基于消失点引导的驾驶场景视频语义分割

郭典典 [1] 范登平 [3,2*] 路通宇 [1] Christos Sakaridis[1] Luc Van Gool[1]
[1] 苏黎世联邦理工学院计算机视觉实验室 [2] 南开大学计算机学院 DISSec
[3] 南开国际先进研究院（深圳福田）

摘要

在驾驶场景的视频语义分割（*VSS*）中，跨帧隐式对应关系的估计与较高的计算开销长期以来是两项主要挑战。已有方法通常采用关键帧、特征传播或跨帧注意力来应对这些问题。不同于上述方法，本文首次利用消失点（*VP*）先验以实现更有效的分割。直观而言，靠近 *VP*、也即距离车辆较远的目标通常更难辨识；同时，在前视摄像头、直线道路以及车辆直线前行的常见情形下，这些目标往往会随时间沿径向远离 *VP* 运动。基于这一观察，本文提出了一种新颖且高效的视频语义分割网络 ***VPSeg***。该网络包含两个分别利用静态与动态 *VP* 先验的核心模块：稀疏到稠密特征挖掘模块（*DenseVP*）和 *VP* 引导的运动融合模块（*MotionVP*）。其中，*MotionVP* 通过 *VP* 引导的运动估计建立显式跨帧对应关系，并帮助模型关注相邻帧中最相关的特征；*DenseVP* 则增强靠近 *VP* 的远处区域中的微弱动态特征。上述模块运行于上下文-细节框架之中，该框架以低分辨率输入建模上下文、以高分辨率输入保留局部细节，从而降低计算开销。随后，上下文特征与局部特征通过上下文感知运动注意力模块（*CMA*）进行融合，并用于最终预测。在 *Cityscapes* 和 *ACDC* 两个主流驾驶场景分割基准上的大量实验表明，*VPSeg* 仅以适度的额外计算开销便取得了优于以往 *SOTA* 方法的性能。代码已发布于 https://github.com/RascalGdd/VPSeg。

*通讯作者：范登平 *(dengpfan@gmail.com)*。郭典典与路通宇对本文贡献相同。

本文为 CVPR 2024 [14] 的中文翻译版，由聂博文翻译，范登平、马琦校稿。

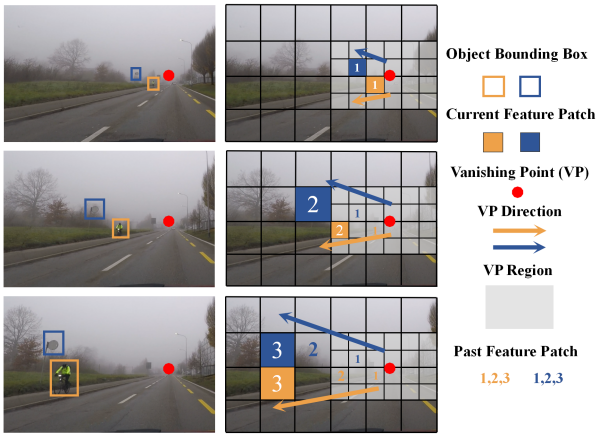


图 1. **本文所提消失点引导运动估计与尺度自适应划分模块背后的设计直觉**。在前视摄像头、直线道路与车辆直线前行的典型情形下，如图所示，随着视频时间推移，目标沿径向远离消失点运动。此外，靠近消失点的区域通常包含距离车辆较远的目标，这些目标在图像中呈现为更小的尺度。

1. 引言

在自动驾驶兴起的初期，对车辆周围环境的全面感知已成为基本要求。语义分割，即将摄像头帧中每个像素分类到一组预定义类别，是该领域的核心任务。然而，自动驾驶场景下的语义分割面临独特挑战：目标种类繁多、尺度变化剧烈、遮挡时有发生，加之光照与天气条件跨度极大，使模型必须实时解析复杂视觉输入。其中，难分割目标 [12] 尤为棘手，这类目标尺寸小、出现频次低，或其外观与背景及相邻目标难以区分。典型实例如远处的交通标志或光照不足处的行人，它们对驾驶决策影响重大，因此准确分割这类目标至关重要。

连续帧中的动态上下文为识别这些困难样本提供了线索，研究者因此尝试通过视频语义分割（VSS）[25, 35, 45, 46, 65] 利用时序信息来解决该问题。然而，同

时处理多帧图像需要大量计算资源。此外，现有 VSS 方法在两类主要场景下仍难以建立对应关系。其一，远处小目标随时间的相对运动往往十分细微，容易被忽略。其二，在高速驾驶场景中，目标位置与外观的快速变化给运动估计带来挑战。常见方法以光流辅助 VSS [19, 52, 65]，但光流不仅无法应对快速运动，还会引入更高的推理延迟。近期工作转向使用局部注意力来利用时序信息 [25, 45]。然而，注意力机制中较粗的粒度虽能捕获更广泛的上下文，却可能丢失细微的运动特征。此外，局部注意力采用的非动态特征跟踪也容易遗漏快速运动特征。

受透视投影基本原理的启发，本文假设消失点（VP）能够为应对驾驶场景 VSS 的上述挑战提供有价值的先验。如图 1所示，视频连续帧中目标的表观运动通常依赖于 VP 的位置：在前视摄像头、直线道路与车辆直线前行的典型情形下，静态目标随时间沿径向远离 VP 移动。因此，这种与 VP 相关的运动先验可用于约束运动估计，从而建立显式的跨帧对应关系。此外，帧中靠近 VP 的区域通常对应驾驶场景的远处部分，其中的目标在图像中也呈现为更小尺度；这种与 VP 相关的静态帧内先验，有助于模型定量定位这些关键区域，并增强其中的特征。同时，VP 检测仅需分析帧中的特定线段或特征点，大幅降低了额外计算开销，且不会显著降低推理速度。因此，如何利用上述动态与静态 VP 先验引导 VSS，成为一个关键问题。

针对这一问题，本文提出 VPSeg，一种面向 VSS 的消失点引导网络。VPSeg 通过两个新颖模块分别利用动态与静态 VP 先验：VP 引导的运动融合（MotionVP）与稀疏到稠密特征挖掘（DenseVP）。具体而言，MotionVP 通过 VP 引导的运动估计建立显式跨帧对应关系，进而生成动态上下文。DenseVP 则在靠近 VP 的区域采用尺度自适应划分策略，针对该区域内通常难以分辨的运动，提取更细粒度的特征。MotionVP 与 DenseVP 均置于上下文-细节框架中：动态上下文与局部上下文从双线性下采样后的低分辨率输入中提取；随后，通过上下文感知运动注意力（CMA）融合局部上下文与动态上下文，生成细节权重图，并以此引导动态上下文与高分辨率特征融合，得到最终预测。本文贡献总结如下：

- 提出了 MotionVP，一种面向 VSS 的 VP 引导运动估计策略，可建立显式跨帧对应关系，对运动幅度较大的高速驾驶场景尤其有效。
- 提出了 DenseVP，一种面向 VSS 的 VP 引导尺度自适应划分方法，可在 VP 区域内针对困难样本提取更细粒度的特征，从而提升远处难分割目标的表征能力。
- 设计了 VPSeg，一种面向 VSS 的高效上下文-细节框架，通过低分辨率上下文建模、高分辨率细节保留及二者的自适应融合，降低视频帧分割的计算开销。

2. 相关工作

语义分割即用一组目标类别对图像进行像素级标注 [36]。随着深度学习兴起，各类分割网络相继涌现 [4–6, 10, 15, 18, 20, 21, 26, 30, 57, 59, 61, 63, 64, 66]，挖掘更丰富的空间上下文已成为该任务的核心议题。视频语义分割（VSS）则对连续视频帧进行语义分割，并利用时序上下文。现有 VSS 方法大致可分为高效 VSS 与高性能 VSS 两类：前者通过特征复用以精度换取速度；后者采用计算开销较大的逐帧分割网络，以提升当前帧分割质量。本文方法属于后者，并通过消失点先验引导 VSS，在性能与模型复杂度之间实现更优平衡。

2.1. 高效 VSS

高效 VSS 旨在降低计算开销、提升分割效率 [17, 19, 24, 28, 34, 43, 56, 65]。其中最典型的思路是关键帧方法：模型仅在关键帧上运行高开销的特征提取与分割网络，非关键帧特征则由关键帧特征调整得到，以减少计算量 [19, 24, 28, 34, 56, 65]。例如，Accel [19] 利用光流进行特征对齐 [52]，DFF [65] 通过光流场传播深度特征图，DVSNet [56] 则采用自适应关键帧与区域调度策略。此外，也有工作 [17, 43] 采用特征传播方法，即将历史帧特征复用于当前帧以加速计算。这类方法虽然通过特征共享与传播提升了分割效率，但依赖替代特征往往会导致分割结果不准确。

2.2. 高性能 VSS

高性能 VSS 侧重于利用输入视频的时序连续性提升分割精度 [25, 31, 35, 38, 45, 46]。与高效 VSS 不同，这类方法采用逐帧网络，即对每一帧均运行计算开销较大的完整分割网络，并通过挖掘视频帧之间的时序相关性来增强当前帧的分割质量。例如，Liu 等人 [31] 利用知识蒸馏将帧间时序一致性作为额外约束，以获得更鲁棒的 VSS；Sun 等人 [46] 通过估计跨帧亲和度来增强时序信息聚合。近期工作的另一趋势 [38, 45, 46] 是引入注意力机制 [49]，使模型能够动态关注视频序列

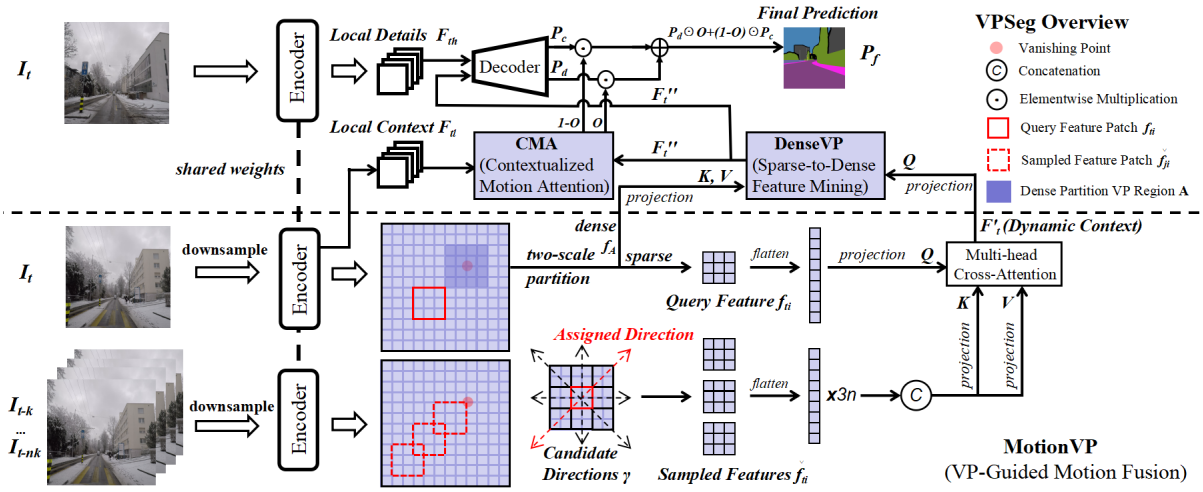


图 2. **VPSeg 网络概述**。在 MotionVP 模块（图中下半部分）中，视频帧经下采样后提取上下文特征，再通过交叉注意力得到动态上下文 F'_t；随后，DenseVP 通过双尺度划分策略进一步增强 F'_t，挖掘 VP 区域中更细粒度的特征 $f_{\mathbf{A}}$。在框架上部，模型分别从下采样后的目标帧和高分辨率目标帧 I_t 中提取局部上下文 F_{tl} 与局部细节 F_{th}。在 CMA 中，增强后的动态上下文 F''_t 与局部上下文 F_{tl} 交互，生成细节权重图 O，并由该权重图引导动态上下文与高分辨率局部细节 F_{th} 融合，生成最终预测 P_f。

的特定部分，从而更好地利用时序上下文。然而，这类方法计算开销巨大，通常局限于低分辨率输入，因而难以实际应用于高分辨率视频帧分割任务。

2.3. VP 检测

消失点（VP）是透视投影中的几何量，表示三维空间中平行直线在二维图像上的视觉汇聚点。VP 已被用于相机标定 [1]、车道偏离预警 [58]、地图构建 [29] 等多个领域。现有研究已提出多种 VP 检测方法，大致可分为两类：传统手工设计方法与基于深度学习的方法。

传统方法主要包括纹理检测 [37, 44, 67] 和边缘检测 [22, 47]。其中，纹理检测方法先搜索图像中纹理的主导方向，再通过投票确定 VP 位置，计算量较大，难以实时运行。在结构化道路场景中，基于霍夫变换 [33] 的边缘检测方法 [11, 13] 更为常用：其先提取图像中的直线，再映射到霍夫空间，对候选 VP 进行投票。基于深度学习的方法 [3, 55] 主要使用 CNN [23] 直接从原始图像像素预测 VP 位置。然而，缺少自动驾驶专用数据集，加之推理耗时较长，限制了这类方法的实际应用。本文采用基于霍夫变换的边缘检测方法 [11, 33]，以兼顾检测精度与推理速度。

3. VPSeg：面向 VSS 的 VP 引导网络

研究动机。深度图 [8]、布局 [48]、纹理 [53] 等结构化线索已被证明对场景理解至关重要，但其在 VSS 中的应用仍然研究不足。一方面，深度估计等模型在远处区域的表现往往不稳定；另一方面，多数基于深度学习的结构化线索方法计算耗时，难以与其他任务协同使用。

相比之下，VP 检测快速且鲁棒，尤其借助边缘检测技术即可适用于广泛的结构化驾驶场景。此外，根据输入视频各帧中的 VP 位置得到的运动信息可用于捕获跨帧对应关系，因为像素跨帧的表观运动方向通常与其相对于 VP 的偏移方向保持一致（见图 1）。因此，如何利用这种 VP 引导的运动先验，便成为 VSS 中一个尚待探索且极具研究价值的问题。值得注意的是，现有高性能 VSS 方法的计算需求较高。该领域近期的尝试要么依赖昂贵硬件，要么在适用性上局限于低分辨率数据集（如 [35, 45, 46] 所示）。综上，本文进一步提出另一关键问题：如何在保持较高分割精度的同时，更高效地利用视频帧中的时序信息？

本文针对上述两个问题提出了有效解决方案。通过 VP 引导的运动融合（MotionVP），建立显式跨帧对应关系，从相邻帧中提取相关的动态特征；稀疏到稠密特征挖掘（DenseVP）则采用自适应的输入帧划分策略，为 VP 区域内细微且难以分辨的运动挖掘更细粒度的特征。此外，上下文-细节框架采用高低分辨率输入分工：在低分辨率输入上提取上下文特征，在高分辨率输入上提取基于细节的特征；再通过上下文感知运动注意力（CMA）将高分辨率局部预测与下采样上下文预测融合，以降低计算开销。图 2 展示了 VPSeg 的网

络架构，其中集成了上述新提出的模块。

3.1. MotionVP：消失点引导的运动融合

为帧序列中同一目标建立跨帧特征对应关系是 VSS 面临的关键挑战。本文观察到，在典型自动驾驶场景中，静态与动态目标的相对运动通常遵循车道线走向（见图 1）。通常，车辆行驶方向与 VP 方向一致，因此视频中多数目标会随时间沿径向远离 VP。

为将这一 VP 动态先验用于运动估计，本文初始化了四个候选方向，$\gamma = n \cdot \frac{\pi}{4}, n = 0,1,2,3$，对应的向量表示为：$\mathcal{V} = \{(+1,0),(+1,+1),(0,+1),(-1,+1)\}$。假设有一个由 $n+1$ 个视频帧组成的训练样本 $\mathcal{I} = \{I_{t-nk},...,I_{t-2k},I_{t-k},I_t\}$，对应时间戳为 $\mathcal{T} = \{t-nk,...,t-2k,t-k,t\}$，其中 $t-k$ 表示比 t 早 k 帧，k 为帧采样间隔。本文使用指定的 n 个历史帧来增强目标帧 I_t 的语义分割。首先，使用预训练 Transformer 编码器，从双线性下采样后的视频帧 $\mathcal{I}$ 中提取上下文特征图 $\mathcal{F} = \{F_{t-nk},...,F_{t-2k},F_{t-k},F_t\}$，其中 $F \in \mathbb{R}^{c\times h\times w}$，$c,h,w$ 分别表示通道数、高度和宽度。随后，将特征图划分为大小为 $s\times s$ 的特征块。对于索引集合 $\mathcal{D} = \left\{(x_i,y_i) \in \mathbb{N}^2 | x_i < \frac{w}{s}, y_i < \frac{h}{s}\right\}$ 中第 i 个特征块索引 (x_i,y_i)，在第 j 帧中对应的特征块记为 $f_{ji} \in \mathbb{R}^{c\times s^2}$。对每个块 (x_i,y_i)，将从块中心指向 VP 的向量作为其运动方向：

$$(\Delta x_{ji}, \Delta y_{ji}) = (\hat{x}_j - x_i, \hat{y}_j - y_i), \tag{1}$$

其中 $(\hat{x}_j,\hat{y}_j)$ 表示第 j 帧中估计得到的块级 VP 位置。与 $(x_i,y_i) \in \mathcal{D}$ 不同，$(\hat{x}_j,\hat{y}_j)$ 的取值范围为 $[0,\frac{w}{s}-1]\times[0,\frac{h}{s}-1]$。随后，将与该块运动方向最接近的候选方向记为其“分配方向”。第 j 帧中块 (x_i,y_i) 的分配方向 (u_{ji},v_{ji}) 可计算为

$$(u_{ji},v_{ji}) = \underset{(u,v)\in\mathcal{V}}{\text{argmin}}\ \text{dist}((u,v),(\Delta x_{ji},\Delta y_{ji})), \tag{2}$$

其中 dist$(\cdot)$ 用来量化运动方向与各候选方向的差异：

$$\text{dist}((u,v),(\Delta x,\Delta y)) = \mid \alpha(u,v) - \alpha(\Delta x,\Delta y) \mid, \tag{3}$$

式中 $\alpha(u,v) = \text{NumPy.arctan2}(v,u)$。随后，沿分配方向分别向前、向后采样相邻块，以捕获双向对应关系。随着帧间隔 $t-j$ 增大，采样距离 $(u_{ji},v_{ji})'$ 也按采样系数 Δd 线性增大，其中较大的 Δd 适用于更高的车速和更大的帧采样间隔 k：

$$(u_{ji},v_{ji})' = \frac{t-j}{k} \times \Delta d \times (u_{ji},v_{ji}). \tag{4}$$

至此，将第 j 帧中的前向采样块 $(\check{x}^f_{ji},\check{y}^f_{ji})$、后向采样块 $(\check{x}^b_{ji},\check{y}^b_{ji})$ 和局部采样块 $(\check{x}^l_{ji},\check{y}^l_{ji})$ 的特征拼接，得到采样特征 $\check{f}_{ji} \in \mathbb{R}^{c\times 3s^2}$：

$$\begin{aligned}
(\check{x}^f_{ji},\check{y}^f_{ji}) &= (x_i,y_i) + (u_{ji},v_{ji})', \\
(\check{x}^b_{ji},\check{y}^b_{ji}) &= (x_i,y_i) - (u_{ji},v_{ji})', \\
(\check{x}^l_{ji},\check{y}^l_{ji}) &= (x_i,y_i).
\end{aligned} \tag{5}$$

给定当前帧 I_t 中第 i 个块的局部特征 $f_{ti} \in \mathbb{R}^{c\times s^2}$，以及从相邻帧中采样得到的特征集合 $\mathcal{S} = \left\{\check{f}_{ji} \in \mathbb{R}^{c\times 3s^2}, j \in \mathcal{T} \text{且} j \neq t\right\}$，本文以 f_{ti} 作为查询，以 $\mathcal{S}$ 作为键和值进行交互：

$$Q_i = \text{W}_q(f_{ti}), \quad K_i = \text{W}_k(\text{C}(\mathcal{S})), \quad V_i = \text{W}_v(\text{C}(\mathcal{S})), \tag{6}$$

其中，W$(\cdot)$ 和 C$(\cdot)$ 分别表示全连接层和拼接操作。接着使用交叉注意力 CA$(\cdot)$ [49] 计算第 t 帧中特征块 (x_i,y_i) 的块级动态特征 $f'_{ti} \in \mathbb{R}^{c\times s^2}$：

$$f'_{ti} = \text{CA}(Q_i,K_i,V_i). \tag{7}$$

随后将所有块的块级动态特征按原空间位置平铺重组，得到第 t 帧的完整帧级动态特征 $F'_t \in \mathbb{R}^{c\times h\times w}$。

综上所述，VP 引导的运动估计通过对运动估计施加约束，建立了显式的特征对应关系。得益于逐步增大的采样距离 $(u_{ji},v_{ji})'$，这一快速且简洁的算法同样适用于高速场景。需要说明的是，MotionVP 模块使用低分辨率帧，目的是优先提取上下文特征而非高分辨率细节，从而获得更全面的动态上下文理解。随后，动态上下文通过上下文感知运动注意力与高分辨率细节特征融合（见第 3.3 节），得到最终的语义预测。

3.2. DenseVP：稀疏到稠密特征挖掘

VP 附近的目标通常尺寸微小，且帧间运动幅度极小。为此，本文提出尺度自适应的 VP 引导特征增强模块，即稀疏到稠密特征挖掘（DenseVP）。简言之，本文将稠密划分与稀疏划分相结合，为 VP 附近通常细微且不易分辨的运动挖掘更细粒度的特征；而在帧的其余部分则采用更稀疏、计算效率更高的特征块划分方式。由于 VP 检测存在不稳定性，本文将 VP 周围的区域称为 VP 区域，并将其作为一种更粗略但更鲁棒的替代方案。具体而言，先根据 VP 的位置确定 VP 区域。沿用第 3.1 节中的符号，特征图包含 $\frac{h}{s}\times\frac{w}{s}$ 个特征块，第 i 个特征块的索引为 (x_i,y_i)，第 j 帧中估计的块级 VP 位置为 $(\hat{x}_j,\hat{y}_j)$。VP 块 (x'_j,y'_j) 表示距离 VP 最近的特征块，定义为

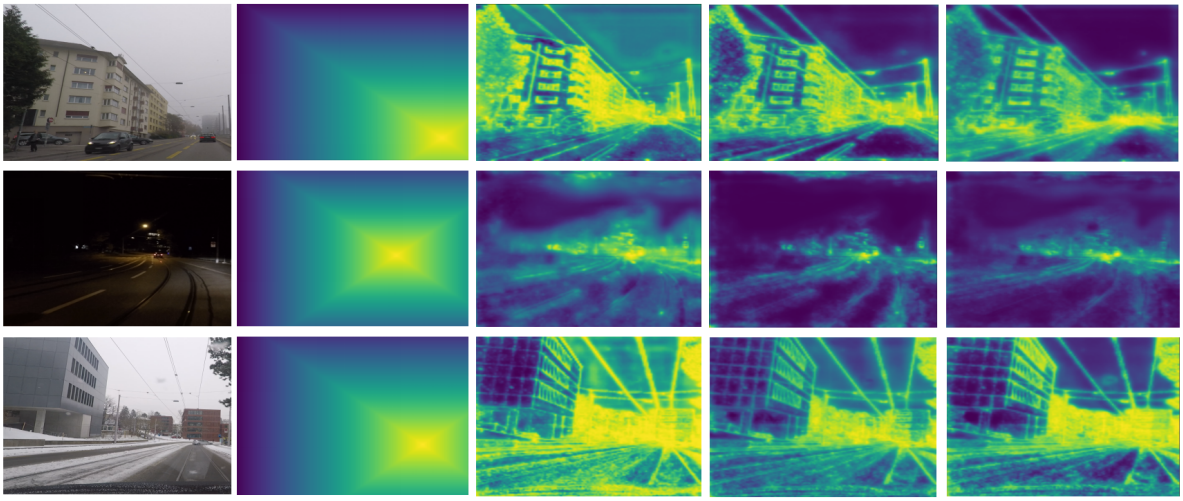
$$(x'_j,y'_j) = \underset{(x,y)\in\mathcal{D}}{\text{argmin}}\ [(x-\hat{x}_j)^2 + (y-\hat{y}_j)^2]. \tag{8}$$

VP 区域 $\mathcal{A} \subset \mathcal{D}$ 是 VP 块附近的矩形区域，包含按稀疏网格排列的 $(2a+1)\times(2b+1)$ 个块，且 $a,b \in \mathbb{N}$：

$$\mathcal{A} = \left\{(m,n) | (m-x'_j)^2 \le a^2\ ,\ (n-y'_j)^2 \le b^2\right\}. \tag{9}$$

随后，在 VP 区域内采用重叠稠密网格划分。设置步长为 $\lceil s/2 \rceil$，得到 m 个重叠块，其中 $m = (\lfloor \frac{2as}{\lceil s/2 \rceil} \rfloor + 1)(\lfloor \frac{2bs}{\lceil s/2 \rceil} \rfloor + 1)$。随后，将动态上下文 F'_t 作为查询，将从 VP 区域稠密提取的特征 $f_{\mathcal{A}} \in \mathbb{R}^{\,c\times ms^2}$ 作为键和值：

$$Q_{\mathcal{A}} = \text{W}_q(F'_t), \quad K_{\mathcal{A}} = \text{W}_k(f_{\mathcal{A}}), \quad V_{\mathcal{A}} = \text{W}_v(f_{\mathcal{A}}), \tag{10}$$



(a) 当前帧 I_t (b) VP 邻近图 E (c) $N=0$ (d) $N=1$ (e) $N=2$

图 3. **CMA 中包含 N 层运动注意力时的细节权重图 O 可视化**。随着 N 增大，细节权重图与动态特征的交互增强；场景近处区域或简单语义类别对应的权重逐渐降低。靠近 VP 的高亮远处区域表明，这些区域的最终预测 P_f 主要基于细节预测 P_d，而非 P_c。VP 邻近图作为位置先验，辅助模型精确定位这些远处区域。

其中，W(·) 表示全连接层。随后通过交叉注意力操作 CA(·) [49] 用稠密特征 $f_{\mathcal{A}}$ 增强动态上下文 F'_t，以获取 VP 区域运动的更细粒度表示：

$$F''_t = \mathrm{CA}(Q_{\mathcal{A}}, K_{\mathcal{A}}, V_{\mathcal{A}}), \quad F''_t \in \mathbb{R}^{c\times h\times w}. \tag{11}$$

上述双尺度特征划分利用静态 VP 位置信息，增强 VP 区域内的动态上下文；该区域通常包含远处目标。因此，DenseVP 利用与帧中 VP 位置相关的静态先验，MotionVP 则利用与目标表观运动相关的动态 VP 先验；表观运动取决于目标相对 VP 的位置。

3.3. CMA：上下文感知运动注意力

给定增强后的动态上下文 $F''_t \in \mathbb{R}^{c\times h\times w}$，本文的目标是将其与高分辨率细节特征融合，以生成最终预测。对于目标帧 I_t，低分辨率输入和高分辨率输入分别记为 I_{tl} 与 I_{th}；对应的局部上下文特征和局部细节特征分别为 $F_{tl} \in \mathbb{R}^{c\times h\times w}$ 与 $F_{th} \in \mathbb{R}^{c\times h\times w}$。首先，随机初始化可学习的查询 $Q \in \mathbb{R}^{c\times K}$（$K$ 为类别数），再通过 VP 感知的交叉注意力 $\mathrm{CA}_E(\cdot)$，利用 F_{tl} 将其转化为上下文感知查询：

$$Q_c = \mathrm{CA}_E(\mathrm{W}_q(Q), \mathrm{W}_k(F_{tl}), \mathrm{W}_v(F_{tl})), \tag{12}$$

其中 W(·) 表示全连接层，$Q_c \in \mathbb{R}^{c\times K}$ 为融入上下文后的查询。$\mathrm{CA}_E(\cdot)$ 定义为

$$\mathrm{CA}_E(Q,K,V) = \mathrm{Softmax}(\frac{QK^T}{\sqrt{c}} + E)V + Q. \tag{13}$$

其中，E 表示 VP 邻近图嵌入。它是以 VP 为中心的伪深度图：像素 (x,y) 的深度值为 $1-\Delta D$，其中 $\Delta D \propto \max\{\frac{|y-\hat{y}_j^p|}{h}, \frac{|x-\hat{x}_j^p|}{w}\}$，$(\hat{x}_j^p, \hat{y}_j^p)$ 为 VP 的像素坐标；随后通过运动注意力融合局部上下文与动态上下文：

$$F_m = \mathrm{CA}_E(\mathrm{W}_q(Q_c), \mathrm{W}_k(F''_t), \mathrm{W}_v(F''_t)), \tag{14}$$

其中，融合后的上下文为 $F_m \in \mathbb{R}^{c\times K}$。细节权重图 $O \in \mathbb{R}^{K\times h\times w}$ 计算为：

$$O = F_m^T F_{tl}. \tag{15}$$

最终预测 P_f 在细节权重图 O 的引导下，由上下文预测 P_c 与细节预测 P_d 加权融合得到：

$$P_f = (1-O)\odot P_c + O \odot P_d, \tag{16}$$

其中，$\odot$ 表示逐元素乘法，P_c 和 P_d 分别由 F_{tl} 与 F_{th} 经过 DAFormer [16] 解码器生成。对于不同语义类别和不同空间位置，O 学习以不同方式权衡高分辨率细节与动态上下文。如图 3所示，远处区域中 P_d 的权重通常更高，表明这些区域的最终预测 P_f 主要由细节预测 P_d 决定；而在较近区域和简单语义类别中，最终预测主要依赖上下文预测 P_c。VP 邻近图作为位置先验嵌入，引导细节权重图 O 更侧重这些深处区域。VPSeg 的总损失定义为：

$$\mathcal{L}_{\text{total}} = (1-\lambda_d)\mathcal{L}_{\text{CE}}(P_f, G) + \lambda_d \mathcal{L}_{\text{CE}}(P_d, G), \tag{17}$$

其中，G 表示真实标签，λ_d 表示细节预测的损失系数。最终损失由融合预测和细节预测对应的交叉熵损失加权求和得到，该设计在优化特征融合的同时，也有助于增强高分辨率细节表征。

4. 实验

4.1. 实验设置

实现细节。本文端到端模型采用 AdamW [32] 优化器训练 160k 次迭代，批量大小设置为 4，初始学习率

为 2×10^{-4}，细节预测损失权重设为 $\lambda_d = 0.1$。沿用 SegFormer [54] 的设置，采用在 ImageNet [41] 上预训练的 MiT [54] 作为骨干网络。为保持方法的通用性，分割当前帧时只引入历史帧，帧采样间隔设为 $k = 3$。数据增强策略包括随机缩放、翻转、裁剪和光度扰动。与其他仅使用低分辨率输入的高性能 VSS 模型 [35, 45, 46] 不同，VPSeg 使用 1024×2048 分辨率输入，并对输入按 0.5 的比例进行双线性下采样得到上下文特征。

VP 估计采用 Canny 边缘检测，阈值设置为 50 和 150，孔径大小为 3。MotionVP 的采样系数 Δd 设置为 1。实验表明该数值足以覆盖快速运动目标，并取得良好性能。在 DenseVP 中，VP 区域设为 3×3 的稀疏块区域（$a = 1, b = 1$）。推理阶段采用单尺度滑动窗口测试。所有实验均在 4 块 NVIDIA RTX 3090 GPU 上完成。

数据集。本文实验主要在两个自动驾驶数据集上进行：ACDC [42] 和 Cityscapes [9]。ACDC 包含大量图像（训练集 1,600 段、验证集 406 段），并均匀覆盖雾天、夜晚、雨天和雪天四类常见恶劣条件。更重要的是，ACDC 采用两阶段标注策略，并生成二值“无效掩码”以标记图像中的模糊区域：首先人工给出初始语义标签，随后借助恶劣条件视频对其进行细化，最终完成标注，从而支持对不确定区域分割性能的有效量化评估。此外，本文还在 Cityscapes 视频数据集上进行了实验，该数据集训练集和验证集包含来自 21 个城市的 3,475 个视频片段。

评估指标。实验结果通过以下三项指标进行评估：mIoU、iIoU [9] 和 IA-IoU。由于 mIoU 倾向于降低同一类别内小实例的权重，本文进一步使用实例级交并比（iIoU），以更充分地评估模型在微小目标上的性能，其定义为：

$$i\text{IoU} = \frac{i\text{TP}}{i\text{TP} + i\text{FN} + \text{FP}}, \tag{18}$$

其中，TP、FN 和 FP 分别表示真正例、假负例和假正例。iIoU 会为较小实例中的像素赋予更高权重 i。

针对 ACDC [42]，本文提出了一项新指标：无效区域交并比（IA-IoU）。如前文所述，ACDC 数据集中的无效掩码能够揭示图像中的模糊区域，也就是所谓的不确定区域，具有重要的评估价值。因此本文专门在这些无效区域内计算 mIoU，作为 IA-IoU。具体来说，假设共有 K 个语义类别和 n_{max} 张验证图像。对于第 n 张图像对应的无效掩码 M_n，有：

$$\hat{P}_{nz} = P_{nz} \cap M_n, \tag{19}$$

其中，P_{nz} 表示模型对第 n 张图像中第 z 个类别的预测。同理，对真实标签进行处理：

$$\hat{G}_{nz} = G_{nz} \cap M_n, \tag{20}$$

其中，G_{nz} 表示第 n 张图像中第 z 个类别的真实标签。第 z 个类别的 IA-IoU 可通过下式计算：

$$\text{IA-IoU}_z = \frac{\sum_{n=0}^{n_{max}} |\hat{P}_{nz} \cap \hat{G}_{nz}|}{\sum_{n=0}^{n_{max}} |\hat{P}_{nz} \cup \hat{G}_{nz}|}. \tag{21}$$

平均 IA-IoU（mIA-IoU）为各类别 IA-IoU 的平均值。该指标专门评估真实标签中标记为不确定或模糊的区域。

4.2. 与 SOTA 方法的比较

在表 1 中，本文在 ACDC 和 Cityscapes 数据集上将 VPSeg 与 SOTA 方法进行了性能比较，定性结果见图 4。结果表明，本文方法能够更准确地捕捉难以分辨或快速的运动，从而实现更鲁棒的分割。

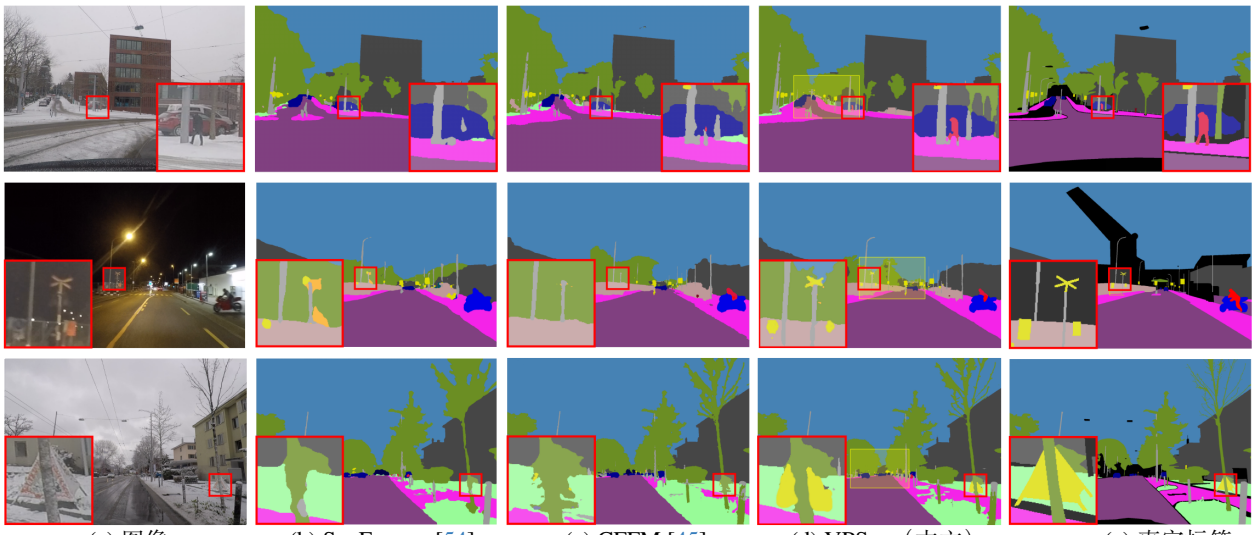
在 ACDC 上，与基线模型 SegFormer [54] 和 SOTA VSS 模型 MRCFA [46] 相比，VPSeg 的 mIoU 分别提升 2.10% 和 1.85%。在更侧重模糊区域性能的 mIA-IoU 指标上，提升效果更为显著：模型相较 SegFormer 和 MRCFA 分别实现了 3.06% 和 2.67% 的性能提升。表 2 进一步展示了不同模型在选定类别上的 IA-IoU 性能：对于骑行者（+8.5%）和交通标志（+3.4%）等难分割目标，VPSeg 取得了显著的性能提升；对于天空（+0.2%）和公交车（+0.1%）等分割区域较大的简单样本，也能与对比方法性能相当。表 1 还提供了 miIoU 的结果，其中 VPSeg 相比 SOTA 方法 CFFM [45] 提升 2.11%，表明该模型更擅长处理远处难分割小目标。

除 ACDC 外，VPSeg 在 Cityscapes 上也取得了 SOTA 性能。与基线模型 SegFormer 和 VSS 模型 CFFM 相比，本文的 mIoU 分别提升 1.14% 和 1.02%。在 miIoU 上，提升幅度进一步扩大至 +1.78% 和 +1.70%，验证了该方法在微小目标分割上的鲁棒性。无论采用 MiT-B1 [54] 还是 MiT-B3 [54] 骨干网络，VPSeg 均明显优于相应的基线模型，表明本文所提出的模块具有通用性。

如表 3所示，本文方法在资源效率上同样表现突出。相较于使用 MiT-B3 作为骨干网络的 SegFormer，VPSeg 仅额外增加了 2.2M 参数，占 SegFormer 总参数量（44.6M）的 4.9%。与 Video K-Net [27]、MRCFA 等高性能 VSS 模型相比，本文方法以 1124.7 GFLOPs 实现了更高的计算效率。尽管受限于 VP 推理速度（见表 5），VPSeg 的 FPS 仍与其他方法相当。总体而言，VPSeg 仅带来有限的额外计算开销，却取得了 SOTA 分割性能。

4.3. 消融实验

DenseVP 的作用。 在表 4中，本文对不同设置下的稀疏到稠密特征挖掘模块（DenseVP）进行了消融实验。



(a) 图像 (b) SegFormer [54] (c) CFFM [45] (d) VPSeg（本文） (e) 真实标签

图 4. **ACDC 上的定性对比**。黄框表示采用稠密划分的 VP 区域。对于 VP 附近的远处微小难分割目标，以及近处被遮挡的快速移动目标，本文方法均给出更准确的预测。

方法	骨干网络	参数量 (M)↓	mIoU↑		miIoU↑		mIA-IoU↑	VSS
			ACDC	Cityscapes	ACDC	Cityscapes	ACDC	
DeepLabv3+[7]	ResNet-101	62.7	72.79	79.09	43.14	56.89	36.32	✗
PSPNet [62]	ResNet-101	68.0	72.26	78.34	40.95	56.78	37.29	✗
OCRNet [60]	ResNet-101	55.6	70.39	80.09	43.36	59.55	33.38	✗
SeaFormer [50]	SeaFormer-L	14.0	70.09	77.70	40.41	56.96	33.26	✗
SegFormer [54]	MiT-B1	**13.7**	70.25	78.56	41.14	58.25	34.05	✗
SegFormer [54]	MiT-B3	44.6	75.38	81.32	46.51	60.01	38.42	✗
Video K-Net [27]	Swin-B	104.6	69.03	76.62	39.33	56.02	31.67	✓
ETC [31]	ResNet-101	68.1	71.45	79.50	42.28	58.35	36.71	✓
TCB [35]	ResNet-101	72.5	70.56	78.70	41.76	57.84	35.96	✓
NetWarp [52]	PSPNet	90.6	73.71	80.60	45.59	59.63	36.58	✓
CFFM [45]	MiT-B3	49.6	75.47	81.44	47.31	60.09	37.88	✓
MRCFA [46]	MiT-B3	48.2	75.63	81.31	46.28	60.56	38.81	✓
VPSeg （本文）	MiT-B1	14.9	72.86	79.56	43.42	59.53	37.96	✓
VPSeg （本文）	MiT-B3	46.8	**77.48**	**82.46**	**49.42**	**61.79**	**41.48**	✓

表 1. **ACDC [42] 与 Cityscapes [9] 验证集上的方法对比**。本模型在 mIoU、miIoU [9] 与 mIA-IoU 上均取得最佳结果。

	行人	骑行者	汽车	公交车	植被	自行车	交通标志	天空	道路
DeepLabv3+ [7]	48.2	18.2	41.3	40.1	69.6	26.1	35.0	84.7	67.7
SegFormer [54]	47.9	19.0	44.6	45.9	71.1	35.4	32.6	85.4	**74.8**
SeaFormer [50]	41.2	10.6	36.2	41.5	70.1	29.7	29.4	84.5	62.9
ETC [31]	45.8	15.5	41.1	33.5	**71.6**	29.6	34.6	84.5	70.6
CFFM [45]	48.6	21.4	**46.6**	46.5	70.8	32.8	31.9	84.6	74.2
MRCFA [46]	47.8	24.3	46.1	46.1	71.3	35.7	34.1	85.1	74.7
VPSeg （本文）	**51.8**	**32.8**	46.3	**46.6**	71.5	**39.3**	**38.4**	**85.6**	74.5

表 2. **表 1 中各方法在 ACDC 上的各类别 IA-IoU**。SegFormer 与 VPSeg 均采用 MiT-B3 [54] 作为骨干网络。

随着 VP 区域扩大，模型性能逐步提升，并在 VP 区域由 3×3（$a = 1, b = 1$）个特征块组成时达到峰值。相比 mIoU，IA-IoU 对 DenseVP 更为敏感，表明其尺度自适应划分更有助于提升不确定区域的分割性能。

方法	骨干网络	参数量 (M)↓	mIoU↑	GFLOPs↓	FPS↑
Video K-Net [27]	Swin-B	104.6	69.03	1430.0	-
TMANet [51]	ResNet-101	54.2	71.62	1385.9	2.4
ETC [31]	ResNet-101	68.1	71.45	-	1.2
TCB [35]	ResNet-101	72.5	70.56	-	1.9
MRCFA [46]	MiT-B3	48.2	75.63	1436.4	**4.4**
CFFM [45]	MiT-B3	49.6	75.47	1534.8	3.8
VPSeg （本文）	MiT-B3	**46.8**	**77.48**	**1124.7**	3.4

表 3. 高性能 VSS 方法在 ACDC 上的 FPS 与 GFLOPs 对比。

不同位置嵌入的影响。 由于 VP 邻近图本质上是一种伪深度图，本文尝试将 CMA 中的 VP 邻近图嵌入替换为深度图嵌入。实验结果见表 5，其中深度图由 MiDaS V3-Hybrid [40] 生成。由于深度估计在远处区域性能不

VP 区域大小	mIoU (A.)↑	mIA-IoU (A.)↑	mIoU (C.)↑
不使用 DenseVP	77.15	41.02	82.19
$a=0, b=0$	77.28	40.87	82.34
$a=1, b=1$	**77.48**	**41.48**	82.46
$a=2, b=2$	77.36	41.33	82.37
$a=3, b=3$	77.41	41.32	**82.50**

表 4. 采用 MiT-B3 骨干网络时，VP 区域大小在 ACDC（A.）和 Cityscapes（C.）上的消融实验。

位置嵌入	mIoU (A.)↑	mIA-IoU (A.)↑	mIoU (C.)↑	FPS↑
无位置嵌入	77.02	40.91	81.16	**4.2**
深度图嵌入	77.19	40.78	82.23	1.8
VP 邻近图嵌入	**77.48**	**41.48**	**82.46**	3.4
深度图嵌入 + VP 邻近图嵌入	77.46	41.37	82.39	1.6

表 5. 采用 MiT-B3 骨干网络时，不同位置嵌入在 ACDC（A.）和 Cityscapes（C.）上的消融实验。

方法 / k	3	5	7	9
SegFormer [54]	75.38	75.38	75.38	75.38
MRCFA [46]	75.63	75.44	75.32	75.30
CFFM [45]	75.47	75.35	75.22	75.07
VPSeg （本文）	**77.48**	**77.52**	**77.39**	**77.43**

表 6. 采用 MiT-B3 骨干网络时，帧采样间隔 k（默认值为 3）在 ACDC 上的消融实验。

稳定，深度图嵌入对 mIoU 的提升有限，且会导致 IA-IoU 下降。此外，MiDaS 推理耗时较长，导致 FPS 显著降低。单独使用 VP 邻近图嵌入可在中等推理速度下取得最佳的 mIoU 和 mIA-IoU；同时使用深度图与 VP 邻近图嵌入时，性能与单独使用 VP 邻近图嵌入相当，但整体 FPS 仍受 MiDaS 推理开销限制。

帧采样间隔 k 的影响。 表 6 研究了帧采样间隔 k 的影响。值得注意的是，当 k 从 3 增加到 5 时，VPSeg 的 mIoU 指标进一步提升。对于更大的 k，其他方法的 mIoU 显著下降，而 VPSeg 仍表现出更强的鲁棒性。可能的原因是，沿 VP 方向进行运动搜索能够覆盖更大的运动范围，因此尤其适用于高速场景。

运动注意力层数 N 的影响。 表 7 展示了 VPSeg 的性能随 CMA 中运动注意力层数 N 的变化情况。随着 N 增大，模型性能显著提升，验证了动态特征融合的有效性。当 $N \geq 2$ 时，性能增益趋于饱和。因此，对于使用 MiT-B3 骨干网络的 VPSeg，本文选择 $N=2$，以在性能与模型复杂度之间取得最佳平衡。

5. 结论

建立跨帧对应关系和降低计算开销，是驾驶场景 VSS 中亟待解决的两个问题。针对这些问题，本文从 VP 先验出发，提出了一种新颖且高效的 VSS 网络 VPSeg，

N	mIoU (A.)↑	mIA-IoU (A.)↑	mIoU (C.)↑	参数量 (M)↓
0	76.65	40.33	82.12	**45.9**
1	77.21	41.23	82.33	46.4
2	77.48	**41.48**	**82.46**	46.8
3	**77.51**	41.39	82.44	47.3

表 7. 采用 MiT-B3 骨干网络时，CMA 中运动注意力层数的消融实验。

通过静态与动态 VP 先验协同提升分割性能。具体而言，DenseVP 采用尺度自适应划分方式进行稀疏到稠密的特征挖掘，以增强靠近估计 VP 的区域中的精细动态特征；MotionVP 则通过 VP 引导的运动估计建立显式跨帧对应关系。与此同时，VPSeg 采用上下文-细节框架，以低分辨率输入提取下采样上下文、以高分辨率输入保留局部细节，并进一步通过 CMA 自适应融合二者。大量实验结果表明，VPSeg 在 Cityscapes 和 ACDC 两个主流驾驶场景分割基准上均取得了优于 SOTA 方法的性能，同时仅引入适度的额外计算开销。

6. 致谢

本研究由丰田欧洲汽车公司 TRACE-Zürich 研究项目提供经费支持。

参考文献

[1] Michel Antunes, João P. Barreto, Djamila Aouada, and Björn Ottersten. Unsupervised vanishing point detection and camera calibration from a single manhattan image with radial distortion. In *CVPR*, 2017. 3

[2] John Canny. A computational approach to edge detection. *IEEE TPAMI*, PAMI-8(6):679–698, 1986. 12, 13

[3] Chin-Kai Chang, Jiaping Zhao, and Laurent Itti. Deepvp: Deep learning for vanishing point detection on one million street view images. In *IEEE ICRA*, 2018. 3

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *ICLR*, 2015. 2

[5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous con-

volution, and fully connected CRFs. *IEEE TPAMI*, 40(4): 834–848, 2018. 2

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 7

[8] Po-Yi Chen, Alexander H. Liu, Yen-Cheng Liu, and Yu-Chiang Frank Wang. Towards scene understanding: unsupervised monocular depth estimation with semantic-aware representation. In *CVPR*, 2019. 3

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 6, 7, 12

[10] Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qun Liu, and Gang Wang. Context contrasted feature and gated multi-scale aggregation for scene segmentation. In *CVPR*, 2018. 2

[11] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *CACM*, 15 (1):11–15, 1972. 3

[12] Deng-Ping Fan, Ge-Peng Ji, Peng Xu, Ming-Ming Cheng, Christos Sakaridis, and Luc Van Gool. Advances in deep concealed scene understanding. *VI*, 1(1):16, 2023. 1

[13] Yasutaka Furukawa and Yoshihisa Shinagawa. Accurate and robust line segment extraction by analyzing distribution around peaks in hough space. *CVIU*, 92(1):1–25, 2003. 3

[14] Diandian Guo, Deng-Ping Fan, Tongyu Lu, Christos Sakaridis, and Luc Van Gool. Vanishing-point-guided video semantic segmentation of driving scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3544–3553, 2024. 1

[15] Junjun He, Zhongying Deng, Lei Zhou, Yali Wang, and Yu Qiao. Adaptive pyramid context network for semantic segmentation. In *CVPR*, 2019. 2

[16] Lukas Hoyer, Dengxin Dai, and Luc Gool. Daformer: improving network architectures and training strategies for domain-adaptive semantic segmentation. In *CVPR*, 2022. 5

[17] Ping Hu, Fabian Caba, Oliver Wang, Zhe Lin, Stan Sclaroff, and Federico Perazzi. Temporally distributed networks for fast video semantic segmentation. In *CVPR*, 2020. 2

[18] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. CCNet: Criss-cross attention for semantic segmentation. In *ICCV*, 2019. 2

[19] Samvit Jain, Xin Wang, and Joseph E Gonzalez. Accel: A corrective fusion network for efficient semantic segmentation on video. In *CVPR*, 2019. 2

[20] Zhenchao Jin, Tao Gong, Dongdong Yu, Qi Chu, Jian Wang, Changhu Wang, and Jie Shao. Mining contextual information beyond image for semantic segmentation. In *ICCV*, 2021. 2

[21] Zhenchao Jin, Bin Liu, Qi Chu, and Nenghai Yu. ISNet: Integrate image-level and semantic-level context for semantic segmentation. In *ICCV*, 2021. 2

[22] Jana Košecká and Wei Zhang. Efficient computation of vanishing points. In *IEEE ICRA*, 2002. 3

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 25, 2012. 3

[24] Shih-Po Lee, Si-Cun Chen, and Wen-Hsiao Peng. GSVNet: Guided spatially-varying convolution for fast semantic segmentation on video. In *ICME*, 2021. 2

[25] Jiangtong Li, Wentao Wang, Junjie Chen, Li Niu, Jianlou Si, Chen Qian, and Liqing Zhang. Video semantic segmentation via sparse temporal transformer. In *ACM MM*, 2021. 1, 2

[26] Xia Li, Yibo Yang, Qijie Zhao, Tiancheng Shen, Zhouchen Lin, and Hong Liu. Spatial pyramid based graph reasoning for semantic segmentation. In *CVPR*, 2020. 2

[27] Xiangtai Li, Wenwei Zhang, Jiangmiao Pang, Kai Chen, Guangliang Cheng, Yunhai Tong, and Chen Change Loy. Video k-net: A simple, strong, and unified baseline for video segmentation. In *CVPR*, 2022. 6, 7

[28] Yule Li, Jianping Shi, and Dahua Lin. Low-latency video semantic segmentation. In *CVPR*, 2018. 2

[29] Hyunjun Lim, Yeeun Kim, Kwangik Jung, Sumin Hu, and Hyun Myung. Avoiding degeneracy for monocular visual slam with point and line features. In *IEEE ICRA*, 2021. 3

[30] Jianbo Liu, Junjun He, Yu Qiao, Jimmy S Ren, and Hongsheng Li. Learning to predict context-adaptive convolution for semantic segmentation. In *ECCV*, 2020. 2

[31] Yifan Liu, Chunhua Shen, Changqian Yu, and Jingdong Wang. Efficient semantic video segmentation with per-frame inference. In *ECCV*, 2020. 2, 7

[32] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2017. 5

[33] Evelyne Lutton, Henri Maître, and Jaime Lopez Krahe. Contribution to the determination of vanishing points using hough transform. *IEEE TPAMI*, 16(4):430–438, 1994. 3, 12, 13

[34] Behrooz Mahasseni, Sinisa Todorovic, and Alan Fern. Budget-aware deep semantic video segmentation. In *CVPR*, 2017. 2

[35] Jiaxu Miao, Yunchao Wei, Yu Wu, Chen Liang, Guangrui Li, and Yi Yang. VSPW: A large-scale dataset for video scene parsing in the wild. In *CVPR*, 2021. 1, 2, 3, 6, 7

[36] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: a survey. *IEEE TPAMI*, 44 (7):3523–3542, 2022. 2

[37] Peyman Moghadam, Janusz A. Starzyk, and W. Sardha Wijesoma. Fast vanishing-point detection in unstructured environments. *IEEE TIP*, 21(1):425–430, 2012. 3

[38] Matthieu Paul, Martin Danelljan, Luc Van Gool, and Radu Timofte. Local memory attention for fast video semantic segmentation. In *IEEE IROS*, 2021. 2

[39] C. Jeremy Pye and J. A. Bangham. A fast algorithm for morphological erosion and dilation. In *EUSIPCO*, 1996. 12, 13

[40] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: mixing datasets for zero-shot cross-dataset transfer. *IEEE TPAMI*, 44(3), 2022. 7

[41] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 6

[42] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *ICCV*, 2021. 6, 7, 12

[43] Evan Shelhamer, Kate Rakelly, Judy Hoffman, and Trevor Darrell. Clockwork convnets for video semantic segmentation. In *ECCV*, 2016. 2

[44] Jinjin Shi, Jinxiang Wang, and Fangfa Fu. Fast and robust vanishing point detection for unstructured road following. *IEEE TITS*, 17(4):970–979, 2016. 3

[45] Guolei Sun, Yun Liu, Henghui Ding, Thomas Probst, and Luc Van Gool. Coarse-to-fine feature mining for video semantic segmentation. In *CVPR*, 2022. 1, 2, 3, 6, 7, 8

[46] Guolei Sun, Yun Liu, Hao Tang, Ajad Chhatkuli, Le Zhang, and Luc Van Gool. Mining relations among cross-frame affinities for video semantic segmentation. In *ECCV*, 2022. 1, 2, 3, 6, 7, 8

[47] Thorsten Suttorp and Thomas Bücher. Robust vanishing point estimation for driver assistance. In *IEEE ITSC*, 2006. 3

[48] Arces Talavera, Daniel Stanley Tan, Arnulfo Azcarraga, and Kai-Lung Hua. Layout and context understanding for image synthesis with scene graphs. In *ICIP*, 2019. 3

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2, 4, 5, 14

[50] Qiang Wan, Zilong Huang, Jiachen Lu, Gang Yu, and Li Zhang. Seaformer: Squeeze-enhanced axial transformer for mobile semantic segmentation. In *ICLR*, 2023. 7

[51] Hao Wang, Weining Wang, and Jing Liu. Temporal memory attention for video semantic segmentation. In *ICIP*, 2021. 7

[52] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 2, 7

[53] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 3

[54] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021. 6, 7, 8, 13

[55] Li Xingxin, Liqiang Zhu, Yu Zujun, and Wan Yanqin. Adaptive auxiliary input extraction based on vanishing point detection for distant object detection in high-resolution railway scene. In *IEEE ICEMI*, 2019. 3

[56] Yu-Syuan Xu, Tsu-Jui Fu, Hsuan-Kung Yang, and Chun-Yi Lee. Dynamic video segmentation network. In *CVPR*, 2018. 2

[57] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. DenseASPP for semantic segmentation in street scenes. In *CVPR*, 2018. 2

[58] Ju Han Yoo, Seong-Whan Lee, Sung-Kee Park, and Dong Hwan Kim. A robust lane detection method based on vanishing point estimation using the relevance of line segments. *IEEE TITS*, 18(12):3254–3266, 2017. 3

[59] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 2

[60] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *ECCV*, 2020. 7

[61] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *CVPR*, 2018. 2

[62] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 7

[63] Mingmin Zhen, Jinglu Wang, Lei Zhou, Shiwei Li, Tianwei Shen, Jiaxiang Shang, Tian Fang, and Long Quan. Joint semantic segmentation and boundary detection using iterative pyramid contexts. In *CVPR*, 2020. 2

[64] Yizhou Zhou, Xiaoyan Sun, Zheng-Jun Zha, and Wenjun Zeng. Context-reinforced semantic segmentation. In *CVPR*, 2019. 2

[65] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *CVPR*, 2017. 1, 2

[66] Zhen Zhu, Mengde Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation. In *ICCV*, 2019. 2

[67] Zhaozi Zu, Yingtuan Hou, Dixiao Cui, and Jianru Xue. Real-time road detection with image texture analysis-based vanishing point estimation. In *IEEE PIC*, 2015. 3

A. 消失点检测

本文采用经典的消失点（VP）检测流程：先执行 Canny 边缘检测 [2]，再在边缘图上执行霍夫变换 [33]。算法 1 给出了完整流程。对于给定高度为 H、宽度为 W 的灰度图像 x，VP 估计步骤如下：

预处理。为增强边缘检测的鲁棒性，先对输入图像进行形态学开运算（腐蚀后膨胀）[39] 去噪，再使用 Canny 边缘检测生成边缘图。由于“天空”区域位于图像的上部，本文仅对图像底部 2/3 的区域进行 Canny 边缘检测。接着对边缘应用霍夫变换，得到候选直线集合；集合中每条直线 ℓ 的斜率（$\Delta H/\Delta W$）记为 slope(ℓ)。

直线筛选。得到霍夫直线 [33] 后，需要决定保留或舍弃哪些直线，本文设计了一个基于直线靠近 VP 可能性的筛选准则。由于 VP 通常位于图像中心附近，因此首先删除与图像中心距离超过 $d_{\max} = 160$ 像素的直线。此外，大多数水平线或垂直线（如树木、电线）对 VP 检测无贡献。因此，删除霍夫直线集合中斜率 slope(ℓ) $\notin S$ 的任何直线 ℓ，其中 $S = (-5, -0.2) \cup (0.2, 5)$ 是预先设定的斜率保留区间。

单元投票。剔除无关直线后，计算所有直线对的交点。若筛选后剩余 N_{line} 条直线，理论上可得到 $N_{\text{line}}(N_{\text{line}}-1)/2$ 个交点，记为 R。当直线数量过多时，则从中随机采样 100 条直线。随后将图像划分为多个矩形单元，统计每个单元内的霍夫直线交点数。实际操作中仅处理图像下部的单元，以避开天空区域。最后返回交点数最多单元的中心作为 VP 估计位置。

获得 VP 后，仍需将 VP 位置信息传递给模型。由于预处理流程中包含裁剪操作，本文将 VP 邻近图与输入帧同步裁剪，并将裁剪后的输入帧与邻近图拼接，作为新的模型输入。

基于霍夫变换的 VP 检测在自动驾驶场景中快速且鲁棒。然而，处理边缘杂乱或模糊的图像时仍存在挑战。例如，十字路口场景可能出现多个 VP，行人密集区域则可能引入噪声直线，从而影响 VP 检测精度。为此，未来工作将结合手工设计的 VP 检测方法与深度学习，以更好地兼顾精度和推理速度。

算法 1 消失点检测

输入： 灰度图像 $x \in [0, 255]^{H \times W}$，最大中心到直线距离 $d_{\max}$，斜率保留区间 S，以及位于 x 内部、以 $c_i = (H_i, W_i), i = 1, ..., N_{\text{cell}}$ 为中心且边长为 $L = \lfloor H/4 \rfloor$ 的方形单元

$x \leftarrow$ 形态学开运算(x, 结构元 $= \mathbf{I}_{5\times5}$)
$x \leftarrow$ Canny 边缘检测($x, 50, 150$, 孔径大小 $= 3$)
lines $\leftarrow$ 霍夫直线($x, \rho = 1, \theta = \frac{\pi}{180}$, 阈值 $= 200$)
$c \leftarrow (W/2, H/2)$
对于 $\ell \in$ lines **执行**
    如果 $d(c, \ell) > d_{\max}$ 或 slope(ℓ) $\notin S$ **则**
        从 lines 中删除 ℓ
    结束条件
结束循环
$R =$ 求交点(lines)
对于 $i = 1, ..., N_{\text{cell}}$ **执行**
    $n_i \leftarrow$ 单元 i 内 R 的交点数量
结束循环
$i_{\text{opt}} = \arg\max_i n_i$
返回 $c_{i_{\text{opt}}}$

B. 补充消融实验

VP 邻近图嵌入的影响。 线性 VP 邻近图嵌入即以 VP 为中心的伪深度图：像素 (x, y) 的深度值设为 $1 - \Delta D$，其中 $\Delta D \propto \max\{\frac{|y-\hat{y}_j^p|}{H}, \frac{|x-\hat{x}_j^p|}{W}\}$，$(\hat{x}_j^p, \hat{y}_j^p)$ 为 VP 的像素坐标。在此基础上，进一步构造另外两种 VP 邻近图：幂次衰减和欧氏衰减（见图 6）。

- 线性衰减（图 6b）：$\Delta D \propto \max\{\frac{\Delta y}{H}, \frac{\Delta x}{W}\}$
  在线性衰减中，像素 (x, y) 的深度值随其到 VP 的距离线性减小；这种构造计算开销低，是本文默认方案。
- 幂次衰减（图 6c）：$\Delta D^2 \propto \max\{\frac{|\Delta y|}{H}, \frac{|\Delta x|}{W}\}$
  在幂次衰减中，深度值的平方与距离呈线性关系。该嵌入更为集中，但 VP 附近的深度值下降也更快。
- 欧氏衰减（图 6d）：$\Delta D \propto \sqrt{(\frac{|\Delta y|}{H})^2 + (\frac{|\Delta x|}{H})^2}$
  在欧氏衰减中，深度值随欧氏距离线性减小。该嵌入呈圆形且各向同性，但忽略了图像宽高比 $\frac{H}{W}$。

表 8 研究了上述 VP 邻近图嵌入的影响。值得注意的是，使用线性 VP 邻近图嵌入时，VPSeg 在 ACDC [42] 和 Cityscapes [9] 上均取得最高的 mIoU 和 mIA-IoU。采用幂次与欧氏嵌入的实验结果略低。可能原因是，欧氏衰减未考虑图像宽高比 $\frac{H}{W}$，而幂次衰减会使 VP 附近的深度值下降过快。

采样系数 Δd 的影响。 表 9 给出了不同 Δd 下的实验

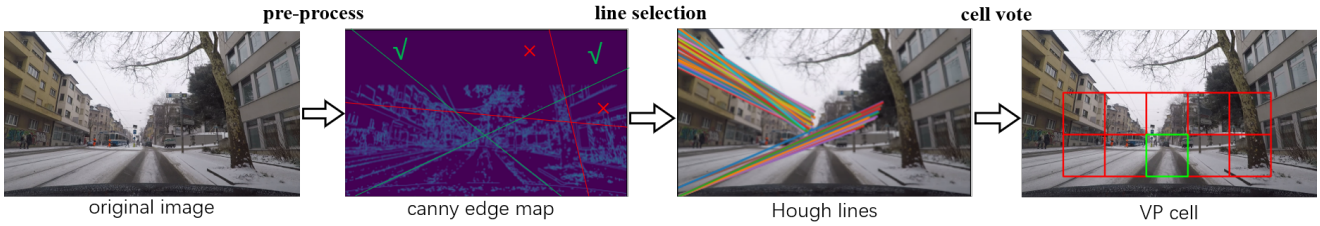
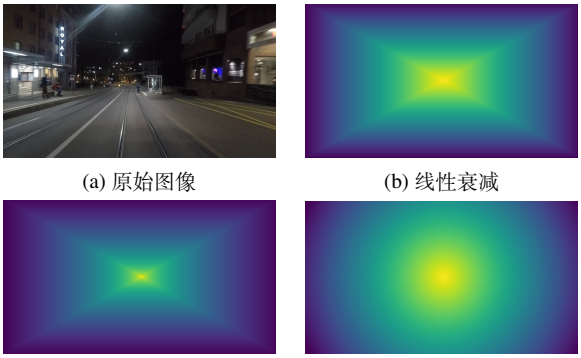


图 5. VP 检测流程。首先对输入帧进行形态学开运算 [39] 与 Canny 边缘检测 [2]，得到边缘图；随后执行霍夫变换 [33]，并剔除对 VP 检测无贡献的直线；最后通过单元投票统计各单元中的交点数量，确定最终 VP 位置。



(a) 原始图像　(b) 线性衰减

(c) 幂次衰减　(d) 欧氏衰减

图 6. 不同类型的 VP 邻近图嵌入。(a) 为输入帧；(b) 为线性衰减的 VP 邻近图。相比线性衰减，(c) 中的幂次衰减图在 VP 附近的深度值下降更快；(d) 为欧氏衰减邻近图，未考虑图像宽高比 $\frac{H}{W}$。

VP 邻近图嵌入	mIoU (A.)↑	mIA-IoU (A.)↑	mIoU (C.)↑
线性衰减	**77.48**	**41.48**	**82.46**
幂次衰减	77.29	41.23	82.29
欧氏衰减	77.33	41.16	82.35

表 8. 采用 MiT-B3 [54] 骨干网络时，不同 VP 邻近图嵌入在 ACDC（A.）与 Cityscapes（C.）上的消融实验。

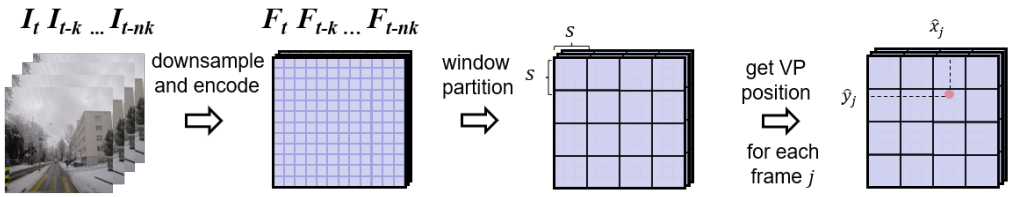
Δd	mIoU (A.)↑	mIA-IoU (A.)↑	mIoU (C.)↑
0	76.74	40.57	81.83
1	**77.48**	**41.48**	**82.46**
2	77.12	41.01	82.23
3	76.88	40.65	81.79

表 9. 采用 MiT-B3 [54] 骨干网络时，不同采样系数 Δd 在 ACDC（A.）与 Cityscapes（C.）上的消融实验。

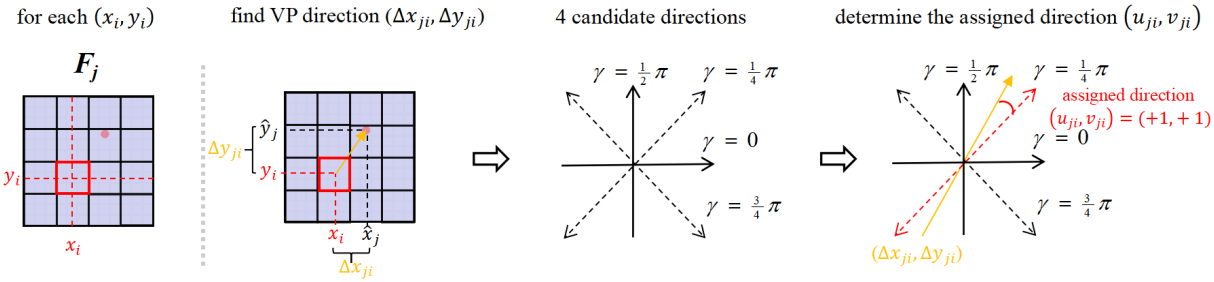
结果。结果表明 $\Delta d = 1$ 时模型能够充分覆盖快速运动目标，并取得良好性能。当 $\Delta d = 0$ 时，MotionVP 仅进行局部块采样，不适用于高速驾驶场景，因此 mIoU 与 mIA-IoU 均较低；当 $\Delta d > 1$ 时，VPSeg 性能急剧下降，表明继续增大 Δd 对 VP 引导的运动估计并无必要。

C. 详细流程

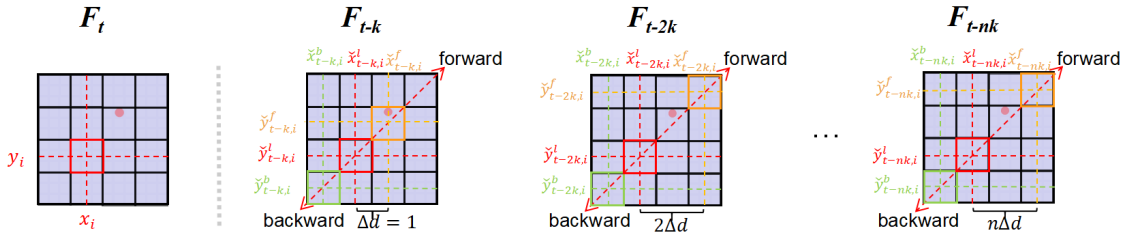
为利用动态和静态 VP 先验，本文提出了 MotionVP 与 DenseVP。MotionVP 用于提取动态上下文，可分为四个部分：特征块划分与 VP 检测、方向分配、块采样和特征聚合。DenseVP 通过在 VP 区域引入更细粒度的注意力来增强动态上下文，包含确定 VP 块索引、选择 VP 区域和生成稠密特征三个步骤。增强后的动态上下文送入预测头，用于最终预测。MotionVP 与 DenseVP 的详细流程分别见图 7和图 8；表 10 解释了 MotionVP 与 DenseVP 中符号的类型、取值域和含义。



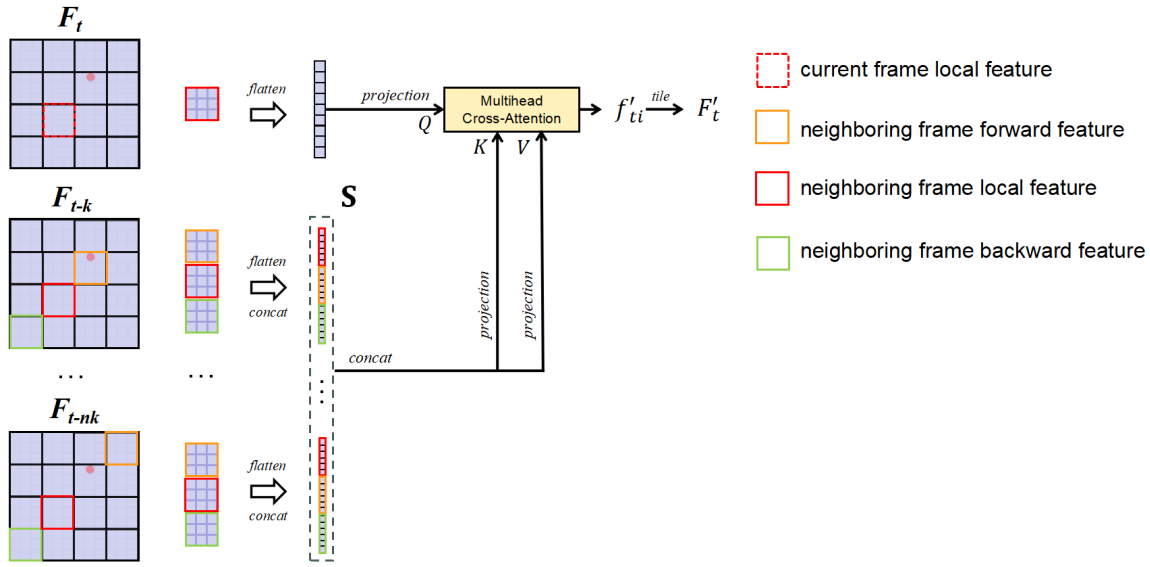
(a) **特征块划分与 VP 检测：**给定 $n+1$ 帧输入视频，先用预训练 Transformer 编码器提取特征图，再将特征图划分为大小为 $s \times s$ 的特征块，索引记为 i。



(b) **方向分配：**对每个块 (x_i, y_i) 确定其分配方向。令 $(\Delta x_{ji}, \Delta y_{ji})$ 表示从块中心指向 VP 的向量，分配方向 (u_{ji}, v_{ji}) 取与该向量最接近的候选方向。



(c) **块采样：**得到分配方向后，沿该方向前后采样相邻块。帧间隔越大，采样距离也随 Δd 增大；同时将采样位置限制在特征图边界以内，避免越界。



(d) **特征聚合：**通过交叉注意力 [49] 生成动态上下文 F'_t。对每个块 (x_i, y_i)，以当前帧块特征作为查询，以相邻帧采样特征作为键和值。交叉注意力得到块级动态特征 f'_t 后，再按原位置平铺重组为完整的帧级动态上下文 F'_t。

图 7. MotionVP 的详细流程。

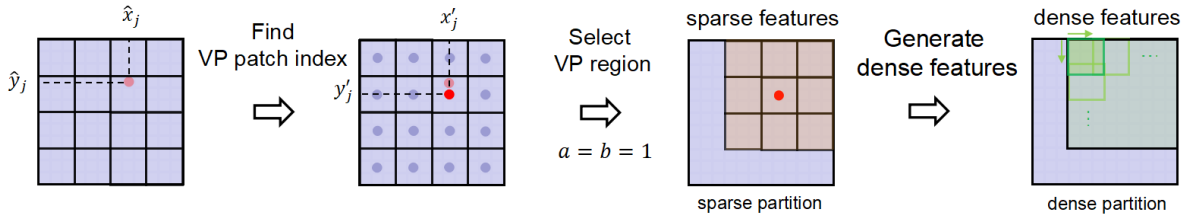


图 8. DenseVP 的详细流程。**确定 VP 块索引：**将距离 VP 最近的块作为 VP 块。**选择 VP 区域：**以 VP 块为中心选取矩形区域，记为 VP 区域 **A**。**生成稠密特征：**在 VP 区域内采用重叠式稠密划分，得到稠密特征 $f_{\mathbf{A}}$。

表 10. 符号表：列出各符号的类型、尺寸、取值域及含义。

符号	类型	尺寸（长度）	取值域	含义
$\mathcal{I}$	集合	$n+1$	-	输入帧集合
$\mathcal{T}$	集合	$n+1$	-	时间戳集合
$\mathcal{F}$	集合	$n+1$	-	特征图集合
$\mathcal{D}$	集合	$\frac{h}{s} \times \frac{w}{s}$	-	块索引集合
$\mathcal{A}$	集合	$(2a+1)(2b+1)$	-	VP 区域中的稀疏块索引集合
$\mathcal{V}$	集合	4	-	候选方向向量表示集合
$\mathcal{S}$	集合	$3n$	-	n 个相邻帧中的采样特征集合
c	标量	-	$\mathbb{N}$	特征通道数
h, w	标量	-	$\mathbb{N}$	特征图的空间高/宽
H, W	标量	-	$\mathbb{N}$	输入帧的空间高/宽
k	标量	-	$\mathbb{N}$	帧采样间隔
K	标量	-	$\mathbb{N}$	语义类别数
s	标量	-	$\mathbb{N}$	特征块大小
m	标量	-	$\mathbb{N}$	VP 区域中稠密块的数量
Δd	标量	-	$\mathbb{N}$	采样系数
$(\hat{x}_j, \hat{y}_j)$	坐标	2×1	$\mathbb{R}$	第 j 帧中的块级 VP 位置
$(\hat{x}_j^p, \hat{y}_j^p)$	坐标	2×1	$\mathbb{N}$	第 j 帧中的像素级 VP 位置
(x_i, y_i)	坐标	2×1	$\mathbb{N}$	第 i 个块的索引
$(\check{x}_{ji}^f, \check{y}_{ji}^f)$	坐标	2×1	$\mathbb{N}$	第 j 帧中第 i 个块的前向采样块索引
$(\check{x}_{ji}^b, \check{y}_{ji}^b)$	坐标	2×1	$\mathbb{N}$	第 j 帧中第 i 个块的后向采样块索引
$(\check{x}_{ji}^l, \check{y}_{ji}^l)$	坐标	2×1	$\mathbb{N}$	第 j 帧中第 i 个块的局部采样块索引
(x'_j, y'_j)	坐标	2×1	$\mathbb{N}$	第 j 帧中的 VP 块索引
I_t	矩阵	$H \times W$	$\mathbb{R}$	时刻 t 的输入帧
F_t	矩阵	$c \times h \times w$	$\mathbb{R}$	时刻 t 的特征图
f_{ti}	矩阵	$c \times s^2$	$\mathbb{R}$	时刻 t 中第 i 个块的块级特征
F_{tl}, F_{th}	矩阵	$c \times h \times w$	$\mathbb{R}$	I_t 的低/高分辨率特征图
$\check{f}_{ji}$	矩阵	$c \times 3s^2$	$\mathbb{R}$	第 j 帧中第 i 个块的采样特征
F'_t	矩阵	$c \times h \times w$	$\mathbb{R}$	时刻 t 的帧级动态特征
f'_{ti}	矩阵	$c \times s^2$	$\mathbb{R}$	时刻 t 中第 i 个块的块级动态特征
$f_{\mathcal{A}}$	矩阵	$c \times ms^2$	$\mathbb{R}$	VP 区域的稠密特征
F''_t	矩阵	$c \times h \times w$	$\mathbb{R}$	时刻 t 的增强动态上下文
E	矩阵	$h \times w$	$\mathbb{R}$	VP 邻近图
Q, Q_c	矩阵	$c \times K$	$\mathbb{R}$	CMA 中的可学习查询/上下文感知查询
F_m	矩阵	$c \times K$	$\mathbb{R}$	融合后的上下文
G_{nz}	矩阵	$H \times W$	$\{0,1\}$	第 n 张图像中第 z 类的真实标签
P_{nz}	矩阵	$H \times W$	$\{0,1\}$	第 n 张图像中第 z 类的预测
M_n	矩阵	$H \times W$	$\{0,1\}$	第 n 张图像的无效掩码