

Tracking Everything Everywhere All at Once



Figure 1: We present a new method for estimating full-length motion trajectories for every pixel in every frame of a video, as illustrated by the motion paths shown above. For clarity, we only show sparse trajectories for foreground objects, though our method computes motion for *all* pixels. Our method yields accurate, coherent long-range motion even for fast-moving objects, and robustly tracks through occlusions as shown in the *dog* and *swing* examples. For context, in the second row we depict the moving object at different moments in time.

1. Introduction

背景

运动估计 (motion estimation) 是计算机视觉中的一个核心任务。传统方法分为两大类：

1. 稀疏特征跟踪 (Sparse Feature Tracking) :

- 跟踪视频中少量显著的兴趣点 (interest points) 。
- 常用于例如运动恢复结构 (Structure from Motion, SfM) 等任务。
- 缺点：只能处理刚性场景，对动态场景和所有像素的运动无能为力。

2. 密集光流 (Dense Optical Flow) :

- 估计两帧图像中每个像素的运动。
- 优点：覆盖了整个图像的像素运动。

- 缺点：仅支持短时帧间运动，无法捕获长期轨迹，对遮挡问题不够鲁棒。
- 这些方法都有一个关键的局限：**无法在长时间内提供全局一致的运动估计。**

核心问题

视频运动估计面临以下三大挑战：

1. **长序列中的轨迹准确性**：现有方法会累积误差，无法保持运动轨迹的准确性。
2. **遮挡下的鲁棒跟踪**：当物体被遮挡后，许多方法无法在其再次出现时正确跟踪。
3. **时空一致性**：运动估计结果通常在时间和空间上缺乏全局一致性。

OmniMotion 的创新点

论文提出了一种名为 **OmniMotion** 的新方法，旨在解决上述挑战。其特点如下：

1. 使用了一个全局一致的运动表示方法，即“准3D (quasi-3D) ”表示。
 - 构造了一个全局的三维“标准体积” (canonical volume) 。
 - 将每一帧的局部坐标映射到标准体积，建立双向映射关系。
 - 这种方法保证了循环一致性 (cycle consistency)，可以跨遮挡进行跟踪。
2. 提出了一种全视频范围的联合优化方法。
 - 直接估计整个视频中每个像素的长时间运动轨迹。
 - 即使是快速运动的物体或复杂场景，该方法也能保持准确性。
3. 该方法同时适用于摄像机和场景运动。

方法总结

- **目标**：为视频中的每个像素提供完整的运动轨迹。

- **技术手段**: 通过一个准3D表示和测试时优化 (test-time optimization) 联合完成。
- **优势**:
 - 跟踪所有像素，即使在被遮挡情况下；
 - 适应任意复杂的摄像机和场景运动；
 - 提供全局一致性。

2. Related Work

2.1 稀疏特征跟踪 (Sparse Feature Tracking)

- **定义**: 通过在图像中检测显著的兴趣点 (如角点) 并跟踪它们的运动，建立帧间对应关系。
- **应用**:
 - 用于 **结构恢复 (Structure from Motion, SfM)** 和 **SLAM (Simultaneous Localization and Mapping)**。
- **局限**:
 - 仅限于一小部分显著点的跟踪。
 - 对于非刚性场景或动态物体的运动估计效果较差。

2.2 光流 (Optical Flow)

- **定义**: 估计图像中每个像素的运动，通过优化或深度学习方法预测帧间像素的对应关系。
- **研究进展**:
 - **优化方法**: 传统光流通过能量最小化计算光流，例如 Horn-Schunck 方法。
 - **深度学习**: 近年来，基于深度学习的光流估计方法 (如 RAFT) 显著提高了精度和效率。

- 局限：
 - 仅适用于短时间内的帧间运动，长时间轨迹容易累积误差。
 - 对于遮挡、快速运动或复杂的长距离运动估计存在困难。

2.3 特征匹配 (Feature Matching)

- 定义：利用局部特征描述子在两帧之间找到匹配点，以建立帧间对应关系。
- 进展：
 - 采用弱监督或自监督方法学习特征匹配，例如基于循环一致性（cycle consistency）。
 - 使用几何约束或 3D 重建提供的特征点匹配。
- 局限：
 - 缺乏时间上的上下文信息，容易导致长时间跟踪不一致。
 - 遮挡处理能力较弱。

2.4 像素级长时间跟踪 (Pixel-Level Long-Range Tracking)

- 最新方法：
 - PIPs：通过局部窗口的上下文信息估计多帧轨迹，但时间窗口较短（8 帧），超出窗口范围容易漂移。
 - TAPIR：结合匹配阶段和逐帧优化进行点跟踪。
 - CoTracker：基于 Transformer 的灵活跟踪算法，能够捕获全视频范围内的运动轨迹。
- 本论文的贡献：
 - 提出的方法可补充这些方法，利用它们的输出作为优化全局运动表示的初始监督信号。

2.5 视频运动优化 (Video-Based Motion Optimization)

- 定义：在整个视频范围内全局优化运动，生成一致的轨迹。
- 相关工作：
 - Particle Video：基于初始光流场生成半稠密的长时间轨迹，但遮挡会中断轨迹。
 - ParticleSfM：将长距离对应关系用于相机姿态估计，但动态物体会被当作异常点处理。
- 本论文的改进：
 - 提出的方法可以在遮挡情况下保持轨迹连续性，并对动态物体进行准确建模。

2.6 神经视频表示 (Neural Video Representations)

- 定义：利用坐标感知的多层感知机 (MLP) 建模视频场景。
- 相关工作：
 - 层级神经地图 (Layered Neural Atlases)：将视频分解为全局纹理图，但对复杂运动建模能力有限。
 - NeRF：使用神经辐射场表示动态场景，但通常需要已知的相机姿态。
- 本论文的创新：
 - 提出的准3D表示通过双射映射建模视频，无需显式分解场景几何和运动。

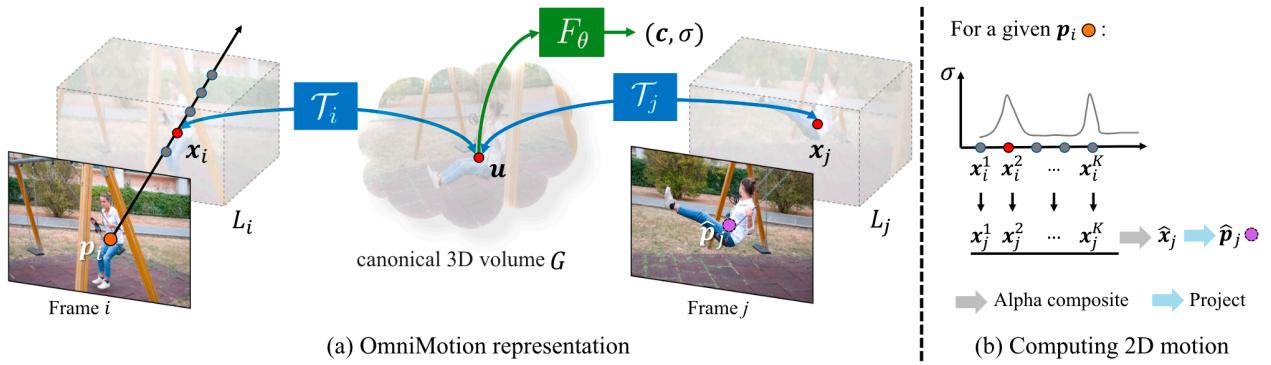


Figure 2: *Method overview.* (a) Our OmniMotion representation is comprised of a canonical 3D volume G and a set of bijections \mathcal{T}_i that map between each frame's local volume L_i and the canonical volume G . Any local 3D location x_i in frame i can be mapped to its corresponding canonical location u through \mathcal{T}_i , and then mapped back to another frame j as x_j through the inverse mapping \mathcal{T}_j^{-1} . Each location u in G is associated with a color c and density σ , computed using a coordinate-based MLP F_θ . (b) To compute the corresponding 2D location for a given query point p_i mapped from frame i to j , we shoot a ray into L_i and sample a set of points $\{x_i^k\}_{k=1}^K$, which are then mapped first to the canonical space to obtain their densities, and then to frame j to compute their corresponding local 3D locations $\{x_j^k\}_{k=1}^K$. These points $\{x_j^k\}_{k=1}^K$ are then alpha-composited and projected to obtain the 2D corresponding location \hat{p}_j .

3. Overview

本文提出了一种 **测试时优化 (Test-Time Optimization)** 方法，针对视频序列进行稠密且长时间的运动估计。具体步骤如下：

输入与目标

- 输入：
 - 一组视频帧。
 - 帧间的噪声运动估计（例如，光流场）。
- 目标：
 - 构建一个全局一致的运动表示。
 - 实现视频中任意像素的长时间轨迹估计，且能够跨遮挡进行跟踪。

方法概述

1. 全局运动表示：

- 提出了一种新的运动表示方法，称为 **OmniMotion**。

- 包括：
 - 一个 **准三维标准体积** (**Quasi-3D Canonical Volume**)，用于表示全局场景信息。
 - **局部到标准空间的双射映射** (**Local-Canonical bijections**)，用于将每帧的局部坐标映射到标准体积。

2. 核心特性：

- **循环一致性** (**Cycle Consistency**)：通过双射映射，确保帧间映射的全局一致性。
- **遮挡处理** (**Occlusion Handling**)：即使在像素被遮挡的情况下，也能保持轨迹的连续性。
- **摄像机与场景运动建模** (**Camera and Scene Motion Modeling**)：能够同时捕获摄像机和场景的复杂运动。

3. 查询过程：

- 优化完成后，该方法可在视频任意时间点上查询像素轨迹，返回整个视频范围内的平滑、准确运动。

后续章节

在后续章节中，论文详细介绍了：

- **OmniMotion 表示的具体设计** (第 4 节)。
- **优化过程的实现方法** (第 5 节)。

4. OmniMotion representation

4.1 Canonical 3D Volume (标准三维体积)

我们将视频内容表示为一个 **标准三维体积** G ，该体积充当所观察场景的三维地图。

关键要点：

1. 三维坐标映射：

- 使用一个基于坐标的网络 F_θ , 它将标准三维体积中的每个三维坐标 $\mathbf{u} \in G$ 映射为:

- **密度 σ** : 表示场景表面在标准空间中的分布。

- **颜色 \mathbf{c}** : 用于优化时的光度一致性。

- 公式:

$$F_\theta(\mathbf{u}) = (\sigma, \mathbf{c}) \quad (1)$$

2. 密度的作用：

- 用于确定场景表面的存在位置。

- 与 3D 双射映射结合, 可实现对多帧的表面跟踪以及遮挡关系的推断。

3. 颜色的作用：

- 在优化过程中, 允许通过光度损失 (photometric loss) 来提高精度。

4.2 3D Bijections (3D 双射映射)

定义了一组连续的双射映射 T_i , 用于将每帧的局部三维坐标 \mathbf{x}_i 映射到标准三维体积 G 的三维坐标 \mathbf{u} 。

双射映射公式：

- 局部坐标到标准空间:

$$\mathbf{u} = T_i(\mathbf{x}_i) \quad (2)$$

- 标准空间到局部坐标:

$$\mathbf{x}_i = T_i^{-1}(\mathbf{u}) \quad (3)$$

特性：

1. 循环一致性 (Cycle Consistency) :

- 任意帧之间的映射可以通过标准体积完成，并保证一致性：

$$\mathbf{x}_j = T_j^{-1} \circ T_i(\mathbf{x}_i) \quad (4)$$

2. 神经网络实现：

- 使用可逆神经网络 (Invertible Neural Networks, INNs) 实现双射映射。
- 基于 Real-NVP 结构，通过仿射耦合层 (Affine Coupling Layers) 实现可逆性。

3. 时间依赖性：

- 映射函数 T_i 的参数依赖于每帧的潜在代码 ψ_i ：

$$T_i(\cdot) = M_\theta(\cdot; \psi_i) \quad (5)$$

4.3 Computing Frame-to-Frame Motion (帧间运动计算)

给定视频帧 i 中的查询像素 \mathbf{p}_i ，计算其在目标帧 j 中的对应像素位置 $\hat{\mathbf{p}}_j$ 。

计算流程：

1. 升维至三维：

- 将像素 \mathbf{p}_i 的二维坐标扩展为三维光线上的点：

$$\mathbf{r}_i(z) = \mathbf{o}_i + z\mathbf{d} \quad (6)$$

其中：

- $\mathbf{o}_i = [\mathbf{p}_i, 0]$ 是光线的起点；
- $\mathbf{d} = [0, 0, 1]$ 是光线方向；

- z 是深度。

2. 采样点：

- 在光线 $\mathbf{r}_i(z)$ 上采样 K 个点 $\{\mathbf{x}_i^k\}_{k=1}^K$ 。

3. 密度和颜色查询：

- 将采样点映射到标准体积 G ，并通过密度网络 F_θ 查询密度 σ 和颜色 \mathbf{c} ：

$$(\sigma_k, \mathbf{c}_k) = F_\theta(T_i(\mathbf{x}_i^k)) \quad (7)$$

4. 映射至目标帧：

- 使用双射映射将采样点从标准空间转换到目标帧 j 的局部坐标：

$$\mathbf{x}_j^k = T_j^{-1}(\mathbf{u}) \quad (8)$$

5. 组合点云 (Compositing) :

- 使用 Alpha 组合对采样点进行聚合，生成目标帧中的对应位置：

$$\hat{\mathbf{x}}_j = \sum_{k=1}^K T_k \alpha_k \mathbf{x}_j^k, \quad T_k = \prod_{l=1}^{k-1} (1 - \alpha_l) \quad (9)$$

其中 $\alpha_k = 1 - \exp(-\sigma_k)$ 是 Alpha 权重。

6. 降维至二维：

- 将三维点 $\hat{\mathbf{x}}_j$ 投影到目标帧 j 的二维平面，得到对应像素位置：

$$\hat{\mathbf{p}}_j \quad (10)$$

5. Optimization

5.1 Collecting Input Motion Data (收集输入运动数据)

1. 使用输入光流数据：

- 本文的优化方法依赖于现有方法生成的初始输入运动数据（如 RAFT 或 TAP-Net）。
- 输入为每对视频帧之间的光流，包含帧间像素的对应关系。

2. 预处理步骤：

- 光流计算：

$$f_{i \rightarrow j} \quad (11)$$

表示从帧 i 到帧 j 的光流。

- 循环一致性检查（Cycle Consistency Check）：
 - $f_{i \rightarrow j}$ 和其反向光流 $f_{j \rightarrow i}$ 之间必须满足前后匹配。
 - 如果差异大于一定阈值（例如 3 像素），则丢弃该对应关系。
- 外观一致性检查（Appearance Consistency Check）：
 - 使用深度特征（如 DINO）计算相似性，过滤掉低相似性匹配。

3. 对遮挡区域的处理：

- 通过多次循环一致性检查，检测并保留遮挡区域的可靠流动。

5.2 Loss Functions (损失函数)

本文优化使用了三个主要损失函数，用于生成全局一致的运动表示。

1. 光流损失（Flow Loss）：

- 定义：最小化优化表示生成的预测光流 $\hat{f}_{i \rightarrow j}$ 和输入光流 $f_{i \rightarrow j}$ 之间的误差。
- 公式：

$$L_{\text{flo}} = \sum_{f_{i \rightarrow j} \in \Omega_f} \|\hat{f}_{i \rightarrow j} - f_{i \rightarrow j}\|_1 \quad (12)$$

其中， Ω_f 是所有过滤后的光流对。

2. 光度损失 (Photometric Loss) :

- 定义：约束优化过程中预测颜色 \hat{C}_i 与输入视频帧颜色 C_i 的一致性。
- 公式：

$$L_{\text{pho}} = \sum_{(i,p) \in \Omega_p} \|\hat{C}_i(p) - C_i(p)\|_2^2 \quad (13)$$

其中， Ω_p 是视频中所有像素点的集合。

3. 时间平滑正则化 (Temporal Smoothness Regularization) :

- 定义：约束三维运动的时间连续性，最小化加速度。
- 公式：

$$L_{\text{reg}} = \sum_{(i,x) \in \Omega_x} \|x_{i+1} + x_{i-1} - 2x_i\|_1 \quad (14)$$

其中， Ω_x 是视频中所有帧的局部三维空间。

总损失函数：

- 损失函数的加权组合：

$$L = L_{\text{flo}} + \lambda_{\text{pho}} L_{\text{pho}} + \lambda_{\text{reg}} L_{\text{reg}} \quad (15)$$

其中， λ 表示各损失项的权重。

5.3 Balancing Supervision via Hard Mining (通过难例挖掘平衡监督信号)

1. 问题：

- 动态区域中的监督信号往往较少，导致模型更多地关注背景的简单运动，而忽略复杂运动。

2. 解决方法：

- 难例挖掘策略：
 - 定期缓存预测光流，生成误差图。
 - 根据预测光流和输入光流之间的欧几里得距离计算误差。
 - 在动态区域（高误差区域）中更频繁地采样监督信号。

3. 实现细节：

- 错误图基于连续帧生成，假设相邻帧的光流预测更可靠。
- 在优化过程中，动态调整采样权重，增强模型对快速运动或遮挡区域的关注。

5.4 Implementation Details (实现细节)

1. 网络结构

- 映射网络 (**Mapping Network**)：
 - M_θ 包含 6 层仿射耦合层 (Affine Coupling Layers)。
 - 每层的输入坐标会通过位置编码 (Positional Encoding) 处理，包含 4 个频率。
- 潜在编码网络 (**Latent Code Network**)：
 - 使用一个两层的多层感知机 (MLP)，每层 256 个通道，作为 GaborNet。
 - 输入为时间 t_i ，输出为每帧的潜在代码 ψ_i ，其维度为 128。
- 标准表示网络 (**Canonical Representation Network**)：
 - F_θ 是一个三层的 GaborNet，每层 512 个通道，用于生成密度和颜色。

2. 表示

- **像素坐标归一化：**
 - 将像素坐标 \mathbf{p}_i 归一化到 $[-1, 1]$ 范围。
 - 深度范围设为 $[0, 2]$ ，每帧的局部三维空间为 $[-1, 1]^2 \times [0, 2]$ 。
 - **标准三维空间初始化：**
 - 为了确保优化稳定，将初始标准空间 G 的映射位置约束在单位球体内。
 - **数值稳定性增强：**
 - 使用 mip-NeRF 360 的压缩操作对标准三维坐标 \mathbf{u} 进行处理，以提高训练的数值稳定性。
-

3. 训练过程

- **优化算法：**
 - 使用 Adam 优化器进行 200,000 次迭代。
- **训练样本：**
 - 每批次随机采样 256 个对应点，来自 8 对图像，总计 1024 个对应点。
 - 每条光线均匀采样 $K = 32$ 个点，使用分层采样（Stratified Sampling）。
- **损失函数：**
 - 总损失函数包括光流损失、光度损失、时间平滑正则化：

$$L = L_{\text{flo}} + \lambda_{\text{pho}} L_{\text{pho}} + \lambda_{\text{reg}} L_{\text{reg}} \quad (16)$$

更多训练细节可以参考附录中的补充材料。

Method	Kinetics				DAVIS				RGB-Stacking			
	AJ \uparrow	$< \delta_{\text{avg}}^x \uparrow$	OA \uparrow	TC \downarrow	AJ \uparrow	$< \delta_{\text{avg}}^x \uparrow$	OA \uparrow	TC \downarrow	AJ \uparrow	$< \delta_{\text{avg}}^x \uparrow$	OA \uparrow	TC \downarrow
RAFT-C [66]	31.7	51.7	84.3	0.82	30.7	46.6	80.2	0.93	42.0	56.4	91.5	0.18
RAFT-D [66]	50.6	66.9	85.5	3.00	34.1	48.9	76.1	9.83	72.1	85.1	92.1	1.04
TAP-Net [15]	48.5	61.7	86.6	6.65	38.4	53.4	81.4	10.82	61.3	73.7	91.5	1.52
PIPs [23]	39.1	55.3	82.9	1.30	39.9	56.0	81.3	1.78	37.3	50.6	89.7	0.84
Flow-Walk-C [5]	40.9	55.5	84.5	0.77	35.2	51.4	80.6	0.90	41.3	55.7	92.2	0.13
Flow-Walk-D [5]	46.9	65.9	81.8	3.04	24.4	40.9	76.5	10.41	66.3	82.7	91.2	0.47
Deformable-Sprites [81]	25.6	39.5	71.4	1.70	20.6	32.9	69.7	2.07	45.0	58.3	84.0	0.99
Ours (TAP-Net)	53.8	68.3	88.8	0.77	50.9	66.7	85.7	0.86	73.4	84.1	92.2	0.11
Ours (RAFT)	55.1	69.6	89.6	0.76	51.7	67.5	85.3	0.74	77.5	87.0	93.5	0.13

Table 1: Quantitative comparison of our method and baselines on the TAP-Vid benchmark [15]. We refer to our method as *Ours*, and present two variants, *Ours (TAP-Net)* and *Ours (RAFT)*, which are optimized using input pairwise correspondences from TAP-Net [15] and RAFT [66], respectively. Both *Ours* and *Deformable Sprites* [81] estimate global motion via test-time optimization on each individual video, while all other methods estimate motion locally in a feed-forward manner. Our method notably improves the quality of the input correspondences, achieving the best position accuracy, occlusion accuracy, and temporal coherence among all methods tested.

Method	AJ \uparrow	$< \delta_{\text{avg}}^x \uparrow$	OA \uparrow	TC \downarrow
No invertible	12.5	21.4	76.5	0.97
No photometric	42.3	58.3	84.1	0.83
Uniform sampling	47.8	61.8	83.6	0.88
Full	51.7	67.5	85.3	0.74

Table 2: Ablation study on DAVIS [50].

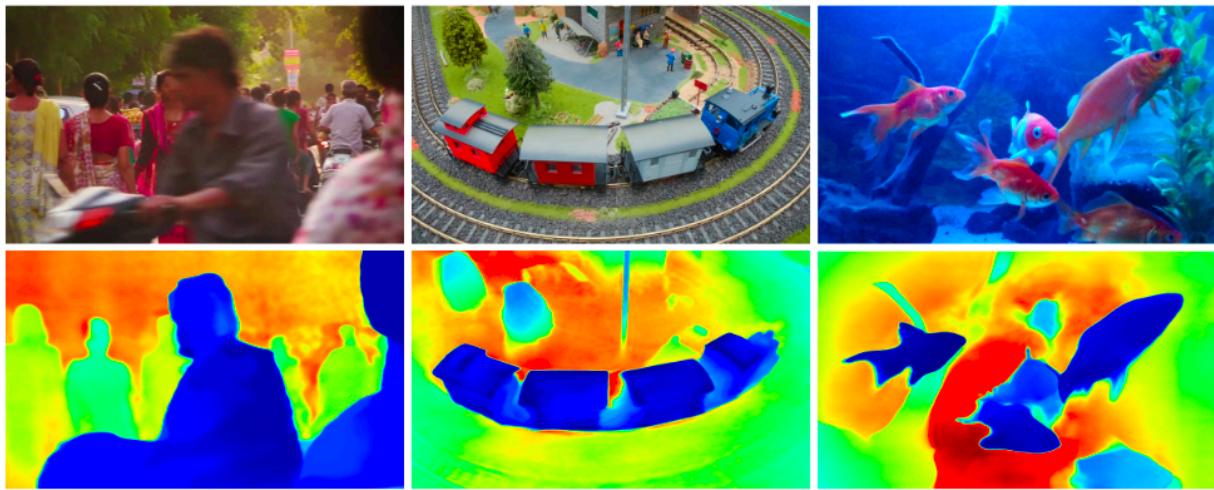


Figure 4: Pseudo-depth maps extracted from our representation, where blue indicates closer objects and red indicates further.

6. Evaluation (评估)

6.1 Benchmarks (基准测试)

数据集

1. DAVIS:

- 来自 DAVIS 2017 验证集的 30 个视频。
- 每个视频长度为 34-104 帧，平均每个视频包含 21.7 个点轨迹标注。

2. Kinetics:

- 来自 Kinetics-700-2020 验证集的 1,189 个视频。
- 每个视频 250 帧，平均每个视频包含 26.3 个点轨迹标注。
- 为了测试优化方法的可行性，随机采样 100 个视频进行测试。

3. RGB-Stacking:

- 一个合成数据集，包含 50 个视频，每个视频 250 帧，30 条轨迹标注。

评价分辨率

- 所有方法在 256×256 分辨率下进行定量评估。
 - 定性结果展示以更高分辨率（480p）运行。
-

6.2 Evaluation Metrics (评估指标)

论文中使用以下评估指标来衡量模型性能：

1. 位置准确性 (Position Accuracy) :

- 测量可见点的平均位置准确性。
- 定义为 δx_{avg} , 计算在 5 个阈值（1、2、4、8 和 16 像素）上的准确率平均值。

2. 平均 Jaccard 指数 (Average Jaccard, AJ) :

- 综合位置准确性和遮挡预测准确性的指标。
- 定义为：

$$AJ = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives} + \text{False Negatives}} \quad (1)$$

- 其中：

- 真正例：在可见点中，预测位置与阈值内的点。
- 假正例：预测为可见但实际遮挡的点。
- 假负例：预测为遮挡但实际可见的点。

3. 遮挡准确性 (Occlusion Accuracy, OA) :

- 测量对点遮挡/可见性的准确预测率。

4. 时间一致性 (Temporal Coherence, TC) :

- 计算预测轨迹与真实轨迹加速度之间的 L_2 距离。
- 加速度定义为相邻帧之间的光流差值。

6.3 Comparisons (比较分析)

方法对比

1. RAFT:

- RAFT-C：通过连续帧的光流预测生成长时间轨迹。
- RAFT-D：直接计算任意帧对之间的光流。

2. PIPs:

- 估计多帧点轨迹，但时间窗口有限（8 帧）。

3. Flow-Walk:

- 学习时空对应关系，支持直接和链式推断。

4. TAP-Net:

- 使用代价体积预测单个目标帧的点位置和遮挡概率。

5. Deformable Sprites:

- 基于层的表示方法，限制于预定义的两层或三层结构。

定量结果

- 在 TAP-Vid 基准测试中，与基线方法的对比结果展示了该方法在位置准确性、遮挡准确性和时间一致性上的显著改进。

定性结果

- 本文方法在复杂场景（如遮挡、快速运动）中表现出色，能够生成平滑且一致的轨迹。

6.4 Ablations and Analysis (消融实验与分析)

消融实验

1. 无可逆映射 (No Invertible) :

- 用单向映射代替双射映射，结果表明循环一致性是生成有效全局表示的关键。

2. 无光度损失 (No Photometric) :

- 去掉光度损失项 L_{pho} ，性能显著下降，表明光度一致性对优化的贡献。

3. 均匀采样 (Uniform Sampling) :

- 替换难例挖掘策略为均匀采样，无法捕获快速运动区域。

分析

- Pseudo-depth maps (伪深度图) 展示了模型学习到的深度排序能力，即使在无显式几何监督的情况下，也能推断相对深度。

7. Limitations (局限性)

尽管本文提出的方法在许多复杂场景下表现优异，但仍存在以下局限性：

1. 非刚性和快速运动

- 对于快速且高度非刚性的运动（如快速变形的物体或薄结构），模型可能难以提供准确的全局运动表示。
- 原因：这些情况下，现有的配对方法（如光流）无法生成足够可靠的对应关系。

2. 初始敏感性

- 由于底层优化问题的高度非凸性，对于某些复杂视频，模型对初始化的敏感性可能导致次优的局部极小值。
- 示例问题：
 - 错误的表面排序。

- 在标准空间中出现重复物体。
-

3. 计算开销

- 光流预处理：

- 需要对所有帧对计算光流，其时间复杂度与序列长度呈二次方增长。
- 改进方向：可探索如关键帧匹配或词汇树的方法，借鉴结构恢复（Structure from Motion, SfM）和 SLAM 的文献。

- 优化过程：

- 类似于其他基于神经隐式表示的方法（如 NeRF），优化过程相对较慢。
 - 潜在改进：最近的研究（如快速优化方法）可能有助于加速这一过程，并扩展到更长的序列。
-

总结

尽管存在上述局限性，本文方法在稠密、长时间运动估计中取得了显著进展，特别是在遮挡处理和时空一致性方面，为视频运动估计领域提供了一个强有力的新工具。