Tracking Everything Everywhere All at Once

Presenter: Christopher Lawson

9/19

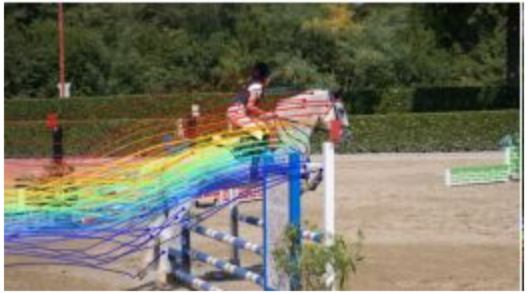
Problem - Motion Estimation

What is Motion Estimation?

- The creation of velocity trajectories made on an object to predict where it will be after X amount of time has passed.

Different Goals?

- Dense Pixel Trajectories
- Long-range Pixel trajectories



Why is Motion Estimation Important

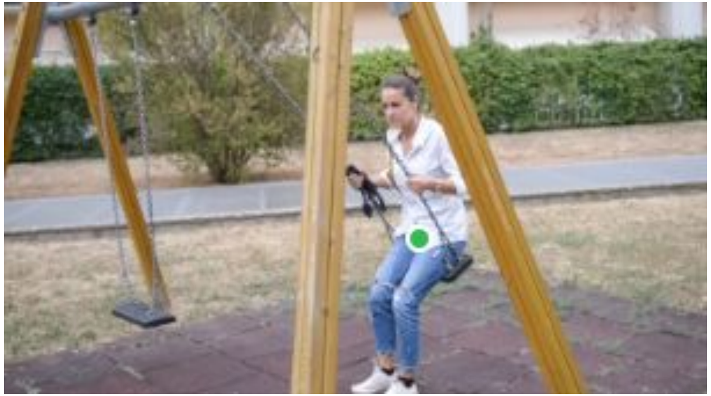
Incredibly important to solve tasks such as:

- How an object will behave
- Where an object will appear

These are important when interacting with the world

Challenges

- Maintaining accurate tracking across long sequences
- Tracking points through occlusions
- Maintaining coherence in space and time



Problem Proposal

A model that:

- Produces globally consistent full-length motion trajectories for all points in a video
- Can track points through occlusions
- Can tackle in-the-wild videos with any combination of camera and scene motion.

Related Work

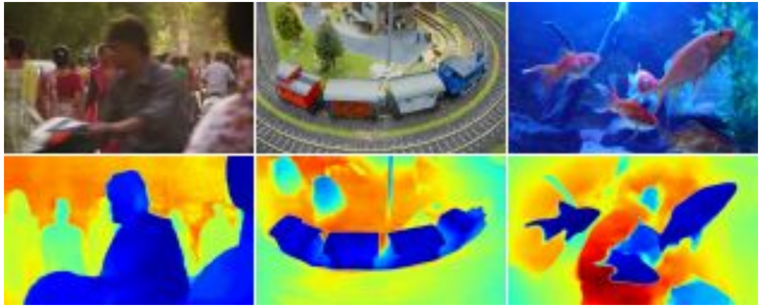
- Balanced trade offs for different results
  - i.e. precision for long-range predictability
- Overarching issue with tracking *all* pixels.

Related Work

- Video-based Motion Optimization
  - Produces a set of semi dense long-range trajectories from optical flow fields
  - Does not allow tracking through occlusions. Reappearing particles are treated as new entities
- Neural Video Representations
  - Uses coordinate-based multi-layer perceptrons to focus on problems such as novel view synthesis and video decomposition.
  - Can create mapping between frames but is expensive and unreliable.
  - Require known camera poses and thus predicted motion is often erroneous.

Approach - Overview

- Represents the video in a canonical 3D volume *G*
- Define a network *F* that maps each coordinate in *G* to a density σ and color *c*
- Density
    - Gives information about canonical space
    - Allows to track surfaces (even through occlusion)
- Color
    - Photometric Loss
    - Perceived Depth



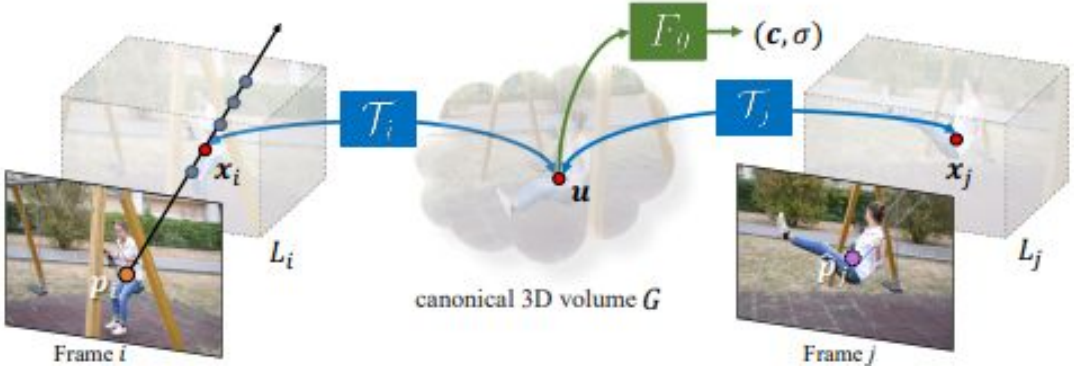
Approach - 3D Bijections

- Define bijective mapping ***T*** maps 3D points ***x*** from each coordinate in frame ***L*** to a canonical 3d coordinate frame called ***u***

$$\boldsymbol{x}_j = \mathcal{T}_{\boldsymbol{j}}^{-1} \circ \mathcal{T}_{\boldsymbol{i}}(\boldsymbol{x}_i).$$

- Can train these mappings as Invertible Neural Networks

$$\mathcal{T}_{\boldsymbol{i}}(\cdot) = M_{\boldsymbol{\theta}}(\cdot; \boldsymbol{\psi}_i)$$



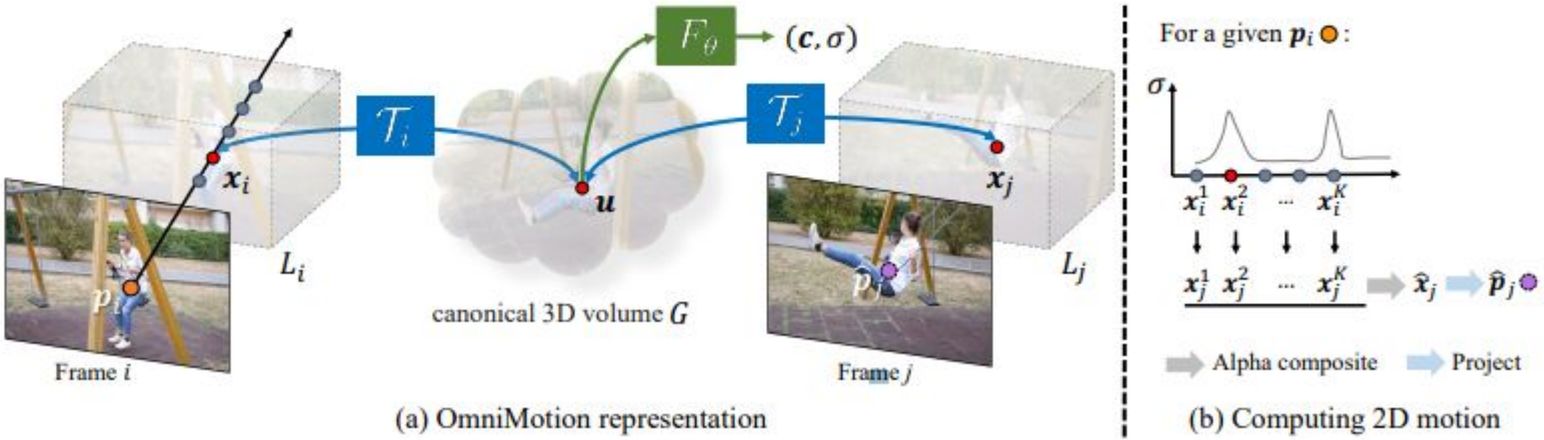
Approach - Frame-to-Frame Motion

- Now to describe 2D motion for a query pixel ***p*** in frame ***i***
- Lift the query pixel to 3D by sampling points on a ray (which contains $\{\boldsymbol{x}_i^k\}$ points), then map the 3D points to a target frame.

- Next obtain colors and densities through: $(\sigma_k, \boldsymbol{c}_k) = F_{\boldsymbol{\theta}}(M_{\boldsymbol{\theta}}(\boldsymbol{x}_i^k; \boldsymbol{\psi}_i))$
- Lastly aggregate points at the target frame through (method taken from NeRF):

$$\hat{\boldsymbol{x}}_j = \sum_{k=1}^{K} T_k \alpha_k \boldsymbol{x}_j^k, \text{ where } T_k = \prod_{l=1}^{k-1} (1 - \alpha_l)$$

Approach - All together



Optimization

- Model works but now needs to train and learn
- Broken into 3 steps

Collect Input Motion Data

Apply Loss

Supervision through Hard Mining

Collecting Input Motion Data

- Done so through using different methods to compute pairwise correspondence
  - RAFT and TAP-Net
- Next compute all pairwise optical flows
- Apply cycle consistency and appearance consistency to filter out spurious correspondence.
- Helps reduce noise but still need more methods

Applying Loss function

- Trying to minimize predicted flow
- Minimize the photometric loss
- Minimize 3D acceleration between points in frame i+1 and i-1
- Total loss is the summation of all 3 losses

$$\mathcal{L}_{\text{flo}} = \sum_{\boldsymbol{f}_{i\to j}\in\Omega_f} ||\hat{\boldsymbol{f}}_{i\to j} - \boldsymbol{f}_{i\to j}||_1$$

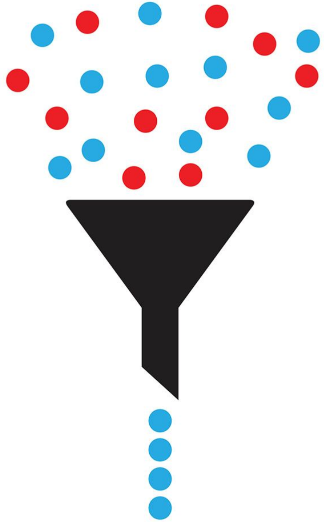
$$\mathcal{L}_{\text{pho}} = \sum_{(i,\boldsymbol{p})\in\Omega_p} ||\hat{\boldsymbol{C}}_i(\boldsymbol{p}) - \boldsymbol{C}_i(\boldsymbol{p})||_2^2$$

$$\mathcal{L}_{\text{reg}} = \sum_{(i,\boldsymbol{x})\in\Omega_x} ||\boldsymbol{x}_{i+1} + \boldsymbol{x}_{i-1} - 2\boldsymbol{x}_i||_1$$

$$\mathcal{L} = \mathcal{L}_{\text{flo}} + \lambda_{\text{pho}}\mathcal{L}_{\text{pho}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}$$

Supervision via Hard Mining

- Lots of data points through pairwise flow, some that is not important/rigid
  - Background pixels remain relatively constant across frames
- Need a way to filter to more important data
- Calculating Euclidean error map to guide the sampling process during optimization.

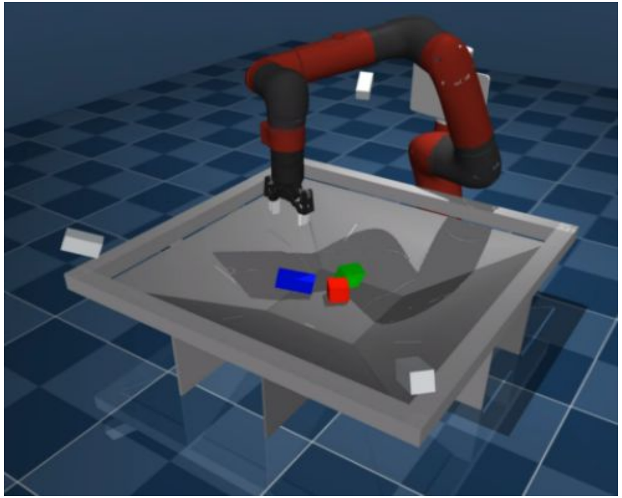


Experimental Setup

Datasets were taking from TAP-Vid

DAVIS
30 videos
34 - 104 frames per video
21.7 point track annotations

Kinetics
1,189 videos
250 frames per video
26.3 point track annotations



RGB-Stacking
50 videos
250 frames per video
30 point track annotations

Experimental work

Four Main Metrics Used:

- $< \delta^x_{avg}$
  - Average position accuracy of visible points across 1, 2, 4, 8 and 16 pixels
- Temporal Coherence (TC)
  - Temporal coherence of tracks by measuring L2 Norm between acceleration of ground-truth tracks and predicted tracks
- Occlusion Accuracy (OA)
  - Accuracy of visibility/occlusion at each frame
- Average Jaccard (AJ)
  - Evaluates occlusion and position accuracy on same thresholds as above.

Experimental work

Baselines

- RAFT
  - 2-frame optical flow method to generate multi-frame trajectories.
- PIP
  - Method for estimating multi-frame point trajectories that handle occlusions
  - Set to use temporal window of 8 frames
- Flow Walk
  - A multi-scale contrastive random walk to learn space-time correspondences
- TAP-Net
  - Uses cost volume to predict the location of a query point in a target frame
- Deformable Sprites
  - A layer based video decomposition method. Similar to the work at hand, but does not directly produce frame-to-frame correspondence.

Experimental Results - Qualitative

Experimental Results - Quantitative

Method	Kinetics				DAVIS				RGB-Stacking			
	AJ ↑	$< \delta^x_{\text{avg}}$ ↑	OA ↑	TC ↓	AJ ↑	$< \delta^x_{\text{avg}}$ ↑	OA ↑	TC ↓	AJ ↑	$< \delta^x_{\text{avg}}$ ↑	OA ↑	TC ↓
RAFT-C [62]	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 [62]	50.6	66.9	85.5	3.00	34.1	48.9	76.1	9.83	72.1	<u>85.1</u>	92.1	1.04
TAP-Net [14]	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 [21]	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	<u>0.77</u>	35.2	51.4	80.6	0.90	41.3	55.7	<u>92.2</u>	<u>0.13</u>
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 [74]	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)	<u>53.8</u>	<u>68.3</u>	<u>88.8</u>	<u>0.77</u>	<u>50.9</u>	<u>66.7</u>	**85.7**	<u>0.86</u>	<u>73.4</u>	84.1	<u>92.2</u>	**0.11**
Ours (RAFT)	**55.1**	**69.6**	**89.6**	**0.76**	**51.7**	**67.5**	<u>85.3</u>	**0.74**	**77.5**	**87.0**	**93.5**	<u>0.13</u>

Ablation Study

3 different ablation tests:

- No invertible
  - Replaces Invertible mapping network with a separate forward and backward mapping network between frames (without bijections)
- No Photometric
  - Omits the photometric loss from loss function
- Uniform sampling
  - Replaces hard-mining sampling strategy with uniform sampling strategy

Method	TC $\downarrow$	$< \delta^x_{avg} \uparrow$
No invertible	0.97	21.4
No photometric	0.83	58.3
Uniform sampling	0.88	61.8
Full	**0.74**	**67.5**

Method	AJ $\uparrow$	OA $\uparrow$
No invertible	12.5	76.5
No photometric	42.3	84.1
Uniform sampling	47.8	83.6
Full	**51.7**	**85.3**

Discussion

- Where do we set the trade off between having incredibly accurate systems and having incredibly high computation and training costs?

- How can we develop more memory sparing methods for object tracking once we are able to expand these models to longer and longer videos?

- How long can an object remain occluded before the model should forget about it? Or should it even be forgotten at all?

Limitation

- Rapid and highly non-rigid motion
- Thin Structures
  - Fail to provide enough reliable correspondences
- Caught in local minima
  - Due to the highly non-convex nature of the data
- Computationally expensive
  - Pairwise flows which scale quadratically

Future Work

- More efficient pairwise matching
- Better optimization process
  - NeRF -> Block NeRF
  - Neural Graphics Primitives

Further Readings

- Peter Sand and Seth Teller. Particle video: Long-range motion estimation using point trajectories. Int. J. of Computer Vision, 80:72–91, 2008
- Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens Continente, Kucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. In NeurIPS Datasets Track, 2022
- Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM, 65(1):99–106, 2021

Summary

- **Problem:** Created a new test-time optimized method for motion estimation
- **Limitations:** Very expensive and not very good at fine tracking
- **Strengths:** Deals the best with occlusion than other methods

- OmniMotion can estimate complete and globally consistent motion for an entire video.
- Does so by introducing a quasi-3D canonical volume and a per-frame local bijection to produce accurate and smooth long-range tracking through occlusions.

Thank you!