arXiv:2304.02175v1 [eess.SY] 5 Apr 2023

拉普拉斯偏微分方程能否定义穿越低空空域的空中走廊?

Aeris El Asslouj[1], Ella Atkins[2],和 Hossein Rastgoftar[1,3]

摘要——**城市无人驾驶航空器系统（UAS）的飞行需要新的法规，以确保安全并适应前所未有的交通密度。多无人驾驶航空器系统协调对实现这两个目标都至关重要。本文将 UAS 协调建模为理想流体流动，其流场由拉普拉斯偏微分方程控制。流线在空间上定义了间距紧密且冲突消解的航线，并构成空中走廊；这些走廊可安全绕开建筑物及其他构筑物，使 UAS 即使在低空垂直障碍物之间飞行、或靠近山地地形时，也能避免碰撞。我们将城市划分为多个区域，每个区域都有自己的子网络，以实现模块化，并确保航线生成所需的计算时间与总面积呈线性关系。我们通过计算若干高楼林立城市低空空域中的空中走廊，展示所提方法的优势。对于美国城市，我们使用开放的激光雷达高程数据来确定地表高程图；对于美国以外的城市，我们选取了已有高精度三维地形模型的城市。**

I. 引言

预计在城市区域上空及其中飞行的无人航空器系统（UAS）数量将迅速增加[1] ，其应用包括包裹配送 [2]，、航空摄影 [3], 和机载传感 [4]。为了让城市能够安全容纳所有新增的无人航空器系统运行，必须支持更高的空中交通密度、动态路径规划以及无人航空器系统之间的协调。尽管城市环境复杂，无人航空器系统很可能主要在低海拔运行，以确保与新型先进空中交通（AAM）流量之间的间隔，并最大限度地提高短途（< 2 公里）飞行的效率。

空中交通密度的增加意味着碰撞风险也会增加。除不具备可扩展性的广播式自动相关监视（ADS-B）之外，目前还没有通过协同路径规划实现无人航空器系统之间探测与避让（DAA）或间隔保障的能力。事实上，迄今为止，监管机构一直将无人机协调和探测与避让交由安全飞行员负责 [5] ，由他们在视距内运行中通过目视避免冲突，或通过语音与空中交通管制进行通信。

可扩展的航空器协调需要组织基础设施，包括具备全系统及机间通信（V2V）能力的数据链，以便协调并共享航空器意图（飞行航线）。无人航空系统社区目前设想通过依次设置的预定义走廊，让彼此分隔的飞行活动保持适度密度，将其作为当今有人驾驶机场交通区飞行路线的直接延伸。然而，即使设置海拔层，单队列固定走廊仍会限制交通密度，且与本文提出的协作式、无冲突动态交通路径规划能力相比，路径规划灵活性有限。

我们将空中网络定义为多车道道路网络的空中对应形式，不同之处在于空中网络是虚拟的，因此可以根据交通路径规划请求、风和其他因素进行动态重构。每个无人航空器系统都可以通过数据链实时更新其动态空中网络地图。无人航空器系统还可以存储默认的空中网络结构，以便在离线（飞行前）模式下使用。

空中网络有两种用途。首先，它们提供 [间距紧密的] 平滑路径，并与建筑物、地形、其他构筑物及相邻无人航空器系统保持适当的安全间隔。考虑环境地图和安全间隔距离，可显著简化无人航空器系统的飞行规划和动态重新规划航路。其次，它们定义了彼此安全分隔的候选路径，只要每个无人航空器系统将其规划轨迹控制在预期误差界内，就能确保避碰。

作者此前提出了一种方法，可为任何城市环境自动生成密集的空中网络，并使其绕开构筑物 [6]。该方法以包括建筑物和其他构筑物在内的地表高程数据为输入，并采用[7]中提出的流体启发式方法生成多层空中走廊。每一层都是一个海拔固定的平面，其上的空中走廊沿固定的名义方向布设，并绕开建筑物和其他构筑物。本文介绍了作者此前工作的若干改进 [6]，提高了该方法的可扩展性和易用性。具体而言，本文提出了以下几项独特的新贡献：

1) 美国城市的地表高程数据获取方法已实现全自动化。在我们此前的工作中， [6], 建筑物三维模型是根据 Google 地图数据手动创建的，这种方法无法扩展。我们现已开发出软件，可自动查询美国地质调查局（美国地质调查局）的数据库，获取以点云形式表示的地表高程数据 [8]。随后，我们会处理这些点云，以生成高分辨率地表高程图。对于美国以外的城市，只要有现成的城市高分辨率三维模型，我们就会直接从中获取建筑物和其他构筑物的定海拔“切片”。

*本研究得到美国国家科学基金会第 2133690 号和第 1914581 号奖项的资助。[1]亚利桑那大学电气与计算机工程系，美国亚利桑那州图森aymaneelasslouj@arizona.edu[2]弗吉尼亚理工大学航空航天与海洋工程系，美国弗吉尼亚州布莱克斯堡 24061 ematkins@vt.edu[3]亚利桑那大学航空航天与机械工程系，美国亚利桑那州图森hrastgoftar@arizona.edu

2) 当建筑物之间的城市峡谷中存在足够的安全间隔时，我们允许空中走廊穿行其中，以最大限度地提高空域可用性。我们的先前工作 [6] 并非如此，因为它只能处理绕单栋建筑设置空中走廊的情况。

3) 我们通过将空中走廊表示为由固定尺寸立方体组成的链条，确保空中走廊具有最小宽度；立方体的尺寸经过设定，以保证路线之间有足够的间隔。在我们先前的工作中 [6],，空中走廊被定义为两条流线之间的曲面，且未对流线间距作出限制。本文通过逐步构建由立方体链组成的走廊来近似流线，同时在每一步检查立方体是否相交。

4) 我们将空中网络划分为连通区域，使每个区域的子网络都能独立生成。在我们先前的工作中 [6], ，空中网络通过求解微分方程一次性生成。这意味着每次修改都必须从头重新生成整个网络。现在情况已不再如此。我们可以调整空中网络各区域的大小，以控制计算成本。求解一个城市的微分方程所需的时间随城市面积的 3/2 次方增长。若分别独立求解各区域的空中网络，计算时间则随城市面积线性增长。

5) 我们展示了主要城市的空中网络示例，以验证空中网络生成方法能够良好扩展，并能为真实城市生成密集且连通的网络。研究结果还为无人航空系统社区提供了直观示例，展示低空空中网络的多种航路选项。

本文结构如下。第二节介绍我们如何根据美国地质调查局数据或三维城市模型数据生成高程地图。本节还介绍如何根据高程地图生成空中网络。第三节将此方法应用于多个主要城市，第四节则总结全文并讨论未来工作。

II. 方法论

本节介绍从原始高程数据或三维城市模型生成空中网络的步骤。我们生成空中网络的方法包括生成离散高程图、求解离散流体方程，以及使用有限尺寸立方体迭代构建走廊。因此，我们首先在第二节 A 小节介绍空域的形式化离散过程。随后在第二节 B 小节使用美国地质调查局激光雷达高程数据或三维模型生成离散高程图。在第二节 C 小节，我们利用该图求解等高切片的理想流体流动方程，得到绕过建筑物的流线。最后，在第二节 D 小节利用流线图生成空中走廊。

A. 空域的离散化

由于我们生成空中网络所用的所有方法都是离散的，因此必须先将空域离散化。空域原本是一个三维连续空间。它具有一个空间原点$\mathbf{P}_0 = (0, 0, 0)$，以及一组参考基底（$\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$, $\hat{\mathbf{k}}$）。

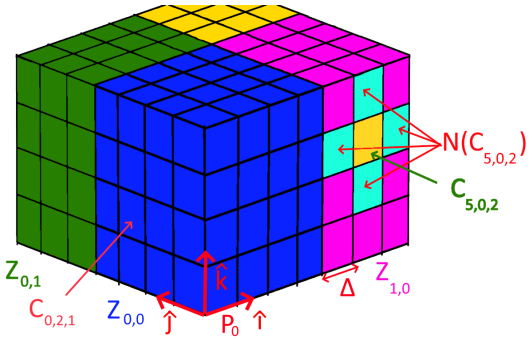


图 1：示例网格单元离散化，其中锚点$\mathbf{P}_0$,参考系（$\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$、$\hat{\mathbf{k}}$）、网格单元边长为 Δ，区域底边长度为 $N = 4$。区域 $Z_{0,0}$、 $Z_{0,1}$ 和 $Z_{1,0}$ 分别以蓝色、绿色和粉色显示。网格单元$C_{5,0,2}$ 以黄色显示，其邻居$\mathbf{N}(C_{5,0,2})$以青色显示。

为对其进行离散化，我们使用如图 1 所示的三维网格立方单元覆盖空域。网格的轴与基向量对齐，每个网格单元的边长为 $\Delta \in \mathbb{R}^+$。我们为每个网格单元分配离散的网格坐标（i, j, k）$\in \mathbb{Z}^3$，并将每个网格单元表示为 $C_{i,j,k}$。所有网格单元的集合记为 $\mathbf{C}$。每个网格单元包含原连续空域中的一个给定立方区域。给定空域中的一个点，我们可以通过函数$\mathbf{F}$获取包含该点的网格单元的网格坐标，从而确定该网格单元：

$$\begin{aligned} \mathbf{F}: \quad \mathbb{R}^3 &\rightarrow \mathbb{Z}^3 \\ (x,y,z) &\rightarrow \left(\left\lfloor \frac{x}{\Delta} \right\rfloor, \left\lfloor \frac{y}{\Delta} \right\rfloor, \left\lfloor \frac{z}{\Delta} \right\rfloor\right) \end{aligned} \tag{1}$$

其中 $\lfloor . \rfloor$ 是向下取整函数。

每个网格单元都被视为与其直接物理接触的网格单元相连。要实现这一点，这些网格单元的网格坐标必须至少在一个维度上不同。与某个网格单元相连的网格单元称为其笛卡尔邻居。图 1 展示了某个网格单元的笛卡尔邻居示例。某个网格单元 $C \in \mathbf{C}$的所有笛卡尔邻居的集合记作$\mathbf{N}(C)$，由函数 $\mathbf{N}$给出：

$$\begin{aligned} \mathbf{N}: \quad \mathbf{C} &\rightarrow \mathcal{P}(\mathbf{C}) \\ C_{i,j,k} &\rightarrow \{C_{i+1,j,k}, C_{i-1,j,k}, C_{i,j+1,k}, \\ &\quad C_{i,j-1,k}, C_{i,j,k+1}, C_{i,j,k-1}\} \end{aligned} \tag{2}$$

其中冪集 P（$\mathbf{C}$）定义了$\mathbf{C}$的所有子集组成的集合。

现在，我们得到了一个由网格单元组成的连通离散化空域。我们将该离散化空域划分为多个区域，以便为每个区域独立生成空中网络。每个区域由多个网格单元组成。一个区域的底部包含 $N \times N$ 个网格单元，并涵盖其上方的所有网格单元。图 1 展示了示例区域。所有区域组成的集合$\mathbf{Z}$ 构成了所有网格单元集合 $\mathbf{C}$ 的一个划分。我们为每个区域指定坐标（a, b）$\in \mathbb{Z}^2$，并将其记为$\mathbf{Z}_{a,b}$。集合$\mathbf{Z}_{a,b}$ 包含$\mathbf{Z}_{a,b}$所覆盖网格单元的（i, j）~~单元~~坐标。要找出区域内包含的所有网格单元，我们使用

如下：

$$\overline{\mathbf{Z}_{a,b}} = \left\{ (i,j) \in \mathbb{Z}^2 \middle| \begin{array}{l} Na \le i < N(a+1) \\ Nb \le j < N(b+1) \end{array} \right\}, \forall (a,b) \in \mathbb{Z}^2. \quad (3a)$$

$$\mathbf{Z}_{a,b} = \left\{ C_{i,j,k} \in \mathbf{C} \middle| \ (i,j) \in \overline{\mathbf{Z}_{a,b}} \right\}, \quad \forall (a,b) \in \mathbb{Z}^2. \quad (3b)$$

B. 高程地图生成

城市空域中存在以建筑物和其他构筑物形式出现的障碍物。为了生成能够安全绕开所有障碍物的空中网络，我们必须确定离散空域中哪些网格单元与障碍物相交。因此，我们将网格单元分为两类。空闲单元是指不与任何障碍物相交的网格单元。占用单元是指与障碍物相交的网格单元。

为简化我们的模型，我们将所有建筑物和结构与地面合并，并将其整体视为海拔各异的地形。因此，对于每个水平网格单元坐标（$i,\ j$）$\in \mathbb{Z}^2$，都存在一个最大单元高度；低于该高度的所有网格单元均为满，高于该高度的所有网格单元均为空。对于每个区域$\mathbf{Z}_{a,\ b} \in \mathbf{Z}$，我们定义一个离散高程图$M_{a,\ b} : \mathbf{Z}_{a,\ b} \to \mathbb{Z}$该图给出每个水平网格单元坐标对应的“地面”单元高度。因此，可以通过函数 T获取网格单元的类型：

$$\begin{array}{rccl} T: & \mathbf{Z}_{a,b} & \to & \{\text{full}, \text{free}\} \\ & C_{i,j,k} & \to & \begin{cases} \text{full} & k \le M_{a,b}(i,j) \\ \text{free} & k > M_{a,b}(i,j) \end{cases} \end{array}. \quad (4)$$

我们使用两种不同的方法来获取某个区域的离散高程图$M_{a,\ b}$。如果生成空中网络的城市位于美国，我们使用美国地质调查局的激光雷达高程数据。如果该城市位于美国境外，则必须使用该城市现有的三维模型。

1）基于美国地质调查局点云生成美国城市的高程地图：

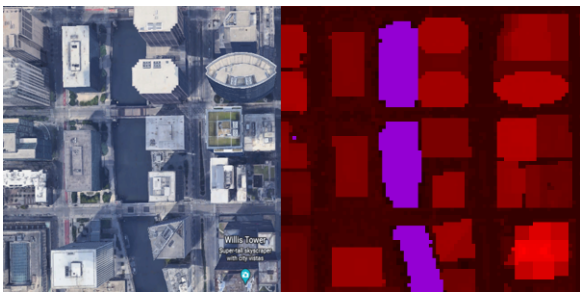


图 2：芝加哥城市区域的谷歌地球卫星视图（左）及其基于美国地质调查局激光雷达数据生成的对应高程地图（右）。红色的深浅与网格单元高度的平方根成正比。紫色表示没有高度数据的坐标对（i,j），其中 $M_{1,0}$ 为正无穷。

对于美国的大多数地区，美国地质调查局提供激光雷达高程数据 [8]。激光雷达点云由配备激光雷达的飞机和旋翼机飞越某一区域获取，用于估算地形高程，包括建筑物及其他构筑物。请注意，激光雷达可能无法捕捉狭窄的垂直障碍物和输电线；研究人员正在探索互补方法来应对这一挑战 [9]。对于给定区域，我们的系统会查询美国地质调查局的数据库，查找包含位于该区域内点的点云。找到的点云被归为一组$\mathbf{P}_{a,\ b}$。这些点云采用投影坐标系。因此，我们首先平移所有点，使得我们的原点$\mathbf{P}_0$ 与数据坐标系的原点重合。然后，我们使用式（1）中定义的函数 $\mathbf{F}$ 将点映射到网格单元，得到网格单元点云$\mathbf{F}\left(\mathbf{P}_{a,\ b}\right)$。最后，对于每个水平网格坐标（$i,\ j$）$\in \mathbf{Z}_{a,\ b}$，我们从网格单元点云中选取最高的单元高度，作为离散高程图$M_{a,\ b}$（$i,\ j$）中的地形高度。因此，$M_{a,\ b}$（$i,\ j$）可表示为：

$$\begin{array}{rccl} M_{a,b}: & \overline{\mathbf{Z}_{a,b}} & \to & \mathbb{Z} \\ & (i,j) & \to & \max\left\{k \middle| \ (i,j,k) \in \mathbf{F}\left(\mathbf{P}_{a,b}\right)\right\} \end{array} \quad (5)$$

某些水平坐标可能没有对应的点云网格单元。在这种情况下，我们将这些坐标的地形海拔设为无穷大。这是最保守的做法，因为它假设没有高程数据的空间中存在极高的地物。图 2 展示了离散高程图示例。

2）基于三维模型生成的美国以外城市高程地图：

对于美国以外的城市，我们使用现有三维模型作为数据源。对于这些城市，我们使用三维建模软件 Blender [10]，在特定单元高度处对三维模型进行切片。这些切片有助于区分每个指定高度处的“占用”网格单元和“空闲”网格单元。

虽然这种方法更为直接，但它假设城市存在三维模型，而这一点并无保证。此外，目前这仍是一个手动过程，尽管切片过程可以实现自动化。最后，高保真三维模型通常是付费模型，而美国地质调查局（USGS）的数据则是公开、免费且覆盖美国本土大部分地区的。

C. 流线图生成

我们的目标是在固定海拔层生成围绕建筑物的空中走廊。为此，我们在建筑物之间的固定海拔二维平面内，求解理想无旋流体流动的方程。这样便可获得所述层中的流线分布。流线分布将在第 II-D 节中用于指导空中走廊生成。

输入参数包括空中网络的层数 $M \in \mathbf{N}$，以及对于每一层 $n \in \{1 \cdots M\}$，其海拔$\hat{h}_n$ 和该层空中走廊的名义方向$\mathbf{d}_n$。

对于每个区域，我们都必须在给定的每个海拔高度求解流体流动方程。在本节中，我们将区域固定为 $\mathbf{Z}' \in \mathbf{Z}$，并将所讨论的层设为 $M' \in \{1 \cdots M\}$，使得 $h' = h_{M'}$ 且 $\hat{\mathbf{d}}' = \hat{\mathbf{d}}_{M'}$。

首先，我们使用式 (1) 中定义的函数 $\mathbf{F}$，将连续的空域高度 h' 转换为单元高度 k'。然后，我们在单元高度 k'处，对区域 $\mathbf{Z}'$ 取一个水平切片 $\mathbf{Z}$。

区域在单元高度 k 处的水平切片，是该区域内单元高度为 k 的网格单元子集。具体而言：

$$\tilde{\mathbf{Z}} = \{C_{i,j,k} \in \mathbf{Z}' | k = k'\}. \tag{6}$$

如果连续理想无旋流体在˜切片 $\mathbf{Z}$ 中流动，它将形成流线，使平面中的每个点都成为某条流线的一部分。每条流线都可以被赋予一个实数索引 ψ，而这个 ψ 分布满足拉普拉斯方程：

$$\nabla^2 \psi = 0. \tag{7}$$

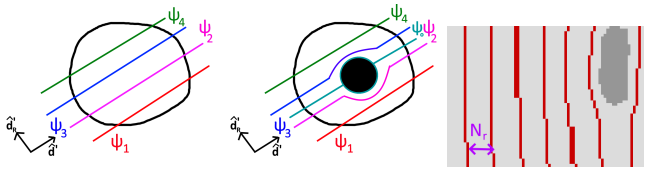
在我们的案例中，切片由离散单元组成。因此，使用离散版本近似 ψ分布，该版本˜为切片中的每个单元 $C \in \mathbf{Z}$ 分配一个流线索引$\psi(C)$。拉普拉斯方程也转换为其离散形式：

$$\sum_{C' \in (\mathbf{N}(C) \cap \tilde{\mathbf{Z}})} (\psi(C) - \psi(C')) = 0, \quad \forall C \in \tilde{\mathbf{Z}}. \tag{8}$$

在公式 (8) 中，$(\mathbf{N}(C) \cap \tilde{\mathbf{Z}})$ 是与 C 相邻且位于同一切片中的所有邻居的集合。要获得 ψ 分布，我们首先必须定义拉普拉斯差分方程的边界条件，然后求解该方程。

1）边界条件：

边界条件有两种类型。第一种是边界单元的 ψ 值，用于控制流线的名义方向。第二种是占用单元的 ψ 值，用于控制流线如何绕过这些单元。



(a) (b) (c)

图 3：（a）为边界单元分配 ψ 值的方法示意图。名义流线方向为$\hat{\mathbf{d}}'$，而$\hat{\mathbf{d}}'_R$ 与其正交。（b）为障碍物分配 ψ 值的方法示意图。为使单条流线经过障碍物边界，为该障碍物分配了 ψ_o。在此$\psi_4 < \psi_3 < \psi_o < \psi_2 < \psi_1$。（c）两个空中走廊起点之间最小单元距离的示意图。

切片边界上的所有网格单元集合记为$\partial\mathbf{Z}$。边界上具有相同 ψ值的任意两个网格单元位于同一条流线上。因此，为使流线具有$\hat{\mathbf{d}}'$的名义方向，方向为$\hat{\mathbf{d}}'$ 的直线上的任意两点必须具有相同的 ψ 值，如图 3（a）所示。一种实现方式是：当沿正交于$\hat{\mathbf{d}}'$的方向移动时，使边界单元的ψ 值发生变化。最简单的方法是将$\hat{\mathbf{d}}'$ 旋转 90 度得到$\hat{\mathbf{d}}'_R$，然后令边界单元的 ψ 值为该网格单元沿$\hat{\mathbf{d}}'_R$的坐标。由此得到：

$$\psi(C_{i,j,k}) = \begin{bmatrix} i & j \end{bmatrix}^T . \hat{\mathbf{d}}'_R, \qquad \forall C_{i,j,k} \in \partial\tilde{\mathbf{Z}}, \tag{9}$$

其中，“·”是内积符号。

公式（9）是 [6] 中用于求边界单元 ψ 值的公式的一种更一般形式。这是因为公式（9）适用于任意名义流线方向 $\hat{\mathbf{d}}'$，而 [6] 中的公式仅适用于 $\hat{\mathbf{d}}' = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$ 或 $\hat{\mathbf{d}}' = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$。

彼此均为邻居的占用单元群构成一个障碍物。为了将占用单元划分为不同的障碍物，我们从任意一个尚未分类的占用单元开始，迭代地将其与相邻的占用单元归为一组。没有剩余的相邻占用单元后，该障碍物即告完成，随后从另一个尚未分类的占用单元开始这一过程。所有占用单元都归入障碍物后，我们为它们赋予 ψ 值。

为了使流线像图 3 (b) 所示那样绕过障碍物，障碍物的边界单元必须具有相同的ψ 值。这等同于让障碍物的所有网格单元都取同一个 ψ 值。ψ 较高的流线会从一侧绕过障碍物，而 ψ 较低的流线则会从另一侧绕行。

我们在此采用的方法与先前的工作不同。在我们之前的论文 [6], 中，所有障碍物都被赋予了 $\psi = 0$。这使得流线无法穿行于建筑物之间，而是都沿着城市区域的边界延伸。这是因为，为所有障碍物设置 $\psi = 0$ 实际上会将它们合并为一个由单条流线覆盖的障碍物。为了解决这一问题，我们修改了为障碍物赋予 ψ 值的方法。

我们的目标是针对每个障碍物，选取使流线流动受到干扰最小的 ψ 值。因此，我们首先选取障碍物的中心坐标$\mathbf{r}_o$。然后使用函数$\mathbf{F}$将其转换为单元坐标。最后，我们使用与边界单元相同的公式，即式（9），来选取其 ψ 值：

$$\psi_o = \mathbf{F}(\mathbf{r}_o)^T . \hat{\mathbf{d}}'_R. \tag{10}$$

2）求解差分拉普拉斯方程

本节介绍一种针对切片$\mathbf{Z}$ 中˜网格单元的新索引系统，以简化方程。不同于像$C_{i、j、k}$那样使用三个索引来指代网格单元，我们用单个索引 $I \in \{1、\cdots、m\}$来指代网格单元，其中 $m = N^2$是该切片中的网格单元数量。这个新索引系统旨在将网格单元分为以下几类：

1）（1、$\cdots$、m_b）为边界单元。2）（m_b+1、$\cdots$、m_b+m_f）为自由非边界单元。3）（m_b+m_f+1、$\cdots$、$m_b+m_f+m_o$）为满非边界单元。

其中：

1) $\boldsymbol{\psi}_b = \begin{bmatrix} \psi(C_1) & \cdots & \psi(C_{m_b}) \end{bmatrix}^T$.

2) $\boldsymbol{\psi}_f = \begin{bmatrix} \psi(C_{m_b+1}) & \cdots & \psi(C_{m_b+m_f}) \end{bmatrix}^T$.

3) $\boldsymbol{\psi}_o = \begin{bmatrix} \psi(C_{m_b+m_f+1}) & \cdots & \psi(C_{m_b+m_f+m_o}) \end{bmatrix}^T$.

令$\mathbf{L} = [L_{IJ}] \in \mathbb{R}^{m \times m}$为$\mathbf{Z}$ 的拉普拉斯矩阵 [11]，其元素表示˜该切片中每个网格单元拥有的邻居数量˜：$\mathbf{Z}$

$$L_{IJ} = \begin{cases} |\mathbf{N}(C_I) \cap \mathbf{Z}_k| & I = J \\ -1 & I \neq J \wedge C_J \in N(C_I), \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

其中，邻近函数 $\mathbf{N}(\cdot)$ 已在式（2）中定义。

矩阵$\mathbf{L}$ 可拆分为九个子矩阵，用于定义三个单元组之间的连接关系：

$$\mathbf{L}=\begin{bmatrix}\mathbf{L}_{bb} & \mathbf{L}_{bf} & \mathbf{L}_{bo}\\ \mathbf{L}_{fb} & \mathbf{L}_{fc} & \mathbf{L}_{fo}\\ \mathbf{L}_{ob} & \mathbf{L}_{of} & \mathbf{L}_{oo}\end{bmatrix}. \tag{12}$$

由此，我们可以将拉普拉斯差分方程 Eq. (8) 的所有实例合并为一个线性系统：

$$\mathbf{L}_{fb}\boldsymbol{\psi}_b+\mathbf{L}_{fc}\boldsymbol{\psi}_f+\mathbf{L}_{fo}\boldsymbol{\psi}_o=0. \tag{13}$$

我们知道所有边界单元和完整非边界单元的 ψ 值，它们分别是 $\boldsymbol{\psi}_b$ 和 $\boldsymbol{\psi}_o$。因此，我们可以通过以下方法求出自由非边界单元 $\boldsymbol{\psi}_f$ 的 ψ 值：

$$\boldsymbol{\psi}_f=-\mathbf{L}_{fc}{}^{-1}(\mathbf{L}_{fb}\boldsymbol{\psi}_b+\mathbf{L}_{fo}\boldsymbol{\psi}_o). \tag{14}$$

不应使用常规求解器求解该线性系统，因为 $\mathbf{L}$ 通常非常大，其规模为$m^2=N^4$。此外，其非零元素的数量级为 N^2，因此它是一个稀疏矩阵 [12]。因此，我们使用稀疏线性系统求解方法，以较低的计算成本求解 Eq. (12) 给出的线性方程组。由于 $\mathbf{L}$ 和 $\mathbf{L}_{fc}$ 均为对称矩阵，我们使用共轭梯度法 [13] 来求得 $\boldsymbol{\psi}_f$。

D. 空中走廊生成

有了完整的离散流线分布 ψ，我们最终可以生成空中走廊。空中走廊是由相连网格单元构成的链，其起点和终点均位于切片 $\mathbf{Z}$ 的边界˜上。它们还近似于流线，且彼此之间以及与障碍物均不相交。

我们对空中走廊的定义确保其宽度有限，且等于网格单元的大小 Δ。这与先前的研究 [6] 不同，先前的研究将空中走廊定义为两条流线之间的区域。直接使用流线来确定走廊边界的问题在于，我们找不到一种简单的方法来确保流线之间至少相隔一定距离。这一点很重要，因为无人机有一定的宽度，因此必须确保空中走廊具有最小宽度，而我们对空中走廊的新定义正是为了做到这一点。

要生成空中走廊，我们从一个边界单元$C_b\in\partial\mathbf{Z}$开始。根据一种称为方向的属性，将边界单元分为三组。

1) 如果在 C_b的位置，$\hat{\mathbf{d}}'$ 指向切片内部，则 C_b 的方向为向前。2) 如果在 C_b的位置，$\hat{\mathbf{d}}'$ 指向切片外部，则 C_b 的方向为向后。3) 否则， C_b 没有方向。

我们仅在 C_b 具有方向时，才将其视为空中走廊的起点。如果 C_b 朝前定向，我们就沿$\hat{\mathbf{d}}=\hat{\mathbf{d}}'$ 方向扩展走廊。如果 C_b 朝后定向，则改用$\hat{\mathbf{d}}=-\hat{\mathbf{d}}'$。要生成空中走廊，我们使用算法 1。

算法1的核心思路是尝试沿着流线前进，同时逐步添加网格单元。空中走廊是由网格单元组成的列表。每次循环运行时，都会选取走廊中最后一个网格单元的所有邻居。

算法 1： 从以下对象生成走廊的算法该 ψ 分布。

输入： $C_b\hat{\mathbf{d}}$、ψ、先前构建的空中走廊
结果： AirCorridor 是一条空中走廊，起点为 C_b 并进入其中的 $\hat{\mathbf{d}}$ 方向
最后单元 $\leftarrow C_b$;
空中走廊 ←列表;
将最后单元添加到空中走廊;
原始ψ $\leftarrow\psi(C_b)$;
当 最后单元 $\notin\partial\mathbf{Z}_k$ 或 最后单元的方向*(*最后单元*)* = 方向(C_b) **执行**
  邻居 $\leftarrow N$(最后单元) $\cap\mathbf{Z}_k$;
  丢弃已满的邻居;
  丢弃属于空中走廊的邻居;
  丢弃沿途进度的邻居 $\hat{\mathbf{d}}$ is 小于最后单元的进度;
  (网格单元的进度 $C_{i,j,k}\in\mathbf{Z}_k$ 沿着 $\hat{\mathbf{d}}$ is 定义为 $\begin{bmatrix}i & j\end{bmatrix}^T\cdot\hat{\mathbf{d}}$.);
  如果 没有剩余邻居**则**
    停止，尝试失败;
  end
  NextCell $\leftarrow$ 与 ψ 最接近的邻居原始ψ;
  如果*NextCell* 属于另一条走廊 **则**
    停止，尝试失败;
  end
  将 NextCell 添加到空中走廊;
  最后单元 ←NextCell;
end
尝试成功;

从选出的网格单元中，移除已满的网格单元、已属于走廊的网格单元，以及无法使走廊沿扩展ˆ方向$\hat{\mathbf{d}}$前进的网格单元。这样可以确保走廊不包含障碍单元，也不会过度绕行。如果没有剩余网格单元，则说明到达了死路，走廊将被丢弃。否则，我们选择ψ 值最接近原始网格单元 C_b的邻居，以确保走廊保持在同一流线附近。如果选中的网格单元已属于已建立的走廊，则将其视为死路并丢弃该走廊，以避免走廊交叉和过度偏离。如果选中的网格单元可用，就将其添加到走廊中，并再次运行循环。当走廊到达一个方向不同于 C_b的边界单元时，循环停止，这意味着走廊到达了切片边界的另一端。

我们会尝试在切片边界上的每个网格单元生成空中走廊。我们正在为其生成这些空中走廊的区域，可能与已有子网络的邻接区域相邻。如果是这样，我们会优先选择能够与邻接区域现有走廊连通的边界单元。尝试完所有这些潜在连接后，再尝试其余

边界单元。这样可以确保空中网络具有最大的区域间连通性。

为避免网络密度出现较大波动，我们在尝试生成走廊的边界单元之间设置最小单元距离 N_r（见图3(c)）。

在区域 $\mathbf{Z}'$内，对于每两层垂直相邻的走廊，我们允许它们之间建立垂直连接。这些垂直连接由连接两层走廊的单元垂直链构成。当每个切片中具有相同水平坐标（i, j）的网格单元都属于各自所在层的走廊时，就会建立这些连接。这确保了空中网络具有最大的层间连通性。

III. 结果

我们在表I所列城市的部分区域测试了空中网络生成方法。表I还列出了各城市对应的空中网络参数及其生成所需时间。对于迪拜、上海和东京，我们使用了现有的三维模型。对于芝加哥，我们使用美国地质调查局激光雷达高程数据创建其离散高程图。迪拜、上海和东京的三维模型图片分别见图4(a)、图4(b)和图4(c)。这些图还展示了为空中网络各层生成的定高切片。芝加哥区域 $\mathbf{Z}^{1,0}$ 的离散高程图 $M_{1,\ 0}$ 见图4(d)。

对于所有城市，所构建的空中网络均包含两个具有正交标称流向的层。下层沿 +x 方向流动，而上层沿 +y 方向流动。迪拜、上海和东京生成的空中网络分别如图5(a)、图5(b)和图5(c)所示。如图5(a-c)所示，生成的空中网络由两个区域 $\mathbf{Z}^{0,0}$ 和$\mathbf{Z}^{0,1}$组成。图5(d)展示了迪拜空中网络的上层，表明空中走廊不会相交（迪拜空中网络的上层表明，我们的方法能够为较高比例的走廊成功生成区域间连接）。此外，芝加哥生成的空中网络如图6所示。我们注意到，空中网络图（图5(a-c)和图6）显示的是从上方观察到的网络，因此空中走廊看起来似乎相交。但实际并非如此，因为沿 +x 方向流动的走廊和沿 +y 方向流动的走廊位于垂直分隔的两个不同层中。

表I：用于生成空中网络样本的城市

城市、国家	参数	生成时间
迪拜，阿拉伯联合酋长国	表II	20.0 秒
中国上海	表III	3.63 秒
日本东京	表IV	3.00 秒
美国芝加哥	表 V	1.78 秒

IV. 结论

本文提出了一种空中网络生成方法，有望安全支持高无人机系统交通密度

表II：迪拜空中网络的参数

Name	值
锚点$\mathbf{P}_0$ 经度	经度 = 55.251389°
锚点$\mathbf{P}_0$ 纬度	纬度 = 25.187778°
锚点$\mathbf{P}_0$ 平均海平面以上高度	alt = 2 米
坐标系	自定义
单元格大小	Δ = 5米
区域水平尺寸	北 = 476
第一走廊层平均海平面以上高度	时$_1$ = 80米
第一走廊层方向	$\hat{\mathbf{d}}_1 = [1.\ 0\quad 0.0]$
第二走廊层平均海平面以上高度	h_2 = 90米
第二走廊层方向	<escape>$\hat{\mathbf{d}}_2 = [0.0\ 1.0]$</escape>
廊道端点之间的最小单元数	N_r = 10

表III：上海空中网络的参数

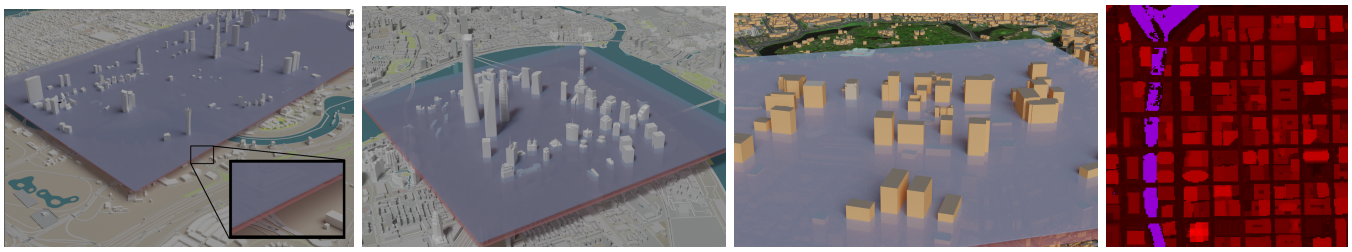
Name	价值
锚点$\mathbf{P}_0$ 经度	经度 = 121.496944°
锚点$\mathbf{P}_0$ 纬度	纬度 = 31.22889°
锚点 $\mathbf{P}_0$ 平均海平面以上高度	海拔 = 0 米
坐标系	自定义
单元格大小	Δ = 5米
区域水平尺寸	N = 350
第一走廊层平均海平面以上高度	小时$_1$ = 99米
第一走廊层方向	$\hat{\mathbf{d}}_1 = [1.\ 0\ 0.\ 0]$
第二走廊层平均海平面以上高度	小时$_2$ = 129米
第二走廊层方向	$\hat{\mathbf{d}}_2 = [0.\ 0\quad 1.\ 0]$
廊道端点之间的最小单元数	N_r = 10

表IV：东京空中网络的参数

Name	值
锚点$\mathbf{P}_0$ 经度	经度 = 139.776944°
锚点$\mathbf{P}_0$ 纬度	纬度 = 35.675278°
锚点 $\mathbf{P}_0$ 平均海平面以上高度	alt = 4 米
坐标系	自定义
单元格大小	Δ = 5米
区域水平尺寸	北 = 287
第一走廊层平均海平面以上高度	h_1 = 76米
第一走廊层方向	$\hat{\mathbf{d}}_1 = [1.0\quad 0.0]$
第二走廊层平均海平面以上高度	h_2 = 106米
第二走廊层方向	$\hat{\mathbf{d}}_2 = [0.0\ 1.0]$
廊道端点之间的最小单元数	N_r = 10

表 V：芝加哥空中网络的参数

Name	值
锚点$\mathbf{P}_0$ 经度	经度 = -87.65250°
锚点$\mathbf{P}_0$ 纬度	lat = 41.87833°
锚点 $\mathbf{P}_0$ 平均海平面以上高度	alt = 4 米
坐标系	EPSG 6455
单元格大小	Δ = 5米
区域水平尺寸	N = 200
第一走廊层平均海平面以上高度	小时$_1$ = 235米
第一走廊层方向	$\hat{\mathbf{d}}_1 = [1.0\quad 0.0]$
第二走廊层平均海平面以上高度	h_2 = 245米
第二走廊层方向	$\hat{\mathbf{d}}_2 = [0.\ 0\quad 1.0]$
廊道端点之间的最小单元数	N_r = 10



(a) (b) (c) (d)

图 4：用于 (a) 迪拜 [14], 上海 [15], 和东京 [16] 的三维模型。在这三幅子图中，空中网络下层的切片以红色表示，上层的切片以蓝色表示。(d) 基于美国地质调查局激光雷达数据绘制的芝加哥区域离散高程图 $_{,\ 0}\ \mathbf{Z}^{1,0}$ 。红色的深浅与单元高度的平方根成正比。紫色表示没有海拔数据的 $(i,\ j)$ 坐标对，其中 $M_{1,\ 0}$ 为正无穷。

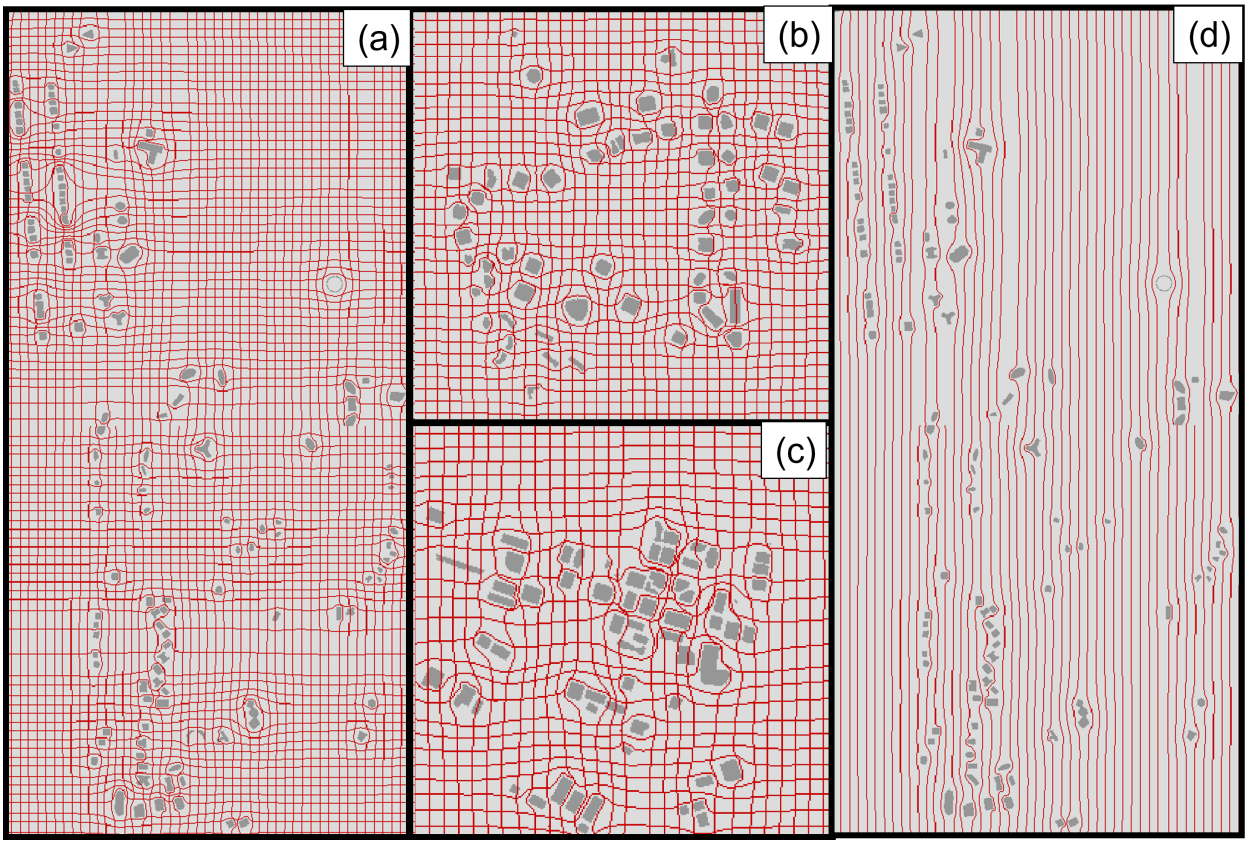


图 5：穿过区域 $\mathbf{Z}^{0,0}$ （下方）和 $\mathbf{Z}^{0,1}$ （上方）的 (a) 迪拜、(b) 上海和 (Tokyo) 空中网络。在这些子图 (a-c) 中，红色表示上下两层的走廊单元，灰色表示穿过下层单元高度处切片的占用网格单元。(d) 迪拜空中网络的上层，穿过区域 $\mathbf{Z}^{0,0}$ （下方）和 $\mathbf{Z}^{0,1}$ （上方）。红色表示走廊单元，灰色表示穿过上层单元高度处切片的占用网格单元。

在高密度城市区域开展低空飞行。与作者此前的研究相比，本研究在自动化、可扩展性、航线间隔安全性和计算开销方面均有所改进。用于定义建筑物之间流线的流体流动方程采用稀疏线性方程组求解器进行求解，并针对城市的每个区域分别独立求解，以最大限度地减少计算时间。通过采用基于有限尺寸块的迭代生成方法，确保空中走廊具有最小宽度。

未来的研究应着重于改进这一过程还可以通过考虑海拔变化来进一步完善。还应在安全间隔、交通密度、数据链路延迟以及真实的无人航空系统轨迹跟踪精度等指标的背景下，进一步研究区域大小、层数和密度等空中网络参数的影响。未来的工作还可探讨如何拓展本研究，以安全、高效地支持在同一城市空中网络中运行且具有不同性能特征的无人航空器系统。

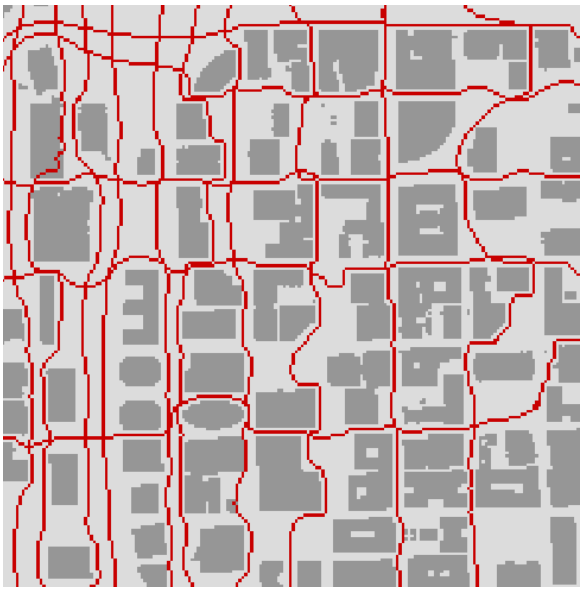


图6：芝加哥通过区域构建的空中网络$\mathbf{Z}_{0,0}$（底部）和$\mathbf{Z}_{0,1}$（顶部）。红色表示两个层中的走廊单元，而灰色表示在下层海拔处沿切片分布的占用单元。

参考文献

[1] Mordor Intelligence行业报告，《无人驾驶飞行器市场：增长、趋势、COVID-19的影响及预测（2022—2027）》，Mordor Intelligence。[2] S. Jung和H. Kim，《Amazon Prime Air无人机配送服务分析》，知识信息技术与系统期刊，第12卷，第2期，第253–266页，2017年。[3] G. Eckert、S. Cassidy、N. Tian和M. E. Shabana，《利用空中无人机摄影构建真实物体的三维模型，以期缩短自然灾害后的响应时间并降低维修成本》，载于科学与信息会议。Springer，2019年，第317–325页。[4] A. V. Savkin和H. Huang，《监视无人机网络的优化部署方法》，*IEEE*系统期刊，第13卷，第4期，第4474–4477页，2019年。[5]《娱乐飞行者与社区组织》。[在线]。网址：https://www.faa.gov/uas/recreational flyers [6] H. Emadi、E. Atkins和H. Rastgoftar，《用于无人机系统交通管理的有限状态固定走廊模型》，*arXiv*预印本 *arXiv:2204 05517*，2022年。[7] H. Rastgoftar和E. Atkins，《用于无人机系统交通协调的基于物理、可自由缩放的连续变形》，*IEEE*控制网络系统汇刊，第7卷，第2期，第532–544页，2020年。[8]《国家地图》。 [在线]。网址：https://www.usgs.gov/programs/national-geospatial-program/national-map [9] P. R. Flanigen、R. R. Copeland、N. Sarter和E. M. Atkins，《机载垂直障碍物探测的当前挑战及缓解措施》，载于面向多域作战的虚拟现实、增强现实与混合现实（*XR*）技术 *III*，第12125卷。SPIE，2022年，第57–69页。[10] B. Foundation，《Blender项目之家——免费开源三维创作软件》。 [在线]。网址：https://www.blender.org/ [11] R. Merris，《图的拉普拉斯矩阵：综述》，线性代数及其应用，第197–198卷，第143–176页，1994年。[在线]。网址：https://www.sciencedirect.com/science/article/pii/0024379594904863[12] T. A. Davis、S. Rajamanickam和W. M. Sid-Lakhdar，《稀疏线性系统直接方法综述》，*Acta Numerica*，第25卷，第383–566页，2016年。[13] J. L. Nazareth，《共轭梯度法》，*WIREs*计算统计学，第1卷，第3期，第348–353页，2009年。 [在线]。网址：https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.13 [14]“迪拜商务湾阿联酋：三维模型。”[在线]。可访问：https://www.cgtrader.com/3d-models/exterior/cityscape/dubai-business-bay-uae-5km [15]“上海市中心中国：三维模型。”[在线]。可访问：https://www.cgtrader.com/3d-models/exterior/cityscape/shanghai-downtown-china [16]“东京日本城市景观：三维模型。”[在线]。可访问：https://www.cgtrader.com/3d-models/exterior/cityscape/cityscape-tokyo-japan-7caab6ac-2b3c-4de0-82a0-7a30a7e29184