Particle Video Revisited: Tracking Through Occlusions Using Point Trajectories

Fig. 1. Persistent Independent Particles. Our method takes an RGB video as input, and estimates trajectories for any number of target pixels. Left: targets and their trajectories, shown separately. Right: trajectories overlaid on the pixels.

Introduction

2006年，Sand 和 Teller 提出了“粒子视频（Particle Video）”的概念，作为特征跟踪和光流之间的一种新型运动表示方法。他们的核心思想是跟踪视频中跨越多个帧的粒子，同时利用长时程的时间先验信息。

目前视频中运动估计的方法主要分为以下两类：

1. **特征匹配（Feature Matching）**：通过匹配帧间的特征点来完成，但通常无法保证时间上的连续性。
2. **光流（Optical Flow）**：估计每个像素的运动，但在遮挡或物体出框时容易丢失跟踪。

当前方法的局限性

- **特征匹配的缺点**：
  - 无法利用时间上下文信息。
  - 在复杂运动序列中容易产生误差。
- **光流的缺点**：
  - 对短时间的运动估计效果较好。
  - 遇到遮挡或物体出界时会失效。

Sand 和 Teller 提出的粒子视频旨在解决这些问题，主要具有以下两个特点：

1. **遮挡期间的持续性（Persistence Through Occlusions）**：即使物体在某些帧中不可见，仍然可以推断出其轨迹。
2. **多帧时间上下文的利用（Multi-frame Temporal Context）**：跟踪粒子时考虑多个帧的时间信息，而不仅限于连续的两帧。

本文贡献

本文提出了一种新的粒子视频方法，称为 **Persistent Independent Particles (PIPs)**，具有以下特点：

- 输入：一个 T 帧的 RGB 视频，以及第一帧中需要跟踪的目标像素坐标 (x, y)。
- 输出：目标像素在所有帧中的位置轨迹矩阵 $T \times 2$。

我们的方法通过如下方式创新：

- 在空间感知和时间感知之间进行极端取舍。
- 每个粒子的轨迹独立估计，不受其他粒子影响。
- 学习了强大的时间先验模型，并引入迭代推理机制。

通过这样的设计，即使在某些帧中目标不可见，只要在时间范围内的某些关键帧中重新找到目标，模型就可以利用时间先验推测出其余帧的合理位置。

方法效果

- 使用合成数据集 FlyingThings++ 进行训练，该数据集包含了具有挑战性的遮挡和多帧轨迹。
- 实验结果表明，PIPs 在遮挡情况下比光流方法和特征匹配方法具有更高的鲁棒性。

Realeted Work

在视频的运动估计领域，相关研究主要集中在以下三个方向：

1. 光流（Optical Flow）

光流是研究视频中像素运动的传统方法，其目标是估计连续两帧图像之间的像素位移。经典光流方法的进展主要体现在以下几个方面：

1. **优化方法**：早期的方法通过优化技术来估计连续两帧之间的运动场，例如经典的Lucas-Kanade方法。
2. **深度学习方法**：近年来，深度学习技术被广泛用于光流估计。例如：
    - FlowNet [6] 使用卷积神经网络从合成数据集中学习像素位移。
    - RAFT [32] 模仿优化算法，使用 4D 相关体构建逐帧像素间的对应关系，并通过迭代更新实现高精度光流估计。

尽管这些方法在单帧配对中表现出色，但在处理长时程运动和遮挡时仍存在局限性。例如：

- **遮挡问题**：目标一旦被遮挡，光流场将丢失目标的表示，从而导致跟踪失败。

- **时间上下文利用不足**：大多数方法仅利用简单的常速假设（如恒速先验），无法全面建模长时程运动的复杂性。

2. 特征匹配（Feature Matching）

特征匹配方法的核心在于对视频中的特征点进行时间上的一致性匹配。近年来的研究包括：

1. **循环一致性学习（Cycle Consistency Learning）**：如 Wang 和 Jabri 等人提出的方法[39][11]，通过优化时间上的循环一致性损失，来学习未标注视频中的特征。
2. **监督方法**：通过真实的特征对应关系对模型进行监督训练。例如：
    - COTR [13] 使用 transformer 架构来定位跨帧图像的特征对应点。

然而，这些方法通常仅关注帧对之间的关系，无法有效捕捉长时程的运动上下文。

3. 带时间先验的跟踪（Tracking with Temporal Priors）

对象跟踪研究领域长期以来一直关注如何处理遮挡和外观变化的问题。例如：

1. **对象跟踪方法**：在非线性时间先验和遮挡场景下表现较好的对象跟踪技术[26]。
2. **时间上下文利用**：已有研究探索了多帧光流估计和遮挡问题，例如[12] 提出的多帧光流方法显式推理了遮挡，但仅使用简单的恒速假设。
3. **粒子视频方法**：Sand 和 Teller 提出的粒子视频在长时间跟踪中首次引入了点级别的时间先验。

本文的独特之处

相比上述方法，本文方法（PIPs）具有以下创新：

1. 明确使用广泛的时间上下文来跟踪目标点。
2. 利用深度网络学习精确的时间先验，同时通过迭代推理机制搜索遮挡后的目标位置。
3. 旨在恢复跨遮挡的 **Amodal Trajectories**（完整轨迹），即使目标暂时被遮挡也能重连轨迹。

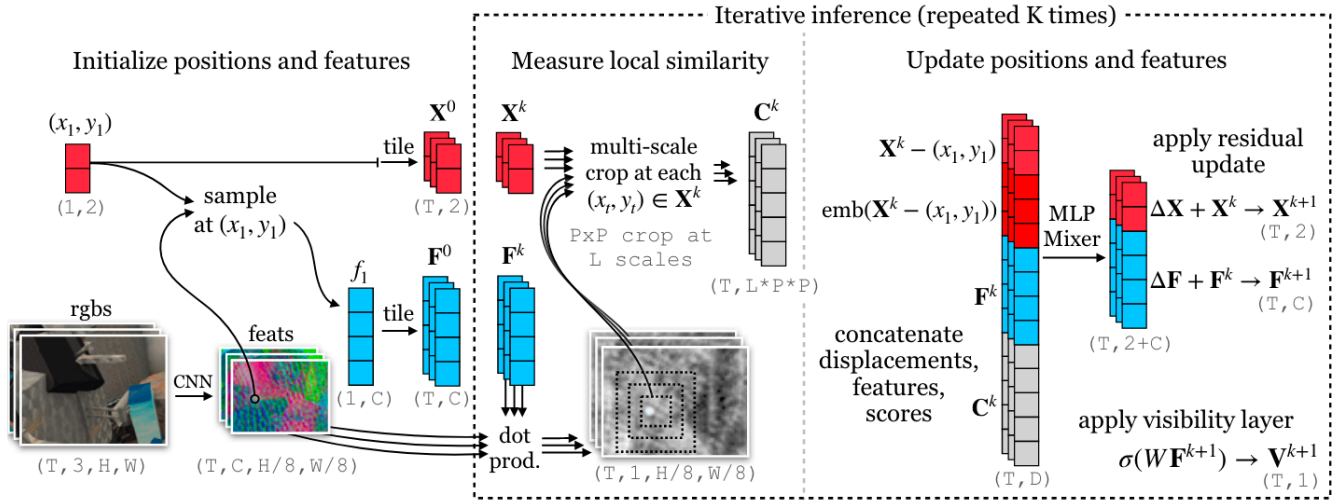

Fig. 2. Persistent Independent Particles (PIPs) architecture. Given an RGB video as input, along with a location in the first frame indicating what to track, our model initializes a multi-frame trajectory, then computes features and correlation maps, and iteratively updates the trajectory and its corresponding sequence of features, with a deep MLP-Mixer model. From the computed features, the model also estimates a visibility score for each timestep of the trajectory.

Method

本文提出了 **Persistent Independent Particles (PIPs)**，一种新的粒子视频方法，用于多帧点轨迹估计，特别是在遮挡和出界情况下。以下是方法的详细分解。

1. 问题定义与输入输出

PIPs 的目标是跟踪视频中任意给定的目标像素。具体描述如下：

- **输入**：
  - 一个包含 T 帧的 RGB 视频。
  - 第 1 帧中需要跟踪的目标像素坐标 (x_1, y_1)。

- **输出**：
  - 一个大小为 $T \times 2$ 的位置轨迹矩阵 $\mathbf{X} = \{(x_t, y_t)\}_{t=1}^{T}$，表示目标像素在每帧中的位置。
  - 可见性估计 $\mathbf{V} = \{v_t\}_{t=1}^{T}$，其中 $v_t \in [0, 1]$ 表示目标在第 t 帧中的可见性。

模型特点：

- 可以同时查询 N 个目标点，并共享部分计算。
- 每个目标点的轨迹独立估计，不受其他目标点的影响。

---

2. 方法概览

PIPs 的方法流程可以分为以下四个阶段：

1. **特征提取（Feature Extraction）**：从输入视频的每一帧中提取视觉特征。
2. **目标初始化（Target Initialization）**：初始化目标的轨迹和特征。
3. **局部外观相似性计算（Local Appearance Similarity）**：计算目标的特征与视频帧中局部区域的相似性。
4. **迭代更新（Iterative Updates）**：通过深度模型逐步优化轨迹和特征。

---

3. 特征提取

在每一帧中，使用 2D 卷积神经网络（CNN）独立提取特征：

- 输入：视频帧大小为 (H, W)。
- 输出：每帧生成的特征图大小为 $(\frac{H}{8}, \frac{W}{8}, C)$，其中 C 为特征通道数。

公式表示：

$$\mathbf{F}_{\text{frame}} = \text{CNN}(\mathbf{I}_{\text{frame}}) \tag{1}$$

其中：

- $\mathbf{I}_{\text{frame}}$ 表示单帧输入图像。
- $\mathbf{F}_{\text{frame}}$ 是特征图。

---

4. 目标初始化

4.1 初始化特征轨迹

从第一帧的特征图中，使用双线性插值采样得到目标初始特征：

$$\mathbf{f}_1 = \text{Sample}(\mathbf{F}_1, (x_1, y_1)) \tag{2}$$

将目标初始特征复制到所有时间步，形成特征轨迹：

$$\mathbf{F}_0 = \{\mathbf{f}_1, \mathbf{f}_1, \ldots, \mathbf{f}_1\}_{t=1}^{T} \tag{3}$$

4.2 初始化位置轨迹

目标位置轨迹被简单地初始化为静止假设：

$$\mathbf{X}_0 = \{(x_1, y_1), (x_1, y_1), \ldots, (x_1, y_1)\}_{t=1}^{T} \tag{4}$$

以上初始化提供了以下两个先验：

1. **外观恒定性（Appearance Constancy）**：初始假设目标特征在时间上保持恒定。
2. **零速度假设（Zero Velocity Prior）**：初始假设目标静止不动。

---

5. 局部外观相似性计算

通过计算目标特征与每一帧特征图的局部相关性来衡量匹配度：

1. **全局相关性计算**：

$$\mathbf{C}_t = \mathbf{F}_t \cdot \mathbf{f}_t^T \tag{5}$$

   其中：

   - $\mathbf{F}_t$ 是第 t 帧的特征图。
   - $\mathbf{f}_t$ 是目标的当前特征。

2. **局部采样**：
   从相关性图 $\mathbf{C}_t$ 中，围绕当前目标位置 (x_t, y_t) 提取局部补丁：

$$\mathbf{C}_{t,\text{local}} = \text{Crop}(\mathbf{C}_t, (x_t, y_t), P \times P) \tag{6}$$

   其中 $P \times P$ 是补丁大小。

3. **多尺度相关性金字塔**：
   为了捕捉不同尺度的相似性，对相关性补丁构建多尺度金字塔：

$$\mathbf{C}_{t,\text{pyramid}} = \{\mathbf{C}^1_{t,\text{local}}, \mathbf{C}^2_{t,\text{local}}, \ldots, \mathbf{C}^L_{t,\text{local}}\} \tag{7}$$

   其中 L 表示尺度的数量。

---

6. 迭代更新

PIPs 通过一个基于 MLP-Mixer 的深度网络模块迭代更新目标的轨迹和特征：

1. **输入整合**：
   将当前的位置偏移、特征和相关性金字塔整合为一个输入矩阵：

$$\mathbf{Z}_t = \text{Concat}(\mathbf{X}_t - \mathbf{x}_1, \mathbf{F}_t, \mathbf{C}_{t,\text{pyramid}}) \tag{8}$$

2. **更新公式**：
   通过 MLP-Mixer 输出轨迹更新和特征更新：

$$\Delta\mathbf{X}_t, \Delta\mathbf{F}_t = \text{MLP-Mixer}(\mathbf{Z}_t) \tag{9}$$

更新轨迹和特征：

$$\mathbf{X}_{t+1} = \mathbf{X}_t + \Delta\mathbf{X}_t \tag{10}$$

$$\mathbf{F}_{t+1} = \mathbf{F}_t + \Delta\mathbf{F}_t \tag{11}$$

3. **可见性估计**：
   从特征轨迹中，通过线性映射预测目标的可见性分数：

$$v_t = \sigma(\mathbf{W} \cdot \mathbf{f}_t) \tag{12}$$

4. **迭代流程**：
   以上步骤重复 K 次，最终输出目标的轨迹 $\mathbf{X}_K$ 和可见性 $\mathbf{V}_K$。

---

7. 损失函数

7.1 位置轨迹监督

通过 L1 距离度量估计轨迹与真实轨迹之间的差异：

$$L_{\text{main}} = \sum_{k=1}^{K} \gamma^{K-k} \|\mathbf{X}_k - \mathbf{X}^*\|_1 \tag{13}$$

其中：

- $\mathbf{X}^*$ 是真实轨迹。
- γ 是递减权重因子。

7.2 可见性监督

通过交叉熵损失监督可见性预测：

$$L_{\text{ce}} = -\sum_{t=1}^{T} \left(v_t^* \log(v_t) + (1 - v_t^*) \log(1 - v_t) \right) \quad (14)$$

Experiment

1. 实验设置

1.1 数据集

本文在以下数据集上进行训练和评估：

1. **FlyingThings++**：
   - 基于 FlyingThings 数据集扩展，包含 8 帧视频序列。
   - 增加了合成遮挡，通过将一个物体叠加在另一视频帧上生成。
   - 主要用于训练和评估长时程轨迹。
2. **KITTI**：
   - 包含城市场景的视频数据，用于评估车辆和行人的轨迹。
   - 使用 10 FPS 的 8 帧子序列。
3. **CroHD**：

- 高分辨率 (1080 × 1920) 的拥挤人群数据集，跟踪人群头部运动。
- 帧率经过下采样，间隔更大。

4. **BADJA**：
   - 动物视频数据集，包含稀疏关键点标注。
   - 用于测试跨物体遮挡后的关键点传播性能。

1.2 基线方法

为了评估 PIPs 的效果，选择了以下具有代表性的基线方法：

1. **RAFT**：
   - 当前光流估计的 state-of-the-art 方法。
   - 使用逐帧连接的方式生成多帧轨迹。
2. **DINO**：
   - 基于视觉 Transformer 的特征匹配方法，通过最近邻方法生成轨迹。
3. **其他方法**：
   - Contrastive Random Walk (CRW)：基于时空图随机游走。
   - Memory-Augmented Self-supervised Tracker (MAST)：通过重建目标帧来学习对应关系。

1.3 评估指标

1. **轨迹误差（Trajectory Error）**：
   - 测量估计轨迹与真实轨迹之间的平均像素误差。
   - 公式表示：

$$\text{Error} = \frac{1}{T} \sum_{t=1}^{T} \|\mathbf{X}_t - \mathbf{X}_t^*\|_2 \tag{15}$$

     其中：
     - $\mathbf{X}_t$ 是估计轨迹在第 t 帧的位置。
     - $\mathbf{X}_t^*$ 是真实轨迹。
2. **遮挡鲁棒性（Occlusion Robustness）**：
   - 分别计算目标在可见（Visible）和遮挡（Occluded）条件下的误差，验证模型在遮挡场景中的性能。

---

2. 实验结果

2.1 FlyingThings++ 数据集结果

下表展示了 FlyingThings++ 数据集上的轨迹误差（单位：像素）：

方法	可见 (Vis.)	遮挡 (Occ.)
DINO [4]	40.68	77.76
RAFT [32]	24.32	46.73
PIPs (Ours)	**15.54**	**36.67**

- **分析**：
  - DINO 在遮挡情况下误差显著增大，因为匹配策略无法处理遮挡。
  - RAFT 对可见目标表现良好，但在遮挡目标时会发生漂移。
  - PIPs 能够通过时间先验重建遮挡后的轨迹，从而在两种条件下均表现最好。

2.2 KITTI 数据集结果

下表展示了 KITTI 数据集上的轨迹误差（单位：像素）：

方法	可见 (Vis.)	遮挡 (Occ.)
DINO [4]	13.33	13.45
RAFT [32]	**4.03**	6.79
PIPs (Ours)	4.40	**5.56**

- **分析**：
  - 在 KITTI 数据集中，目标运动较慢，RAFT 在可见目标上稍优于 PIPs。
  - 然而，在遮挡目标上，PIPs 更好地利用时间先验，取得更低的误差。

2.3 CroHD 数据集结果

下表展示了 CroHD 数据集上的轨迹误差（单位：像素）：

方法	可见 (Vis.)	遮挡 (Occ.)
DINO [4]	22.50	26.06
RAFT [32]	7.91	13.04
PIPs (Ours)	**5.16**	**7.56**

- **分析**：
  - PIPs 在高分辨率场景（如 CroHD 数据集）中表现出明显的优势。
  - 遮挡条件下，PIPs 的时间先验有效缓解了跟踪漂移问题。

2.4 BADJA 数据集结果

下表展示了关键点传播任务的 PCK-T 分数（越高越好）：

方法	Bear	Camel	Cows	Dog-A	Dog	Horse-H	Horse-L	平均 (Avg.)
DINO [4]	75.0	59.2	70.6	10.3	47.1	35.1	56.0	50.5
RAFT [32]	64.6	65.6	69.5	13.8	39.1	37.1	29.3	45.6
PIPs (Ours)	**76.3**	**81.6**	**83.2**	**34.2**	**44.0**	**57.4**	**59.5**	**62.3**

- **分析**：
  - PIPs 在大多数视频中获得了最好的关键点传播性能，平均分数领先其他方法。
  - DINO 在特定视频中表现较好，但对遮挡和复杂运动的鲁棒性不足。

---

3. 消融实验

3.1 时间先验的影响

移除时间先验后，PIPs 的遮挡条件下的轨迹误差显著增加，这验证了时间先验的重要性。

3.2 迭代次数的影响

通过实验发现，使用 $K = 6$ 次迭代时，PIPs 达到最优性能；进一步增加迭代次数对结果影响不显著。

---

4. 总结

实验表明，PIPs 方法在多个数据集和评估任务中均表现优异，尤其是在遮挡条件下能够显著超越现有方法。