

文章编号: 2096-1618(2025)01-0042-07

# 一种基于视差映射的建筑立面建模方法

刘应许, 何晓曦, 孟繁林, 李茹濡, 朱群

(成都信息工程大学软件工程学院, 四川 成都 610225)

**摘要:**现代城市建筑表面结构复杂,对其进行精细化建模非常困难,如何快速对建筑表面进行重建成为建筑建模领域的研究重点。提出一种基于视差映射的建筑表面建模方法,提供一套可以对建筑表面外观进行表达的建筑表面组件 2D 基元库,通过提取基元库中基元模拟建筑表面细节,并根据基元的组合提取所需要的贴图数据,使用基于视差映射的方式对建筑表面添加细节,在视差映射过程中使用松散锥阶图对视差映射的 RayMatching 过程进行加速,最终结果可控可编辑。实验结果表明,该方法可以高效生成建筑模型,提供足够表面细节,快速修改建模结果,并且建模过程对非专业用户友好。

**关键词:**视差映射;锥阶映射;程序化生成;位移贴图;表面重建

**中图分类号:**TP312

**文献标志码:**A

**doi:**10.16836/j.cnki.jcuit.2025.01.007

## 0 引言

在计算机视觉和计算机图形学领域,建筑物三维模型的重建一直是研究的热点,该研究方向在智慧城市建设、虚拟现实与文化遗址数字化保护等领域有着广泛的应用。过去的二十年,建筑建模技术得到长足的发展,从计算机图形学的角度来看,建模主要是对建筑主体的建模和模型的渲染,建筑建模不仅要考虑单个建筑模型,也要考虑到模型渲染效率,对于大规模场景的建模主要面对的就是渲染效率和建模效率两个核心问题,具体在于:如何对建筑组件进行分解与复用,并且合理定义建筑组件之间的联系,加快建模效率;如何优化模型的渲染效率,否则无法获得期望的建筑渲染效果<sup>[1]</sup>。

为克服以上问题,本文提出一种基于图像基元的纹理建模方法,将建筑表面物体拆分为不同的 2D 基元,通过基元的不同组合来模拟建筑的表面,并且使用优化后的视差映射将模拟出的表面添加到建筑实体网格。让用户能够实时低成本地修改建筑模型,快速对带有复杂表面细节的建筑进行建模和渲染,从而更好地满足城市数字化的需求。

## 1 相关工作

### 1.1 建筑表面建模技术

目前建筑表面建模技术大致分为 3 类:基于规则

的建筑表面建模,基于图像的建筑立面生成以及基于点云的表面重建技术。

基于规则的建模:通过定义参数和对应的规则来生成需要的建筑模型。为生成建筑外立面,Wonka 等<sup>[2]</sup>引入了拆分语法的概念,一种形式化的上下文无关语法,用于生成构建模型,拆分语法和 L-system 相似,其中的基本元素是形状而不是符号。Müller 等<sup>[3]</sup>在此基础上发展出了形状语法 CGA,和拆分语法不同,形状语法使用上下文相关规则,允许对元素进行拆分和旋转,基于 CGA 的建模系统可以根据用户编写的规则自动生成各种模型,并且后续拓展成为上下文敏感的 CGA++ 语法语言。Finkenzeller<sup>[4]</sup>概述了现有的立面建模技术,认为应该注意到每个形状在完整建筑中的语义信息,建议在风格化的规则中注意这些语义信息,以便生成不同风格的建筑外立面,并将其应用于相同的建筑轮廓。上述方法都能生成足够复杂的建筑,但是描述复杂对象的规则是具有挑战性的、不直观的,需要操作人员对该类语法语言有深入的了解,不适合没有编程经验的人。

基于图像的建模:通过摄影测量技术来恢复图像中的建筑结构。图像来源于卫星摄影或者街景数据,这类方法可以根据所提供建筑物图像的数量分为单幅图像建模和多幅图像建模。在单幅图像建模中,Guil-lou 等<sup>[5]</sup>使用相机消失点的方式完成建筑建模;Li 等<sup>[6]</sup>提出一种使用基于自注意力机制的方法,从单幅图像中重建三维体素;Sharma 等<sup>[7]</sup>提出一种根据三维场景中的对称和仿射重复点进行三维建模的技术;缪

永伟等<sup>[8]</sup>提出一种基于长方体和圆柱几何体几何基元定标的单幅图像建模方法。但是上述基于单幅图像的方法存在较多局限性,主要体现在目标建筑需要具有高度对称性,否则难以构建出图形之外的结构,单幅图像对原始图片的要求过高,需要特定的拍摄角度,并且单幅图像难以记录深度信息。基于多幅图像的建模方式,Debever等<sup>[9]</sup>提出通过用户交互的方式来标定建筑轮廓;Dick等<sup>[10]</sup>通过对领域专家的知识训练来识别建筑组件进行建模。目前随着计算机视觉的发展,基于运动恢复结构的重建方法非常受欢迎<sup>[11]</sup>,但是多幅图像的方法存在对数据准确度要求高,难以构建建筑局部细节的局限性。

基于点云的建模:这类数据来源于激光雷达扫描,激光雷达能够获取建筑表面的密集点的三维坐标、反射率及纹理信息。AriKan等<sup>[12]</sup>提出一种基于优化追踪算法实现的建筑物建模方法;Nan等<sup>[13]</sup>提出一种交互式工具,让用户能够直接在车载LiDAR扫描的大规模城市点云数据上进行建筑建模工作;张文元等<sup>[14]</sup>提出一种基于3D基元拟合的复杂屋顶点云自动重建方法,结合点云数据以及城市地理标记语言生成建筑模型。但是以上方式采集一次数据需要耗费大量时间,并且数据采集后存在大量的噪声与缺失,数据的有效性较低,在数据处理层面增加较多工作量。

本文解决了目前建模工具中交互性差和建模周期长的问题,提出一种不同于以上3类表面重建技术的建模方式,建筑表面细节通过一系列2D基元的组合而成,再使用视差映射将细节添加到模型表面,用户可以实时调整2D基元的位置来达到修改模型的目的,降低建模的难度,加快模型迭代的速度。

## 1.2 表面细节增强技术

由于物体表面几何细节的存在,让光照变化在渲染中无处不在,最简单方式是使用曲面细分来生成大量三角形网格,但是由于同一时间硬件渲染的三角形数量存在限制,导致无法渲染出模型所有的细节。

Blinn等<sup>[15]</sup>提出扰动表面法线,在不增加模型实际几何复杂度的情况下,实现对模型复杂几何细节的模拟,这种扰动是在粗糙表面的高度图中进行计算,在一定程度上还原了实际表面模型。

对模型表面几何点偏移的算法也陆续被提出,Szirmay等<sup>[16]</sup>对位移贴图的技术进行阐述,但是想要得到最优的位移贴图渲染结果需要很高的曲面细分。视差贴图解决了这个问题,不会对模型实际几何顶点产

生影响,也能有表面细节的呈现,视差贴图使用Ray-Matching计算观察点和高度场的交点,产生正确的渲染效果,在扰动表面法线的基础上增加了真实感。

虽然位移贴图的效果已经得到了极大的提升,但由于使用的是近似算法,会产生不稳定的渲染结果,物体表面细节效果也会根据当前观测点的变化而变化。Dummer<sup>[17]</sup>提出一种锥阶映射的可靠算法,它能够保证再找到视线与高度场的第一个交点,并且能够跳过在RayMatching过程中大量的空白区域。但是锥阶映射需要对锥阶映射图进行预计算,会花费大量的时间。本文提出一种针对锥阶映射图的加速处理方法,能够加快锥度图的生成速度。

## 2 基于图像基元的纹理建模方法流程

本文提出的基于图像基元的纹理建模方法首先通过输入的建筑底面轮廓点进行建筑轮廓生成,然后输入建筑的高度属性、屋顶属性对建筑的主体结构进行构建,建筑主体模型构建完成后,使用提供的图像基元对建筑表面细节进行组合,最终通过位移映射技术对建筑表面细节进行渲染,完成对整个建筑的建模。这种方式的建模在一定程度上减少了工作量,并且大大缩短了建模迭代的周期和减少用户交互的操作。其方法流程如图1所示。

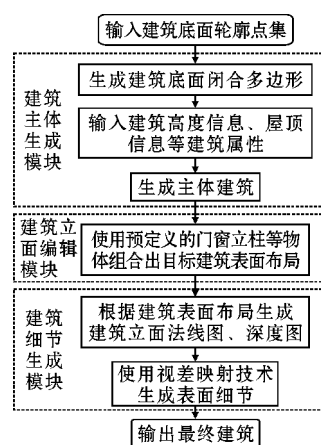


图1 方法流程图

本文方法主要包含如下步骤:

(1) 基于建筑配置属性的建筑主体模型生成,根据用户输入的空间坐标以及更多建筑属性进行建筑低模的生成。

(2) 基于松散锥阶图加速的视差映射的建筑表面细节模拟。为给物体表面提供可视化细节,使用视差贴图来为建模表面进行渲染。视差贴图的真实感来源

于其 RayMatching 迭代中的最大迭代次数,本文采用松散锥阶映射算法<sup>[18]</sup>对 RayMarching 进行加速。松散锥阶映射是一种结合锥阶映射和二分搜索优点的光线步进求交算法,采用积极的空间跳跃代替传统的线性搜索,并且在抵达目标点附近后立即进行二分搜索查找相交点。根据编辑后的建筑表面布局,通过拍摄提取其深度信息,生成对应的深度贴图,并且计算出所需要的锥阶映射图,进行足够逼真的表面几何细节渲染。

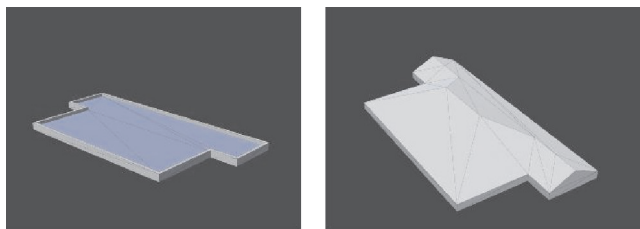
### 3 基于底面轮廓的建筑主体生成

本文用于输入的数据结构被定义为以下4个部分:由建筑底面轮廓点构成的封闭多边形;建筑楼层高度;建筑层数;建筑屋顶类型。

在配置文件中预定义的常见建筑轮廓类型有矩形、T形和I形3种,如图2所示,可以在导入配置后进行一定程度上的手工编辑来生成更多的变体,以实现建筑物主体形状的多样性。预定义了两种常见屋顶类型:平屋顶和坡屋顶,见图3。



图2 预定义建筑轮廓



(a) 平屋顶

(b) 坡屋顶

图3 屋顶示意图

通过建筑地面轮廓和输入数据中的建筑楼层高度与建筑层数,可以推理出建筑立面的 Mesh 结构,在建筑立面的 Mesh 基础上,使用预定义的屋顶类型进行建筑封顶操作。

## 4 松散锥阶加速视差映射

### 4.1 建筑立面基元库

首先对建筑表面基元库进行构建,建筑表面基元库包括窗户、雕花、墙面立柱等在建筑外立面常见的结构,

见图4。其中黑色表示透明的区域,白色表示凸起的区域,使用白色区域的亮度来定义该区域的凸起程度。

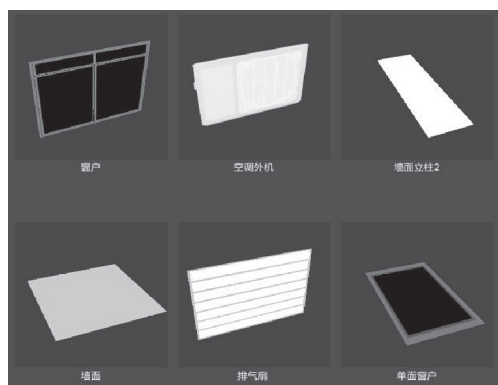


图4 部分建筑立面组件

### 4.2 建筑表面贴图计算

使用预定义好的基元库对建筑表面细节进行组合,如图5(a),使用正交相机对建筑表面几何约束进行拍摄,记录下表面约束中的深度信息和法线信息,使用正交相机直接生成当前墙面的深度图、法线图。使用中心差分法进行计算,如图5所示。

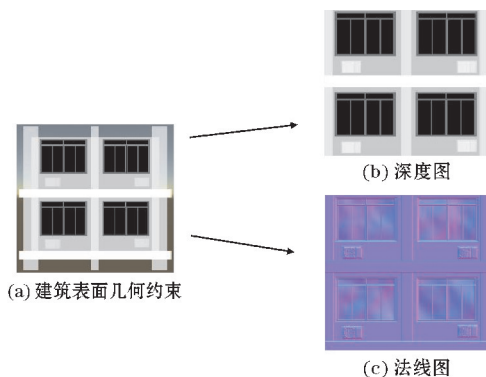


图5 根据几何约束生成的深度图和法线图

图5(a)是编辑好的建筑表面约束,图5(b)为使用正交相机拍摄生成的深度图,图5(c)为使用正交相机拍摄生成的法线图。其中某一点深度值由该点的颜色值确定,深度图中黑色部分表示深度为1,白色部分表示高度为0,其他颜色表示该位置深度属于0~1,深度图记录了物体表面的细节的凹凸关系,也是下文中的松散锥阶图计算的基础。

### 4.3 加速的松散锥阶图计算

要使用松散锥阶映射,需要在上文中生成的深度图基础上进行计算,生成松散锥阶图,松散锥阶图的R通道存储当前纹理的深度值,G通道存储当前纹理对应的最小的锥度,图6为某一对( $t_i$  原始纹理,  $t_j$  目标纹理)对应的锥度  $C_i$  的计算。



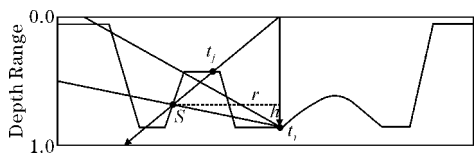


图6 锥度计算示意图

$t_i$  为视线与高度图屏幕的交点的采样纹理值,对每一个原始纹素  $t_i$ ,都有一条以  $(t_i, \text{texCoord. } x, t_i, \text{texCoord. } y, 0)$  为起点,并且指向  $(t_j, \text{texCoord. } x, t_j, \text{texCoord. } y, t_j, \text{texCoord. } \text{depth})$  的射线 Ray,对于每一条这样的射线 Ray,都计算其与高度图的第二个交点  $s$ , $s$  的计算公式如下:

$$s = t_i + \left( \frac{\sqrt{2}}{\text{searchStep}} \right) \cdot \text{RayDirection} \quad (1)$$

searchStep 搜索步长通常取值为 128,当  $s$  的深度值小于当前点的采样深度值时则停止计算,使用该交点来计算锥度值,锥度值的计算公式为

$$C_i = \begin{cases} \frac{\|s, xy - t_i, xy\|}{s, \text{depth} - t_i, \text{depth}}, & s, \text{depth} < t_i, \text{depth} \\ 1.0, & s, \text{depth} \geq t_i, \text{depth} \end{cases} \quad (2)$$

当  $s$  的深度值超过原始纹素  $t_i$  时,被认定为穿过了平面,此时将锥度值设定为 1.0,  $C_i$  的最终值为遍历完所有  $t_j$  后计算出的最小值,见图 4 中绿色虚线。

由于锥度图的计算需要对所有的纹素进行遍历采样,CPU 在每一次对深度图采样时都会带来等待的时间,少量的等待时间在大量的循环中被大量叠加,极大降低了算法效率。因此,在对所有纹素进行查找前,对深度图进行预处理,首先对深度图进行一次采样,将采样出来的深度值存储到 GPU 缓存中,之后在每一次计算中需要深度值时,可以直接读取 GPU 缓存中的数据。在读取当前点  $t_j$  的深度值后,若  $t_j$  的深度值大于  $t_i$  的深度值,可以直接跳过该次循环,此时的  $t_j$  已经位于了  $t_i$  的下方,永远不会被采样到。当所有纹素  $t_i$  计算完成后,得到所需的松散锥阶图(图 7),使用 R 通道存储深度值,G 通道存储锥度值。

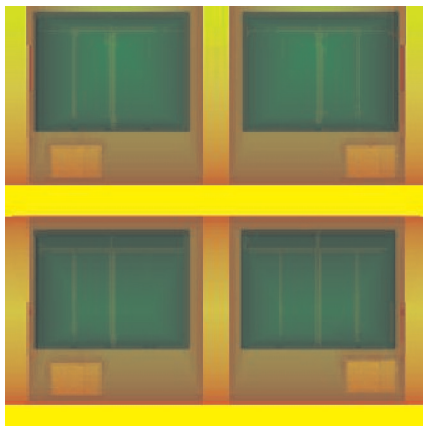


图7 基于高度图生成的松散锥阶图

#### 4.4 使用松散锥阶图进行建筑表面渲染

锥阶映射的渲染方式和传统视差映射类似,通过对计算模型表面的 UV 偏移量来渲染出正确的纹理,优势在于使用预处理好的松散锥阶图对 RayMatching 过程进行加速。首先计算在切线空间内的视线方向 RayView,计算公式如下:

$$\text{RayView} = \text{TBNMatrix} \cdot (\text{CameraPosition} - \text{VertexPosition}) \quad (3)$$

$$\text{RayView} = \frac{\text{RayView}}{\text{abs}(\text{RayView. } z)} \quad (4)$$

使用 TBNMatrix 进行切线空间的转换,在转换完成后使用 RayView 的  $z$  值作为缩放因子对其进行缩放,防止后续渲染效果过于陡峭。

使用原始 UV 对锥阶图进行采样,如图 8 中的点  $k$ ,得到在  $k$  位置的锥度值  $C_k$  和深度值 CurrentHeight,并且定义当前光线的锥度 RayRatio 为:

$$\text{RayRatio} = \|\text{RayView. } xy\| \quad (5)$$

将  $k$  作为起始点开始步进,当前点的位置定义为 CurrentRayPosition,下一次的步进点 NextRayPosition 的计算公式如下:

$$\text{NextRayPosition} = \text{CurrentRayPosition} + (\text{StepLength} \cdot \text{RayView}) \quad (6)$$

$$\text{StepLength} = \frac{C_k \cdot (\text{CurrentHeight} - \text{RayHeight})}{\text{RayRatio} + C_k} \quad (7)$$

RayHeight 为当前点位置的深度,默认从 0 开始。每步进一次就对 CurrentRayPosition 进行一次采样,当采样出的深度值小于当前位置 CurrentRayPosition 的深度值时,说明当前点已经穿透深度图,此时停止步进,将当前位置的点记录为 BSPosition 和上一步的位置进行二分搜索,查找出 Ray 和深度图的准确交点  $I$ ,在二分搜索时,每一次都将搜索范围缩小一半,点  $I$  的计算公式为

$$\text{BSPosition} = \begin{cases} \text{BSPosition} - \text{StepLength} \cdot \text{RayView}, & \delta > 0 \\ \text{BSPosition} + \text{StepLength} \cdot \text{RayView}, & \delta < 0 \end{cases} \quad (8)$$

$$\delta = \text{BSPosition. Depth} - \text{RealDepth} \quad (9)$$

其中 RealDepth 为使用 BSPosition.  $xy$  在锥阶图中采样出的真实深度值。

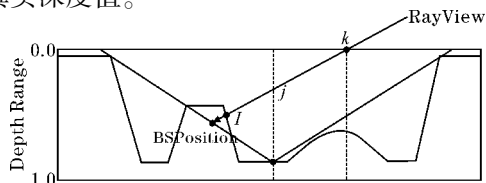


图8 计算纹理示意图

最终输出渲染结果如图9所示。



图9 单个建筑立面最终渲染结果

5 实验以及结果

本方法在 3. 20GHz AMD Ryzen (TM) PC 机上的 Unity 平台利用 C#进行编程实现。利用 Unity 软件进行生成三维模型后后续渲染。在重建过程中,用户只需要对输入的建筑底部轮廓进行编辑,以及对立面约束进行调整,实现实时高效的建模效果。在建模过程中,用户无需专业的建模知识,只需要使用鼠标进行简单交互即可快速生成三维建筑模型。图10为基于倾斜摄影图片的建筑建模结果。



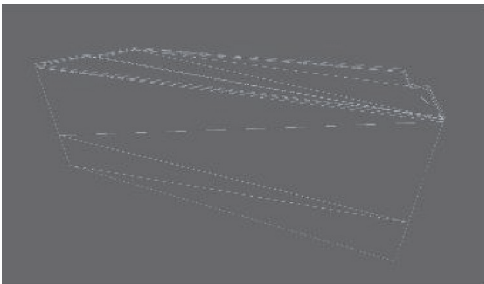
(a) 倾斜摄影图片



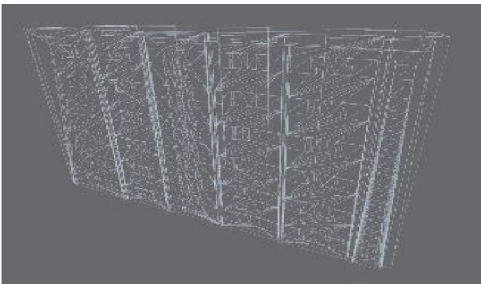
(b) 建模结果

图10 建筑重建结果

经过实验发现,本方法可以在提供足够建筑表面细节的情况下,保持尽量少的三角形数,见图11的模型线框对比图。



(a) 本文方法建模结果



(b) 3DMax 建模结果

图11 模型线框对比

表1统计了图11中模型的顶点数量和三角形数量。在大规模测试场景中,见图12,三角形数量达到170 k,渲染场景所产生的 GPU 调用 Batches 为313个。

表1 模型各项数据对比

形状	本方法	3DMax
三角形	86	1022
顶点	168	1355

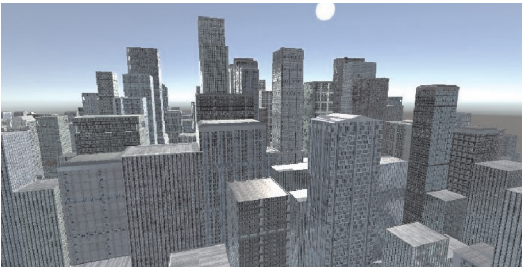
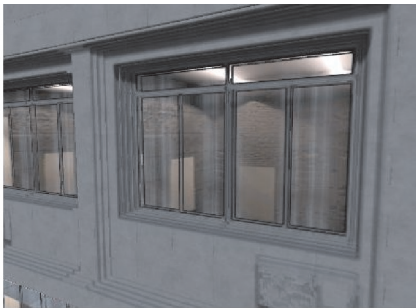


图12 城市场景

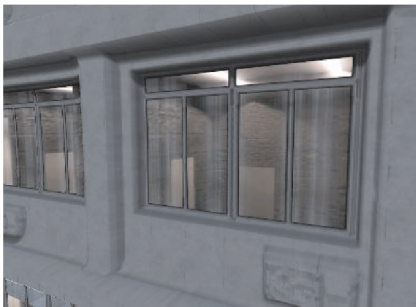
图13为在不同采样次数时,视差映射和松散锥阶映射细节对比图。

可以看出在采样次数较低时,视差映射有明显的分层现象,而松散锥阶映射在同样的采样次数时效果较好。

表2给出了传统视差映射和松散锥阶映射在渲染大规模场景时的性能对比数据。从平均帧率可以看出,松散锥阶映射在锥度图的生成速度方面有更好的性能,采用不同分辨率的深度图进行测试,表3为不同分辨率下锥度图的生成速度。



(a) 视差映射 采样次数为 8



(b) 松散锥阶映射 采样次数为 8

图 13 建筑细节对比图

表 2 不同采样次数的对比实验(平均帧率)

采样数	视差映射	松散锥阶映射
8	70	80
18	57	73
28	48	68

表 3 不同分辨率下的锥度图生成速度 单位/s

方法	64×64	256×256	1024×1024
未加速的生成方法	5.4	10.3	26
本文加速后的生成方法	2.8	3.4	4

6 结束语

提出一种基于纹理信息的建模方法,使用位移贴图对建筑表面进行渲染,针对不同建筑外立面的构成,拆分成可以进行复用的建筑立面组件库,只需要使用组件对建筑立面进行组合,即可完成建筑建模工作。实验表明,本方法不需要用户有专业的建模知识,只需要进行少量的交互就能完成建筑建模工作,可以实时修改,建模过程简单,建模效率高和实用性强,可以满足实际需求中的大部分建筑建模需求。后续工作可以加入为单个建筑立面指定特定渲染着色器;提供用户自定义建筑组件库的接口;接入 OSM 数据自动生成建筑模型;读取图像信息自动生成建筑表面约束的功能。

参考文献:

[1] 李尚林,谢文军,李琳,等. 计算机快速建筑建模方法综述[J]. 计算机学报,2019, 42(9):1966–1990.

[2] Wonka P, Wimmer M, Sillion F, et al. Instant Architecture [J]. ACM Trans. Graph. , 2003, 22 (3) :669–677.

[3] Müller P, Wonka P, Haegler S, et al. Procedural Modeling of Buildings [J]. ACM Trans. Graph. , 2006, 25(3) :614–623.

[4] Finkenzeller D. Modellierung komplexer Gebäudefassaden in der Computergraphik [M]. KITSscientific Publishing,2008:1–10.

[5] Guillou E, Meneveaux D, Maisel E, et al. Using vanishing points for camera calibration and coarse 3D reconstruction from a single image [J]. The Visual Computer,2000,16(7) :396–410.

[6] Li X,Kuang P. 3D-VRVT: 3D Voxel Reconstruction from A Single Image with Vision Transformer [C]. ICCST,2021 :343–348.

[7] Sharma G, Chaudhury S, Srivastava J B. On Exploiting Affine Repetitions for 3D Reconstruction from a Single Image [J]. ICVGIP’ 08, 2008, 12 (1) :489–496.

[8] 缪永伟,汪逊. 基于几何基元定标的单幅图像组合式建筑建模[J]. 浙江工业大学学报,2019, 47:611–617+647.

[9] Debevec P,Taylor C,Malik J. Modeling and Rendering Architecture from Photographs: A Hybrid Geometry-and Image-Based Approach: SIGGRAPH ’96[C]. New York,NY,USA,1996.

[10] Dick A,Torr P,Cipolla R. Modelling and Interpretation of Architecture from Several Images [J]. International Journal of Computer Vision, 2004,60(2) :111–134.

[11] Wang N,Dong J,Fang H,et al. 3D reconstruction and segmentation system for pavement potholes based on improved structure-from-motion (SFM) and deep learning[J]. Construction and Building Materials,2023,398:132499.

[12] Arikan M,Schwärzler Michael,Flöry,et al. O-Snap: Optimization-Based Snapping for Modeling Architec-

- ture[J]. ACM Trans. Graph. ,2013,32(1).
- [13] Nan L, Sharf A, Zhang H, et al. SmartBoxes for Interactive Urban Reconstruction [J]. ACM Trans. Graph. ,2010,29(4).
- [14] 张文元,陈江媛,谈国新. 基于3D基元拟合的复杂屋顶点云三维重建[J]. 地球信息科学学报. 2023,25(8):1531–1545.
- [15] Blinn J. Simulation of Wrinkled Surfaces[J]. SIG-GRAPH Comput. Graph. ,1978,12(3):286–292.
- [16] Szirmay-Kalos L, Umenhoffer T. Displacement Mapping on the GPU—State of the Art[J]. Computer Graphics Forum,2008,27(6):1567–1592.
- [17] Dummer J. Cone Step Mapping: An Iterative Ray-Heightfield Intersection Algorithm [EB/OL]. <http://www.lonesock.net/files/ConeStepMapping.pdf>,2006.
- [18] Policarpo F, Oliveira M. Relaxed Cone Stepping for Relief Mapping[M]. GPU Gems 3,2007:268–280.

## A Parallax Mapping based Approach for Building Facade Modelling

LIU Yinghu, HE Xiaoxi, MENG Fanlin, LI Jiaru, Zhu Qun

(College of Software Engineering, Chengdu University of Information Technology, Chengdu 610225 China)

**Abstract:** The complex structure of modern urban building surfaces makes it very difficult to model building surfaces in the refined way, and how to quickly reconstruct building surfaces has become a research focus in the field of building modeling. We propose a parallax mapping-based building surface modeling method, which provides a set of 2D primitive library of building surface components that can express the appearance of the building surface, simulates the details of the building surface by extracting the primitives in the primitive library, and extracts the required texture data according to the combinations of the primitives, and adds the details of the building surface based on the parallax mapping, and uses the loose cone order map to accelerate the RayMatching process of the parallax mapping in the parallax mapping. The RayMatching process of parallax mapping is accelerated by using a loose cone step map during the parallax mapping process. The final result is controllable and editable. The experimental results show that the method can efficiently generate building models with sufficient surface details, quickly modify the modeling results, and the modeling process is friendly to non-expert users.

**Keywords:** parallax mapping; cone step mapping; procedural modeling; displacement map; surface modeling