(s)$ ，直到全部收敛为止。但是这种方法要求对于环境的转移概率和回报值是已知的，并且由于值迭代一遍遍地对整个状态空间遍历，状态数增加时，算法的复杂性也指数增加。动态规划进行策略演化的算法伪代码

见附录二算法 1。

4.3 蒙特卡罗方法

与动态规划同时更新状态的 V 值不同，蒙特卡罗方法在每次只抽样出一个行为，在抽样完一个情节之后才进行 Q 值的更新，其更新的方程如下：

$$Q_{k+1}(s, a) = Q_{k+1}(s, a) + \frac{1}{n_k(s, a) + 1} [R(s, a) - Q_k(s, a)] \quad (4.18)$$

其中 $n_k(s, a)$ 表示状态-行为对 (s, a) 被抽样的次数。

蒙特卡罗迭代算法的伪代码见附录二算法 2。蒙特卡罗方法克服了动态规划需要知道环境模型以及维数灾难的缺点，随之而来的一个问题是策略的演化问题。在动态规划的值迭代中，并不要对策略 π 进行更新，而蒙特卡罗方法则需要根据当前的 Q 值来更新 π 。在这里我们需要为策略选择一个概率分布模型来决定不同的状态应该选择哪个行为。我们采用了经典的 ϵ greedy 算法，这个算法以概率 ϵ 选择 Q 值最大的行为，而以 $1 - \epsilon$ 的概率随机地选择其它的行为。这种做法可以跳出 Q 值估计错误的情形，能够使得算法跳出极小值，加快收敛。

4.4 增强学习算法

从之前的介绍我们可以知道，动态规划的强大在于使用了单步的值迭代，而蒙特卡罗方法的强大在于其抽样的方法，增强学习则同时结合了动态规划和蒙特卡罗的长处。

增强学习算法的核心思想是时差 (Temporal-Differences)，通过不断迭代 V^π 去估计策略 π 的质量。和动态规划一样，增强学习通过下一状态的 V 值来估计当前状态的 V 值，但是增强学习在估计的时候没有用到下一步的所有状态。和蒙特卡洛方法一样，时差学习次只抽样出一个动作，但是它没有等到整个情节结束后才更新 V 值。增强学习的时差迭代公式如下：

$$V_{k+1}(s) = V_k(s) + \alpha_k [r + \lambda V_k(s') - V_k(s)] \quad (4.19)$$

r 是在状态 s 执行行为 a 后得到的回报值。 $r + \lambda V_k(s')$ 是观测到的 $V^\pi(s)$ 的估计。而 $V_k(s)$ 是当前的估计。 α_k 称为学习比率。一般而言，随着算法的收敛， α_k 不断变小。

时差迭代公式可以用来估计 V^π ，我们通常把它和策略更新的方法结合起来迭代地计算最优的策略。Q-Learning 就是这样一个算法。Q-Learning 实际上和贝尔曼最优方程很接近，它的思想不是根据抽样得来的状态-行为对 (s', a) 来更新策略，而是根据当前的

$\max_a Q(s', a)$ 去更新。它的更新公式为：

$$Q_{k+1}(s, a) = Q_k(s, a) + \alpha_k [r + \gamma \max_{a'} Q_k(s', a') - Q_k(s, a)] \quad (4.20)$$

Q-Learning 算法的伪代码见附录二算法 3。

4.5 半马尔科夫决策过程

在讨论使用增强学习求解最优形状语义之前，我们需要介绍半马尔科夫过程，因为该模型特别适合用于解决分层的马尔科夫过程，而这正是我们求解最优形状语义所需要的。半马尔科夫过程的特点是一些行为持续的时间要比其他的行为长，因此，我们可以添加一个随机变量 τ 用来表示一个行为需要多长时间完成。变量 τ 称为等待时间。这样一来，贝尔曼最优方程就变为了：

$$Q^*(s, a) = R_{ss'}^a + \sum_{s', \tau} \gamma^\tau P(s', \tau | s, a) \max_{a'} Q^*(s', a') \quad (4.21)$$

为了解决半马尔科夫过程问题，动态规划算法使用上式替换值迭代公式即可。对于那些基于 agent 和环境交互的算法，比如蒙特卡罗算法和增强学习算法而言，他们同样可以解决半马尔科夫过程，比如，对于 Q-Learning 而言，只需把对 Q 值估计的方程改为下式即可：

$$Q_{k+1}(s, a) = Q_k(s, a) + \alpha_k [\sum_{i=0}^{\tau-1} \gamma^i r_{t+i} + \gamma^\tau \max_{a'} Q_k(s', a') - Q_k(s, a)] \quad (4.22)$$

半马尔科夫过程的优势在于可以建模包含子任务的过程。对于二分语义的剖分而言，在第一章已经提到，其剖分过程可以用二叉树表示，因此是一个迭代剖分的过程，特别适合于用半马尔科夫模型建模。

第五章 城市建筑立面三维建模

第二章展示了二分形状语义可以有效表达建筑立面结构的生成过程。只要构建建筑立面的二分形状语义剖分树，就可以实现对该建筑立面的解译，因此，现在的问题是如何得到特定建筑立面的最佳形状语义的剖分。本章的 5.1 小节将介绍如何将该问题建模为半马尔科夫过程，并使用增强学习进行求解。已知了立面的形状语义后，5.2 小节介绍如何重建三维立面。

5.1 最优形状语义的求解

我们在 3.3 小节已经给出了该问题的数学定义，为方便讨论，重新叙述如下：

给定建筑立面点云 O ，二分形状语义 $G = (N, T, R, \omega)$ ，设 $L(G)$ 为二分形状语义的所有可能的剖分树（剖分方式）。如何寻找到一个最佳的剖分语言 $s \in L(G)$ ，使得如下的目标函数最大：

$$\arg \max_s \sum_{x, y \in facade} p(s(x, y) | x, y)$$

二分形状语义产生建筑的过程实际上可以用半马尔科夫模型来建模，下面分别介绍马尔科夫模型里的各个要素在建筑剖分中代表的含义。

Agent: 马尔科夫过程中的 Agent 表示的是形状语义剖分建筑立面的过程。在剖分建筑立面的时候，每一步都要处理一个非终端形状，Agent 则决定采用哪种规则及规则参数应用于该形状。

环境: 马尔科夫过程中的环境与 Agent 交互，指的是随着剖分进行不断建立的剖分树。随着剖分的不断进行，剖分树不断完善，环境也就有了对模型更多的知识，这体现在 Q 值的更新中。

状态: 为剖分树中的结点。该结点表示了当前剖分所对应的形状及其位置，用 $state(x, y, symbol)$ 表示，其中 (x, y) 为当前状态所处的位置，symbol 表示当前形状的名字。注意到，symbol 只能是非终端形状。Teoul^[21]采用 $state(pos, symbol)$ 来表示当前状态，其中 pos 表示当前形状在剖分方向的位置（ x 或 y ），虽然这么做大大降低了问题的状态，使得接下来的学习能够较快收敛，但是却无法解决层间不对称的建筑立面的建模问题。本文对状态的编码方式则可以处理立面不对称的情况，随之而来的状态量剧增的问题，我们采用了稀疏矩阵进行解决。

行为: 为 Agent 可以选择的语法规则，用 $action(rule, parameter)$ 表示，其中 rule 表示语法，parameter 表示剖分的宽度。

回报值: 设马尔科夫决策在 t 时刻执行动作 $action(rule, parameter)$ 得到的终端形状为 $A(x, y, w, h, a)$ ，那么相应的回报值为

$$R_t = \sum_{i=x}^{x+w} \sum_{j=y}^{y+h} p(a|x, y) \quad (5.1)$$

其中 $p(a|x, y)$ 表示建筑立面 (x, y) 处是 a 类别的概率。

设 MDP 决策过程的序列为 T , 那么长期的回报为:

$$R = \sum_{t=0}^T R_t \quad (5.2)$$

由于 MDP 决策完毕后, 刚好把建筑立面剖分完毕, 因此有

$$R = \sum_{t=0}^T R_t = \sum_{x, y \in I} p(x, y, s(x, y)) \quad (5.3)$$

所以 MDP 的长期回报值和我们之前建立的目标函数是相等的, 即 MDP 的最优解即是我们要得到的最优形状语义。

我们采用增强学习算法求解以上叙述的 MDP 问题。增强学习采用的是 Q-Learning 算法, 利用增强学习算法求解建筑剖分语义的算法伪代码见附录二算法 4。

5.2 建筑立面深度信息的恢复

增强学习得到的最佳语义可以表示为剖分树的形式, 其中剖分树的叶子节点是组成立面的所有建筑结构 (终端形状), 记为 $Leaf_i(c, x, y, w, h)$, 其中 c 表示叶子节点 i 所属的建筑类别, (x, y, w, h) 定义了建筑结构的外包矩形 R 。通过这些叶子节点, 可以计算各类建筑结构在建筑面上的分布情况, 并构造出建筑该建筑立面的二维平面。

然而还需要得到各种建筑结构的深度以完成最后的三维建模, 为此, 我们查询点云中 R 内的所有点, 利用随机抽样一致模型得到这些点所的平面, 设该平面为:

$$a_i x + b_i y + c_i z + d_i = 0 \quad (5.4)$$

在 2.1.2 小节坐标系转换中, 我们已经求得建筑主平面所在的方程:

$$Ax + By + Cx + D = 0 \quad (5.5)$$

由于建筑结构的平面与主平面平行, 则建筑类别 c 离主平面的距离 (深度) d 为:

$$d = \left| \frac{d_i - D}{\sqrt{A^2 + B^2 + C^2}} \right| \quad (5.6)$$

如此便恢复了建筑结构 c 的深度信息。

第六章 实验验证

6.1 数据集

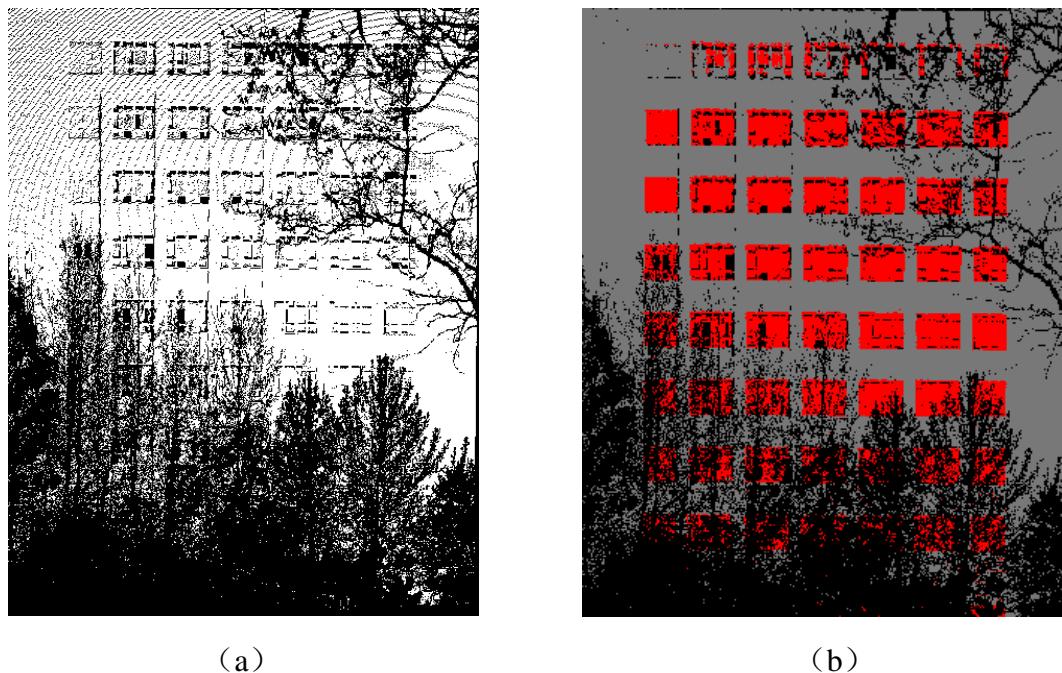
本文实验的数据来自于型号为 RIGEL LMS Z620 的激光雷达扫描仪，扫描精度为 0.02m/点间隔。扫描的建筑来自于中国地质大学（北京）（图 4、7、10）、北京师范大学（图 8、9）以及五道口华清嘉园小区（图 6、11）。为了进一步验证算法的有效性，还从网上下载了 cathedral 教堂的点云¹（图 5、12）。总之，本文采用的数据集包括了对称、不对称结构的多种风格的建筑，并且包含了点云严重缺失的情况，可以较为全面地反映本文提出的算法的有效性。

6.2 建筑立面结构分类结果

图 4、5、6 显示了基于数据驱动的点云分类结果。建筑立面常常被周围的树木遮挡，图 4 显示了因树木遮挡使得立面点云严重缺失的两种建筑结构类别的分类，可以看出，本文设计的三种基于网格的特征值可以有效地在点云大量缺失的情况下正确地分类。并能提取出窗户类别（图中红色部分）隐含的分布模式。

图 5 显示了算法对于复杂建筑的各种结构的分类能力。可以看出，本文提出的分类算法可以正确地识别出突出物（立柱）、门、窗、屋顶这些典型的建筑结构。

图 6 进一步展示了算法对多个立面组成的复杂结构的分类能力。



¹ http://www-sop.inria.fr/members/Florent.Lafarge/benchmark/surface_reconstruction/reconstruction.html

图 4 某点云缺失严重的建筑立面分类结果。(a) 原始点云 (b) 分类后的结果, 红色的为窗户, 灰色的为墙面

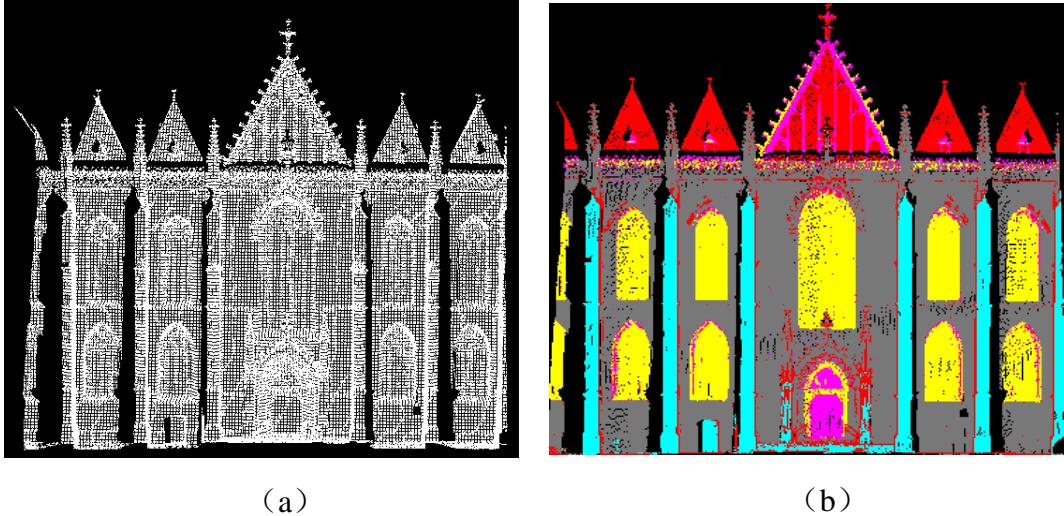


图 5 Cathedral 教堂分类结果。(a)原始点云; (b)分类后的结果。总共有四类, 其中灰色的为墙面, 粉红色的为门, 黄色的为窗户, 红色的为屋顶, 靛青色的为立柱。

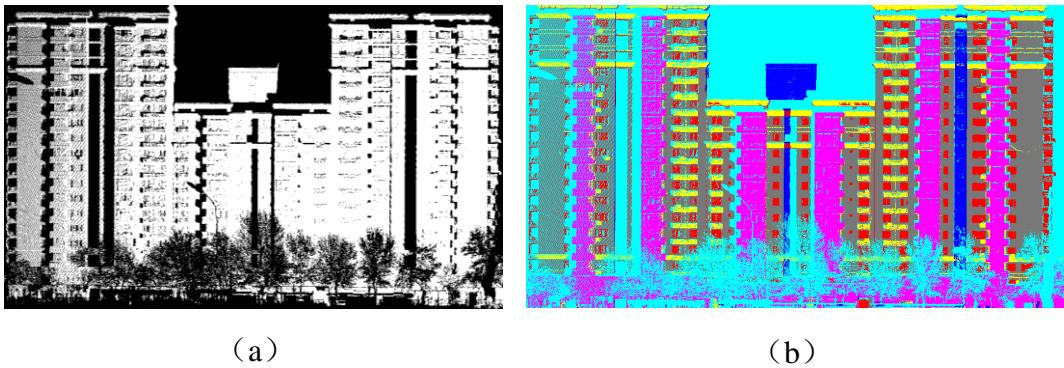


图 6 五道口华清嘉园分类结果。(a)原始点云; (b)分类后的结果。总共有四类, 其中灰色的为墙面, 蓝色的为屋顶, 黄色的为窗户外的小阳台, 红色的为窗户, 粉红色的为突出的大阳台。

6.3 建筑立面解译及重建结果

图 7、8、9 显示了采用表 1 的形状语义对原始点云进行解译的结果。图 7 显示了一个建筑结构数目较少, 但是层间是不对称的立面, 最注意到下面一层的窗户分布明显要不同于其它的楼层。本文方法建模得到的结果很好地体现了原始建筑的结构特征。图 8 图 9 显示了对更为复杂的墙面进行建模的结果, 可以看出, 对于墙面结构众多的情况, 算法也具有很好的表现。图 10 显示了在点云严重缺失的情况下, 本文提出的算法也能很好地对其进行建模。

图 11 显示了采用附录 1 表 1 的形状语义对相应原始点云进行建模的结果。

图 12 显示了采用附录 2 表 2 的形状语义对更复杂的 cathedral 教堂进行建模的结果。

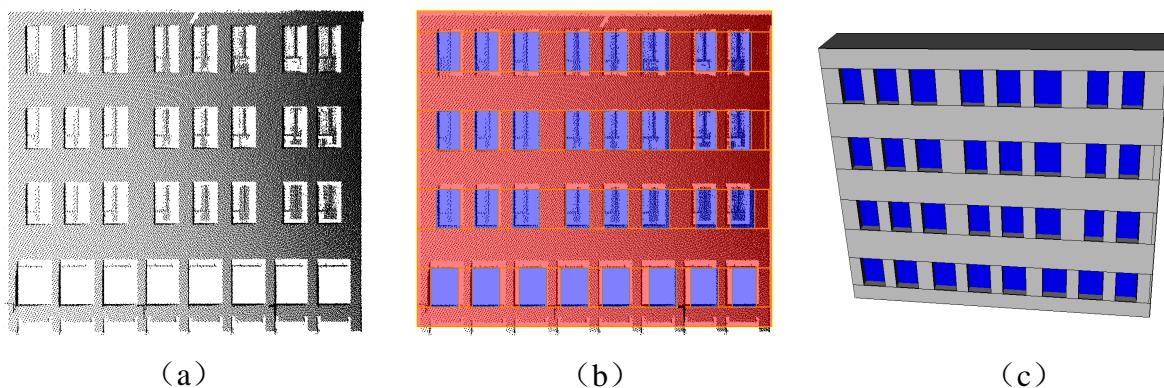


图 7 地质大学图书馆的一个立面建模过程。(a) 原始点云。(b) 形状语义解译结果。(c) 恢复深度建模结果。

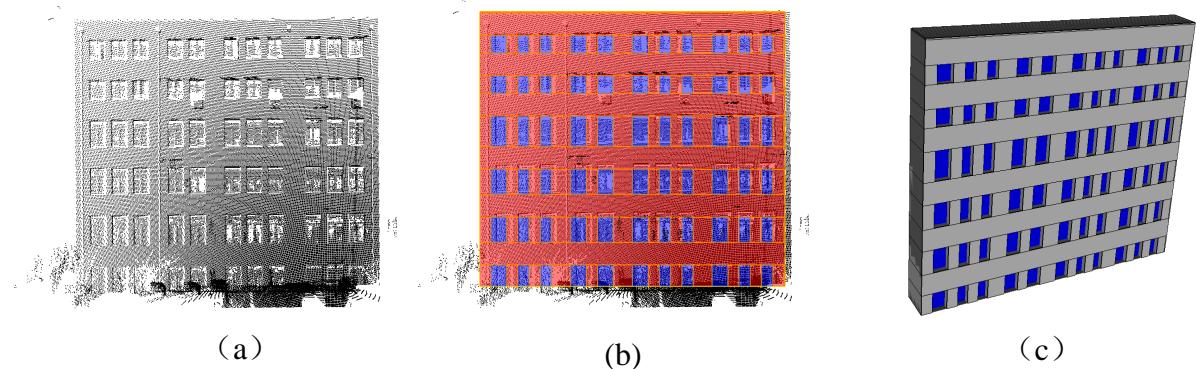


图 8 北师大某建筑建模立面建模过程。(a) 原始点云。(b) 形状语义解译结果。(c) 恢复深度建模结果。

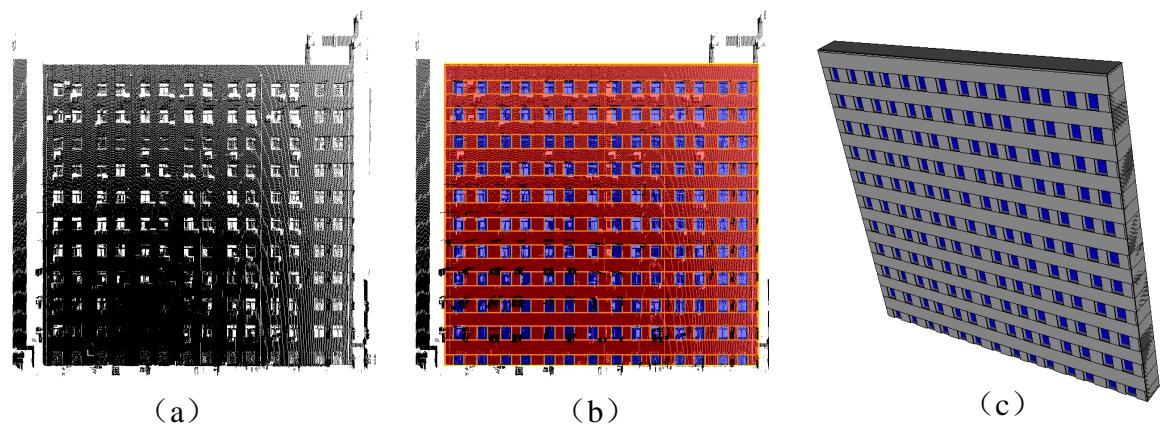


图 9 北师大科技楼立面建模过程。(a) 原始点云。(b) 形状语义解译结果。(c) 恢复深度建模结果。

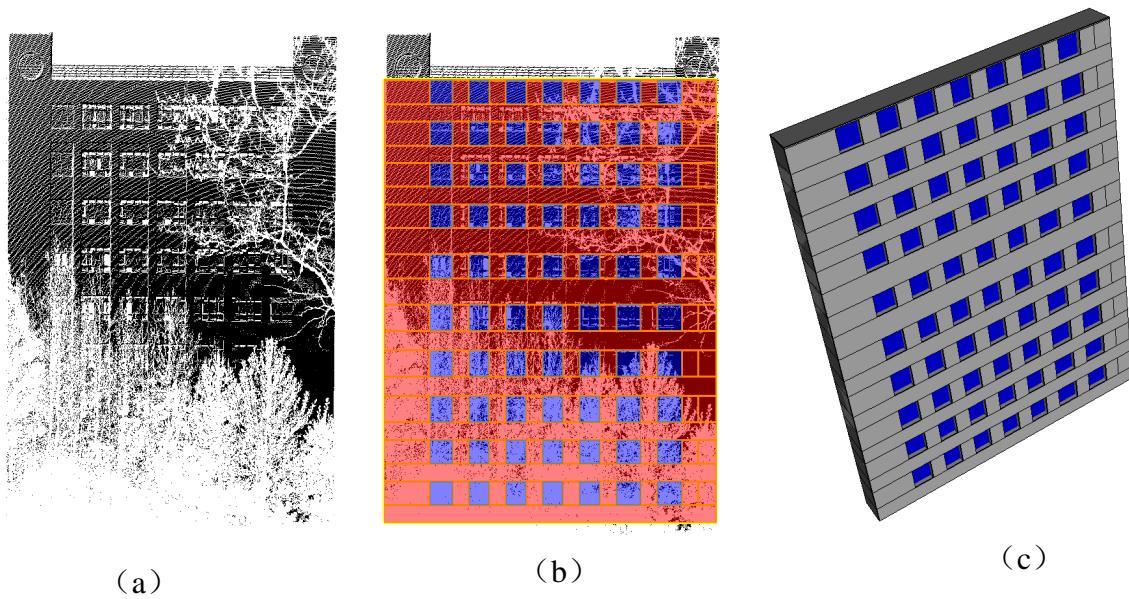


图 10 点云严重缺失的立面建模过程。(a) 原始点云。(b) 形状语义解译结果。(c) 恢复深度建模结果。

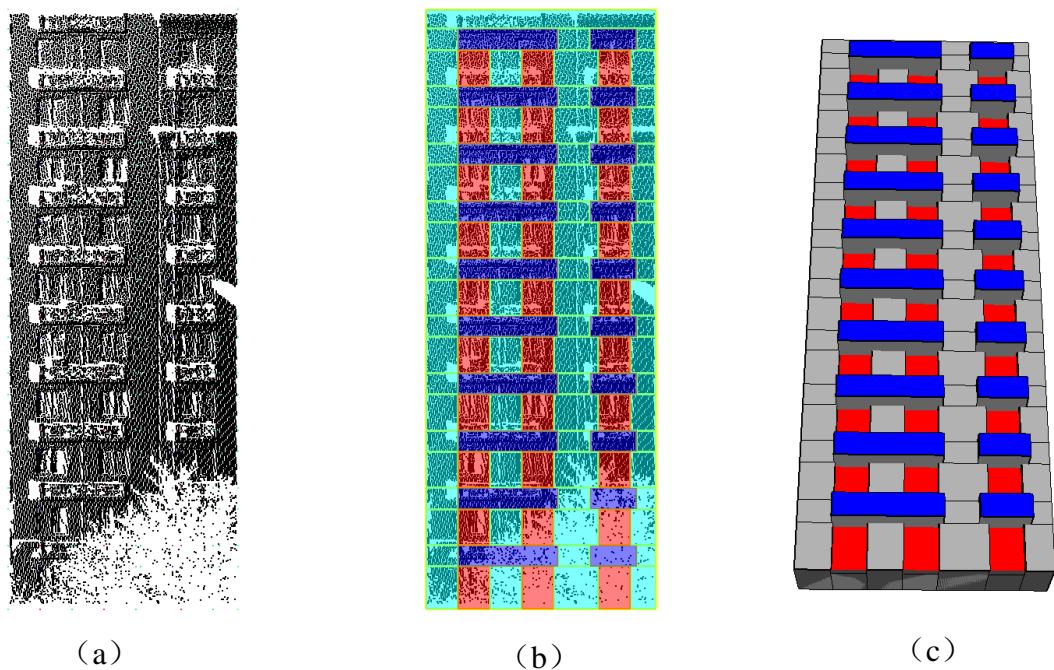


图 11 包含阳台的建筑立面建模过程。(a) 原始点云。(b) 形状语义解译结果。(c) 恢复深度建模结果。

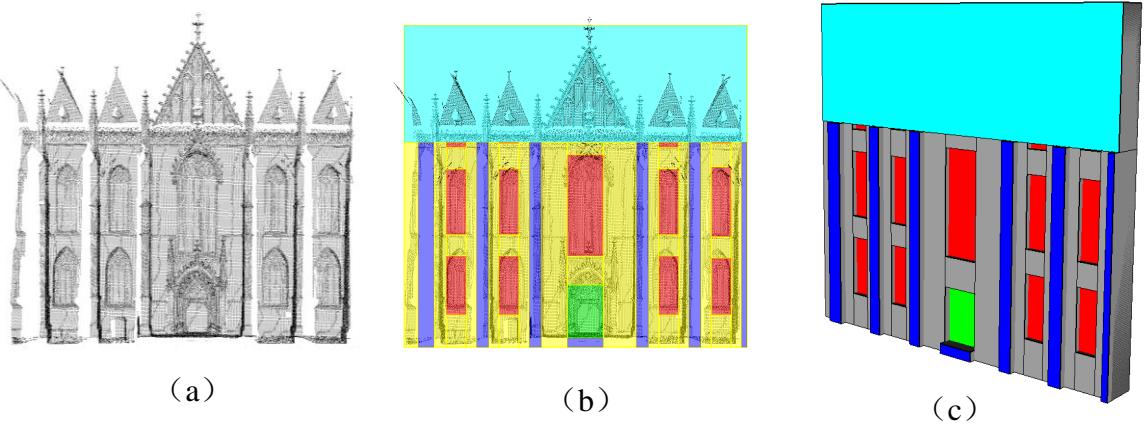


图 12 cathedral 教堂建模过程。(a) 原始点云。(b) 形状语义解译结果。(c) 恢复深度建模结果。

6.4 算法效率

表 2: 实验的效率分析

	支持向量机分类(训练、预测)			增强学习求解	精度(R/N)
	网格属性计算	样本训练	类别预测		
图 7	24.759s	10.609s	1.617s	106.662s	86.252%
图 8	24.541s	9.573	1.478s	271.378s	84.802%
图 9	253.265s	59.707	3.414s	950.347s	86.748%
图 10	161.787s	47.902s	2.759s	623.422s	84.818%
图 11	150.444s	185.403s	1.537s	427.876s	87.648%
图 12	49.338	249.713s	2.315s	831.435s	89.795%

采用 C++ 语言在一台 6G 内存，处理器为 Intel (R) Xeon (R) 的工作站上的实现了点云建筑立面的建模算法并进行了测试。

整个算法分为支持向量机预分类，以及增强学习求解两大部分。在支持向量机预分类中，包括了网格属性计算，样本训练以及类别预测三部分。在网格属性的计算中，由于点云数据量巨大，我们采用了八叉树的方法对点云进行索引，并采用多线程（在试验中采用的是 5 个线程）计算，因此有效地减少了计算的时间；在样本训练中采用了网格法搜索法对两个模型参数进行了搜索，这样可以保证训练的精度（在我们的试验中，精度都达到了 90% 以上），对于同类的建筑，只需训练一次，因此对于大多数建筑只需特征向量计算和类别预测的时间。在增强学习中，图 7-10 的学习次数均设置为 5000 次，图 8 的学习次数为 8000 次。表 5 给出了上述几个实验对应的时间效率。注意到本文进行的测试是在 Debug

模式下进行的，若是采用 Release 模式，效率将会有成倍的提高。

为了定量表示形状语义的剖分结果的精确度，我们将增强学习得到的建筑剖分结果同支持向量机预分类的结果进行比较，回忆 5.1 小节，将增强学习累计回报值为

$$R = \sum_{x,y \in I} p(x, y, s(x, y))$$

R 表示与预分类的结果比较，剖分正确的网格数。设 N 为建筑立面的网格总数，则可以用 R/N 来定量表示增强学习的精度。在表 2 中同样列出了各个实验的精度。

从表 2 可以看出，网格的属性计算和增强学习的求解耗时最多，其中网格属性计算的时间为 24~161s，增强学习的求解时间为 100~1000s 之间。可以看出，本文提出的算法的效率较高，基本可以应对大规模的城市重建。

第七章 结 论

本文结合数据驱动和模型驱动的方法，提出了一个高效的算法对建筑立面进行自动建模。本文的贡献首先在于针对点云建筑立面的特点，设计了基于数据驱动的点云分类方法，结果显示本文提出的分类方法可以很好地区分出建筑面上的不同结构特征。另外，研究了如何使用形状语义来对建筑立面进行解译，设计了适用于各种建筑风格的形状语义，现实世界中的建筑风格各异，但是本文介绍的二分形状语义却可以很好地应对这种情形。最后，将建筑立面的最优语义的求解问题建模成马尔科夫决策模型，讨论了解决马尔科夫模型的几种经典方法，并采用了增强学习算法进行求解，使得建筑立面语义的求解成为可能。

形状语义对建筑立面进行建模的一大优势是可以得到立面建筑学意义上的结构（比如门、窗、阳台等），继而可以进行到立面编辑、立面组合等应用^[23]。虽然 Shen^[30]的方法也可以通过自适应的方法对建筑立面进行剖分，但是无法得到剖分结果建筑学意义上的结构；而与 Teboul 等人^[21-22]的设计形状语义相比，本文在求解最优形状语义的时候对状态定义上没有采用状态集聚的方式，因此可以处理建筑立面不规则的情况，而 Teboul 等人的方法只能处理结构规则的建筑。

未来的研究主要包括两点。首先是拓展现有的二分形状语义到三维空间。本文将建筑立面处理为平面，采用二维的二分形状语义解译，虽然可以在解译后再添加深度信息，但是这种方法却不是一种直接的处理立面建模的方式。最自然的方式是应用三维的形状语义到立面，当然，还是可以利用增强学习的方法求解最佳的三维语义。第二，本文提出的算法在点云的预分类上采用了支持向量机的分类方式，需要用户交互式地选择训练样本，无法实现完全的自动化建模。因此，需要研究自动地进行预分类以使得整个解译和建模过程完全自动化。

参考文献

- [1] Schindler, Konrad, and Joachim Bauer. A model-based method for building reconstruction. Higher-Level Knowledge in 3D Modeling and Motion Analysis, 2003. HLK 2003. First IEEE International Workshop on. IEEE, 2003.
- [2] Brenner, Claus. Towards fully automatic generation of city models. International Archives of Photogrammetry and Remote Sensing 33.B3/1; PART 3 (2000): 84–92.
- [3] Maas H. G. The suitability of airborne laser scanner data for automatic 3D object reconstruction. Automatic Extraction of Man-Made Objects from Aerial and Space Images (III). Ascona, Switzerland: AA Balkema Publishers, 2001: 291–296.
- [4] Vosselman G., Gorte B. G.H., Sithole G., et al. Recognising structure in laser scanner point clouds. International archives of photogrammetry, remote sensing and spatial information sciences, 2004, 46(8): 33–38.
- [5] BECKER, S., AND HAALA, N. Grammar supported façade reconstruction from mobile LIDAR mapping. In Proc. CMRT. Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci, 2009, 229 – 234.
- [6] NING, X., ZHANG, X., AND WANG, Y. Automatic architecture model generation based on object hierarchy. In ACM SIGGRAPH Asia 2010 Research sketches, 2010.
- [7] NAN, L., SHARF, A., ZHANG, H., et al. Smartboxes for interactive urban reconstruction. ACM Trans. Graph, 2010. 29, 4, 93:1 – 93:10.
- [8] ZHENG, Q., SHARF, A., WAN, G., et al. Non-local scan consolidation for 3d urban scenes. ACM Trans. Graph, 2010. 29, 4, 94:1 – 94:9.
- [9] MULLER, P., ZENG, G., WONKA, P., et al. Image-based procedural modeling of facades. ACM Trans. Graph, 2007. 26, 3, 85:1 – 85:10.
- [10]MUSIALSKI, P., RECHEIS, M., MAIERHOFER, S., et al. Tiling of ortho-rectified façade images. In Spring Conference on Computer Graphics, 2010.
- [11]XIAO, J., FANG, T., TAN, P., et al. Image-based facade modeling. ACM Trans. Graph, 2008. 27, 5, 161:1 – 161:10.
- [12]XIAO, J., FANG, T., ZHAO, P., et al. Image-based street-side city modeling. ACM Trans. Graph, 2009. 28, 5, 114:1 – 114:12.
- [13]PARISH, Y. I. H., AND MULLER, P. Procedural modeling of cities. 2001, In Proceedings of ACM SIGGRAPH 2001, ACM Press, E. Fiume, Ed., 301 – 308.
- [14]WONKA, P., WIMMER, M., SILLION, F.. et al. Instant architecture. ACM Transactions on Graphics, 2003, 22, 3, 669 – 677.
- [15]Müller, P., Wonka, P., Haegler, S., et al. Procedural modeling of buildings, ACM, 2006, Vol. 25, No. 3, 614–623.
- [16]Alegre , F and Dellaert, F. A probabilistic approach to the semantic interpretation of building facades. In Workshop on Vision Techniques Applied to the Rehabilitation of City Centres, 2004.
- [17]Feng Han and Song-Chun Zhu. Bottom-up/top-down image parsing by attribute graph grammar. In ICCV, 2005.
- [18]Muller , P., Zeng, G., Wonka, P., et al. Image-based procedural modeling of facades. In SIGGRAPH, 2007.
- [19] Barinova, O., Lempitsky, V., Tretiak ,E., et al. Geometric image parsing in man-made

- environments. In ECCV, 2010. 2273.
- [20] Teboul, O., Simon, L., Koutsourakis , P., et al. Segmentation of building facades using procedural shape prior. In CVPR, 2010.
- [21] Teboul, O., Kokkinos, I., Simon, L., et al. Parsing Facades with Shape Grammars and Reinforcement Learning. IEEE Transactions on Pattern Analysis and Machine Intelligence , 35 (7), 1744–1756, .
- [22] Zhang, Hao, Xu Kai, Wei Jiang Wei, et al. Layered analysis of irregular facades via symmetry maximization. ACM Transactions on Graphics (TOG) 32.4 (2013): 121.
- [23] Riemenschneider, H., Krispel, U., Thaller, W., et al. Irregular lattices for complex shape grammar facade parsing. Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference, 2012, vol., no., pp. 1640, 1647, 16–21.
- [24] Rusu, R. B., Marton, Z. C., Blodow, N., et al. Towards 3D Point Cloud Based Object Maps for Household Environments. Robotics and Autonomous Systems Journal (Special Issue on Semantic Knowledge), 2008.
- [25] Martin A. Fischler and Robert C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. of the ACM, 1981, 24: 381 – 395.
- [26] Pauly, M., Gross, M., Kobbel, L. Efficient simplification of point-sampled surfaces, in: Proceedings of IEEE Visualization, 2002.
- [27] Chang C.C. and Lin C.J.. LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2011, 2:27:1–27:27.
- [28] Shen, Chao-Hui, Huang Shi-Sheng, Fu Hongbo, et al. Adaptive partitioning of urban facades. ACM Transactions on Graphics (TOG) , 2011. Vol. 30. No. 6.

附录 1 几种建筑的二分形状语义

表一：具有阳台、窗户结构的建筑立面形状语义

初始形状	façade
终端形状	wall, window, balcony, shop
非终端形状	facadeBalconyWall, facadeWindowWall, floorWallBalcony, floorBalconyWall, floorWallWindow, floorWindowWall
剖分规则	<pre> facade ↓ shop + facadeBalconyWall facadeBalconyWall ↓ floorWallBalcony + facadeWindowWall floorWallBalcony ↓ wall + floorBalconyWall floorBalconyWall → balcony + floorWallBalcony facadeWindowWall ↓ floorWallWindow + facadeBalconyWall floorWallWindow → wall + floorWindowWall floorWindowWall → window + floorWallWindow </pre>

表二：cathedral 教堂的形状语义

初始形状	façade
终端形状	wall, window, pillar, roof, door
非终端形状	facadeWall, facadeFloor, facadePillar, facadeMiddle, floorWall, floorWindow, floorMiddle, floorDoor
剖分规则	<pre> facade ↓ facadeWall + roof facadeWall ↓ wall + facadePillar facadeWall ↓ wall + facadeFloor facadeWall ↓ wall + facadeMiddle facadePillar ↓ pillar + facadeWall facadePillar ↓ pillar + facadeFloor facadeFloor ↓ floorWall + facadeWall facadeMiddle ↓ floorMiddle + facadeWall floorWall → wall + floorWindow floorWindow → window + floorWall floorMiddle → pillar + floorDoor floorDoor → door + floorWall </pre>

附录 2 算法伪代码

算法 1：动态规划值迭代算法

Initialize randomly $V(s), \forall s \in S$

Repeat

$\Delta \leftarrow 0$

For $s \in S$ do

$V_0 \leftarrow V(s)$

$V(s) \leftarrow \max_a [R_{ss'}^a + \gamma V_k(s')]$

$\Delta \leftarrow \max(\Delta, |V(s) - V_0|)$

End for

Until $\Delta < threshold$

算法 2：蒙特卡罗策略迭代

Initialize randomly π

$Q(s, a) = 0, \forall s, a$

Initialize the number of visits $n(s, a) = 0, \forall s, a$

Loop

Sample an episode from π

For $(s, a) \in episode$ do

$R \leftarrow$ the return from (s, a)

$n(s, a) \leftarrow n(s, a) + 1$

$Q(s, a) \leftarrow Q(s, a) + \frac{1}{n(s, a)} [R - Q(s, a)]$

End for

For $s \in episode$ do

Update π according to $Q(s, .)$

End for

End loop

算法 3: Q-Learning 算法

$$Q(s, a) = 0, \forall s, a$$

Loop

$s \leftarrow$ first state of the episode

repeat

$a \leftarrow$ sample from $\pi(s, .)$

Take action a , observe s', r

$$Q(s, a) = Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$$

$\pi(s, .) \leftarrow \varepsilon - greedy$ with respect to $Q(s, .)$

$s \leftarrow s'$

until $s =$ end of the episode

End loop

算法 4: 利用增强学习算法求解建筑剖分语义的算法伪代码

$$Q(s, a) = 0, \forall s, a$$

$$\varepsilon = \varepsilon_0$$

While $n < T$ do

 Reset environment

$s \leftarrow (\text{Axiom}, 0)$

 update ε

 a sample from $\pi(s, .)$

 Take action a (recursively if subtask), observe s', r

$$Q(s, a) = Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$$

$\pi(s, .) \leftarrow \varepsilon - greedy$ with respect to $Q(s, .)$

$s \leftarrow s'$

 repeat

 until $s =$ end of the episode

End while

致 谢

时光匆匆，转眼间四年的大学生活就要结束了。现在回想，这一路走来，有太多需要感谢的人，正是他们陪伴、指导着我一步步向前，在学术和为人处事上不断提升。首先我要感谢的人是张立强教授，从大二开始我就在张立强教授的指导下进行学术研究。张老师的严谨、认真以及对于前沿科技的敏感一直感染着我。正是他的严格要求督促着我不满现有的研究成果，不断深入，最终我在老师的指导下完成了本科生基金项目，并且论文被国际期刊接收。在进行毕业论文写作的时候，我正在学校外实习，是张老师对我不厌其烦的指导、督促才使我有了完成论文的动力。我在张老师那里学到的不仅是科研的方法，还有做科研的态度和精神，这在我以后的学习和工作中都将受用无穷！

能够完成毕业论文，除了张老师的辛勤指导外，还要感谢邓浩博士对我的编程方面的指导、他深厚的数学知识也使我受益匪浅；感谢李竺强硕士采集了本文实验用的点云数据；感谢易智瑞（北京）软件研发中心有限公司为我提供了研究的环境以及实验用的计算机等，除了硬件的帮助外，同事吴泳锋给了我很大帮助，使得我能够将实际软件开发中的一些思想和技术应用到毕业设计中来；感谢和我一起在公司里实习的熊婷同学常陪我一起在公司加班做毕业论文，你们鼓励和期待给了我不断研究的动力。

作为一个地遥人，大学四年我是自豪的。感谢学院为我们提供的学习平台、感谢所有老师对我的辛勤指导，在地遥学院四年快乐而充实的经历将成为我的生命里一段美好珍贵的回忆！感谢我的室友，大家从全国各地聚集到一个宿舍，在这里我们一起生活了四年，感谢大家的包容和相互帮助、相互支持！感谢 2010 级所有同学的陪伴，能与大家相遇是一件美好的事情，大家一起上课、一起出野外的情景将成为我弥足珍贵的记忆！

最后，要感谢我的父母和姐姐，是你们一路支持着我走到现在，感谢你们对我的无微不至的关心和长久的信任，谢谢你们！

徐翔

2014 年 5 月