Stony Brook University

Visual Object Tracking

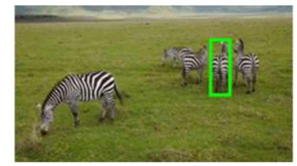
What is Visual Tracking

Visual Tracking

Keep track of something across time

t

Object defined manually or by detection



Model-free single object tracking

Videos from LaSOT (Fan et al. CVPR 2019)

What is Visual Tracking

Visual Tracking

Keep track of something across time

t

Plane/Pose

Video results from Chen et al., ICRA 2017

What is Visual Tracking

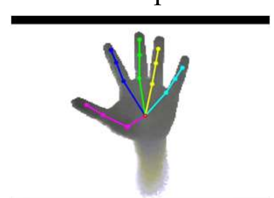
Visual Tracking

Keep track of something across time

t

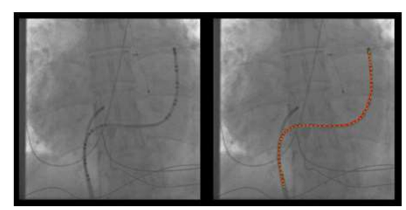
Deformable structures

Hand pose



Li et al., ICCV 2015

Curvilinear structure



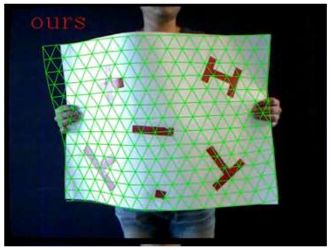
Chu et al., MICCAI 2016

What is Visual Tracking

Visual Tracking

Keep track of something across time t

Deformable surface



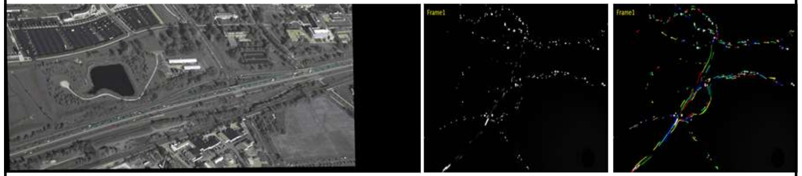
Wang et al., ICCV 2019

What is Visual Tracking

Visual Tracking

Keep track of something across time t

Multiple targets



Highway vehicles

Neutrophil cell motion

Shi et al., CVPR 2013

Fan et al., EMBC 2017

What is Visual Tracking

Visual Tracking

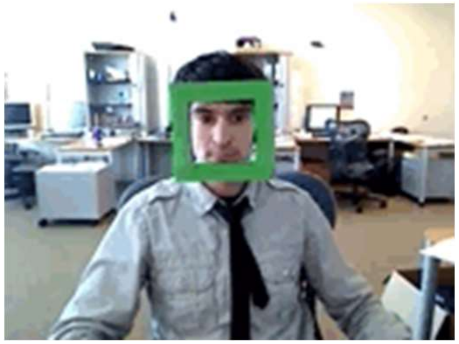
Keep track of something across time t

Camera pose & environment geometry (SLAM)

Oblique corridor - The scene contains multiple Manhattan worlds

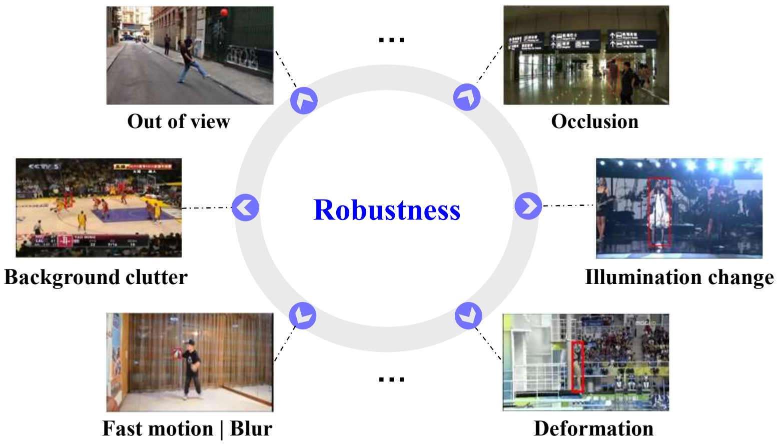
Zou, et al., R-Robotics 2019

Model-Free Tracking



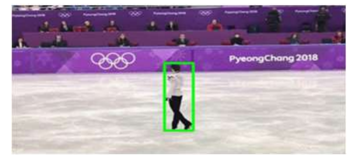
http://vision.ucsd.edu/sites/default/files/fast.gif

Challenges - Robustness



Challenges - Efficiency

Real-time performance (≥25 fps)


A tracker running in ≥25 fps


A tracker running in 10 fps

Deployment of a tracker

Faster in speed, smaller in storage, and lower in power consumption.

Problem formulation

Input:

- A sequence of images (or volumes in 3D),
  - $I_0, I_1, \ldots, I_t, \ldots$
- Target of interest at the initial frame I_0
  - x_0

Output:

- Targets in each of the following frames
  - $x_1, \ldots, x_t, \ldots$

What is a “target”?

- Model a target as a state (vector)

Bounding box
No rotation: x=(pos_x, pos_y, width, height)
With rotation: x=(pos_x, pos_y, width, height, θ)
Affine: x=(a, b, c, d, trans_x, trans_y)

Pose
x=(box_1, box_2, …, box_n)

Contour
x=(point_1, point_2, …, point_n)

Tracking and Detection

Detection: for the t-th frame, I_t, find x_t

$$x_t = \mathrm{argmax}_x \Pr(x|I_t)$$

"Tracking by detection"

Why do we still need tracking?

- Temporal consistency
- Robustness
- Efficiency
- Resolving ambiguity, multiple targets association

Tracking as State Estimation

At frame t, find the best x_t by Bayesian inference

$$\Pr(x_t|I_t, I_{t-1}, \dots, I_1, I_0)$$

In practice, we don't usually use the whole image, instead some part or local features of the image.

Named as observation

$z_1, \dots, z_t, \dots$

So the problem is

$$\Pr(x_t|X_{t-1}, Z_t)$$

where

$X_t = \{x_1, \dots, x_t\}$, $Z_t = \{z_1, \dots, z_t\}$

Tracking as State Estimation

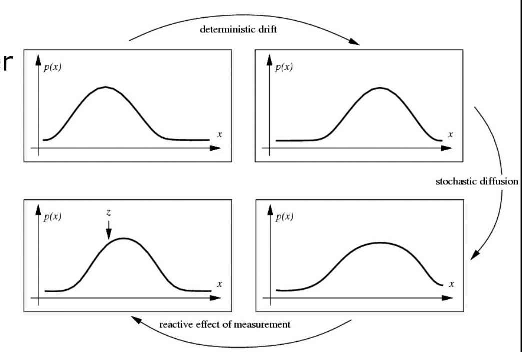
By Markov property

$$\Pr(x_t|X_{t-1}, Z_t) \rightarrow \Pr(x_t|Z_t)$$

- Once we know $\Pr(x_t|Z_t)$, tracking can be easily done by find the peak of $\Pr(x_t|Z_t)$ or the expectation of x_t.
- Problems
  - Is there a closed form of $\Pr(x_t|Z_t)$?
  - How to efficiently estimate $\Pr(x_t|Z_t)$?
- Solution
  - Treat tracking as a probability propagation problem
  - Sampling techniques

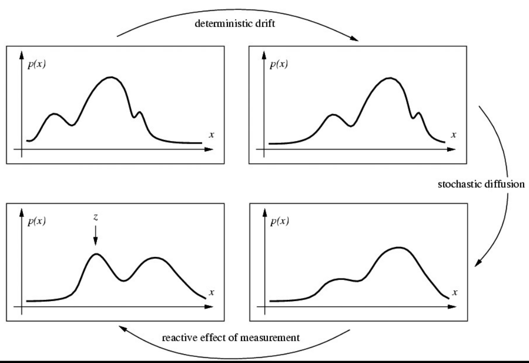
Gaussian

- Assume Gaussian distribution
  - "Everything" is Gaussian then
- Propagation
  - Kalman filter



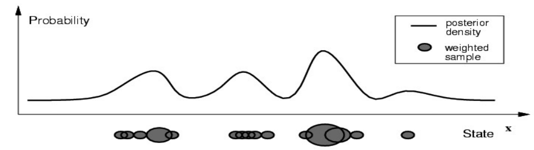
General Non-Gaussian

- No closed form representation
- No closed form propagation solution



CONDENSATION – Factored Sampling

CONDENSATION -- conditional density propagation for visual tracking, M. Isard and A. Blake, IJCV 29(1):5-28, 1998



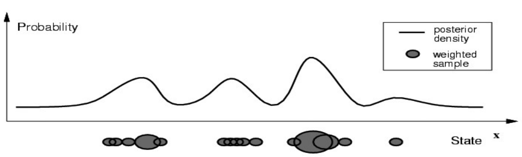
- Non-parametric representation

$$p(x \mid z) = kp(z \mid x)\, p(x)$$

Observation likelihood   Prior density

- Prediction: sample state x from prior density
- Correction: Weight the samples according to observation

CONDENSATION – Factored Sampling



$$\{(s^{(1)}, \pi^{(1)}), \dots, (s^{(N)}, \pi^{(N)})\}$$ Weighted samples

$$s^{(i)} \sim p(x)$$ Prior density

$$\pi^{(i)} = \frac{p(z \mid s^{(i)})}{\sum_{j=1..N} p(z \mid s^{(j)})}$$ Observation likelihood

We are not done yet …

$$\{(s^{(1)}, \pi^{(1)}), \dots, (s^{(N)}, \pi^{(N)})\}$$

$$\Downarrow$$

$$\{(s_t^{(1)}, \pi_t^{(1)}), \dots, (s_t^{(N)}, \pi_t^{(N)})\}$$

- We need to model temporal information
  - For prediction, we need prior from ALL previous observation Z_{t-1}

$$p(x_t \mid Z_{t-1}) = \int_{x_{t-1}} p(x_t \mid x_{t-1})\, p(x_{t-1} \mid Z_{t-1})\, dx_{t-1}$$

  - We then update the posterior probability

$$p(x_t \mid Z_t) = k_t\, p(z_t \mid x_t)\, p(x_t \mid Z_{t-1})$$

Components in the tracker

$$p(x_t \mid Z_{t-1}) = \int_{x_{t-1}} p(x_t \mid x_{t-1})\, p(x_{t-1} \mid Z_{t-1})\, dx_{t-1}$$

$$p(x_t \mid Z_t) = k_t\, p(z_t \mid x_t)\, p(x_t \mid Z_{t-1})$$

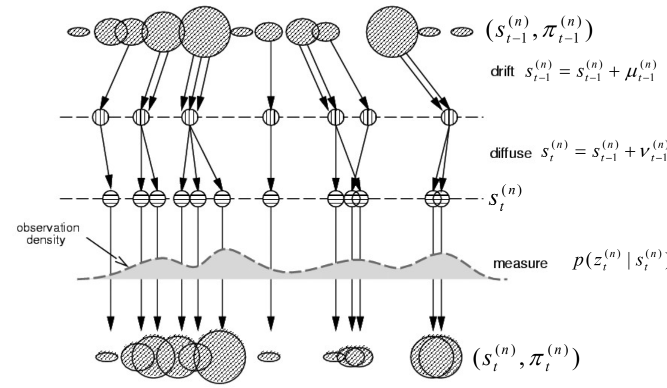
x_t : state, target representation

z_t : observation, object feature

$p(x_t \mid x_{t-1})$: state transition probability (drift, motion, etc)

$p(z_t \mid x_t)$: observation likelihood

Particle filtering



Iterate

From the "old" sample-set $\{\mathbf{s}_{t-1}^{(n)}, \pi_{t-1}^{(n)}, c_{t-1}^{(n)}, n = 1, \ldots, N\}$ at time-step $t-1$, construct a "new" sample-set $\{\mathbf{s}_t^{(n)}, \pi_t^{(n)}, c_t^{(n)}\}, n = 1, \ldots, N$ for time t.

Construct the n^{th} of N new samples as follows:

1. **Select** a sample $\mathbf{s}_t'^{(n)}$ as follows:
   (a) generate a random number $r \in [0, 1]$, uniformly distributed.
   (b) find, by binary subdivision, the smallest j for which $c_{t-1}^{(j)} \geq r$
   (c) set $\mathbf{s}_t'^{(n)} = \mathbf{s}_{t-1}^{(j)}$
2. **Predict** by sampling from

$$p(\mathbf{x}_t \mid \mathbf{x}_{t-1} = \mathbf{s}_t'^{(n)})$$

to choose each $\mathbf{s}_t^{(n)}$. For instance, in the case that the dynamics are governed by a linear stochastic differential equation, the new sample value may be generated as: $\mathbf{s}_t^{(n)} = A\mathbf{s}_t'^{(n)} + B\mathbf{w}_t^{(n)}$ where $\mathbf{w}_t^{(n)}$ is a vector of standard normal random variates, and BB^T is the process noise covariance — see section 5.

3. **Measure** and weight the new position in terms of the measured features $\mathbf{z}_t$:

$$\pi_t^{(n)} = p(\mathbf{z}_t \mid \mathbf{x}_t = \mathbf{s}_t^{(n)})$$

then normalise so that $\sum_n \pi_t^{(n)} = 1$ and store together with cumulative probability as $(\mathbf{s}_t^{(n)}, \pi_t^{(n)}, c_t^{(n)})$ where

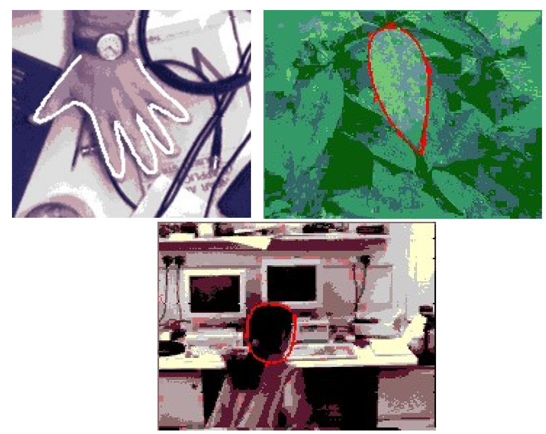
$$\begin{aligned} c_t^{(0)} &= 0, \\ c_t^{(n)} &= c_t^{(n-1)} + \pi_t^{(n)} \quad (n = 1, \ldots, N). \end{aligned}$$

Once the N samples have been constructed: **estimate**, if desired, moments of the tracked position at time-step t as

$$\mathcal{E}[f(\mathbf{x}_t)] = \sum_{n=1}^{N} \pi_t^{(n)} f\left(\mathbf{s}_t^{(n)}\right)$$

obtaining, for instance, a mean position using $f(\mathbf{x}) = \mathbf{x}$.

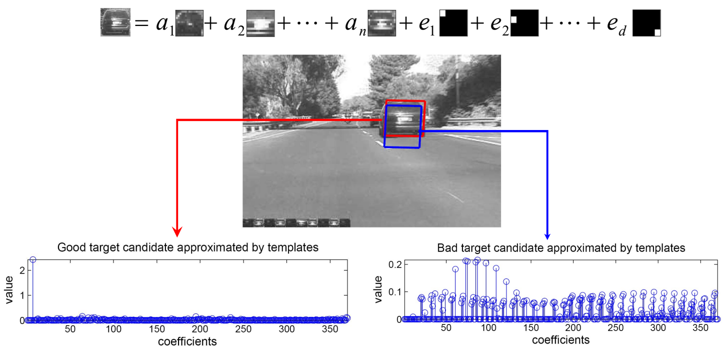
Particle filtering results



http://www.robots.ox.ac.uk/~misard/condensation.html

Sparse Representation

A candidate is approximated by a sparse combination of templates.



Mei & Ling, ICCV 2009

Multiple-Instance Representation

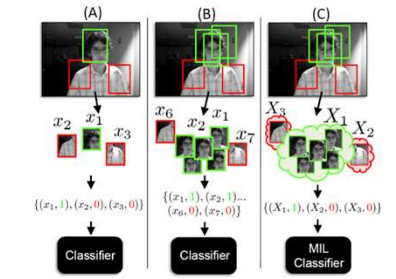


Figure 1. **Updating a discriminative appearance model:** (A) Using a single positive image patch to update a traditional discriminative classifier. The positive image patch chosen does not capture the object perfectly. (B) Using several positive image patches to update a traditional discriminative classifier. This can confuse the classifier causing poor performance. (C) Using one positive bag consisting of several image patches to update a MIL classifier. See Section 3 for empirical results of these three strategies.

A candidate region contains multiple instances (at least one positive).

Leverage the tool of multiple instance learning (MIL).

Sliding window for local search.

Babenko, Yang, & Belongie, CVPR 2009

Structured Inference

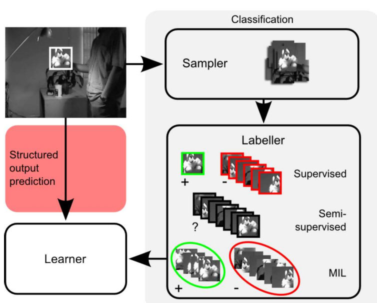


Figure 1. Different adaptive tracking-by-detection paradigms: given the current estimated object location, traditional approaches (shown on the right-hand side) generate a set of samples and, depending on the type of learner, produce training labels. Our approach (left-hand side) avoids these steps, and operates directly on the tracking output.

Modeling structure information in visual representation.

Structured kernel method for direct output prediction.

Hare, Saffari, & Torr, ICCV 2011

Observations and Motivations

- Representation is important.
- Number of particles is important.
- Particle filter versus sliding window?
- Speed versus accuracy?

Conjecture

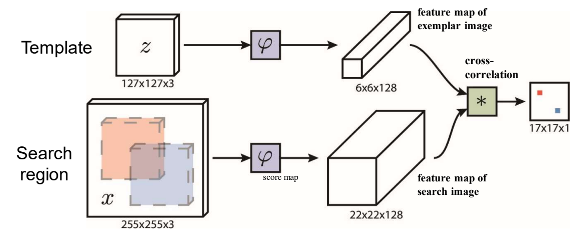
- Effective representation + efficient sliding window

Correlation Filter Tracking

- Early ones with (nearly) raw intensity features
  - Learned "template matching" in frequency domain
  - Extremely efficient, but less impressive
  - MOSSE: Bolme et al. CVPR 2010
- Advanced solutions with richer features
  - KCF with HOG, Henriques et al. PAMI 2015
  - With deep features, Danelljan et al. 2015, Ma et al. 2015, Qi et al. 2015, …
- Bring correlation filter into an end-to-end pipeline
  - Siamese tracking, Bertinetto et al. ECCVW 2016

Deep Similarity for Tracking

SiamFC: Fully Convolutional Siamese Network for Tracking.
Bertinetto *et al.* ECCVW 2016

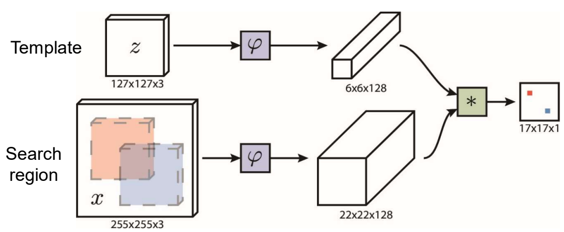


z: exemplar image
x: search image
φ: a convolutional embedding function (table1 in the paper)
function $f(z,x) = \varphi(z) * \varphi(x) + b1$

Each pixel value in score map represents the similarity of each position in the search image to the exemplar image. The higher the score, the more similar they are.

Siamese Tracking

SiamFC: Fully Convolutional Siamese Network for Tracking. Bertinetto *et al.* ECCVW 2016



Representation
- Template and search region go through the same feature extraction network
- No model update

Tracking inference
- Correlation (kind of template matching), the maximum in the response map
- No model update
- Efficient (50+ fps)

Architecture in Detection

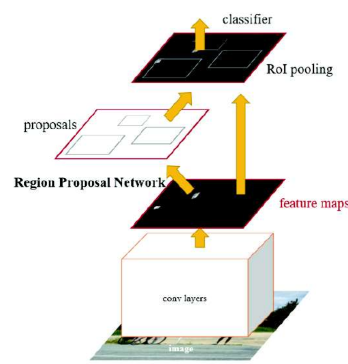


Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the 'attention' of this unified network.

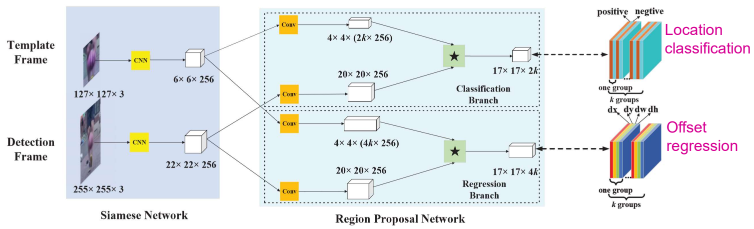
Can we borrow something from detection?

Region proposal network (RPN):
- reduce the number of candidate targets
- improve reference accuracy.

Ren et al, "Faster R-CNN: Towards Real-Time Object Detection, NeurIPS 2015.

SiamRPN

SiamRPN: Siamese Region Proposal Network for Tracking. Li et al., CVPR 2018



Representation

- Siamese network
- No target representation update, no representation pyramid for scale estimation

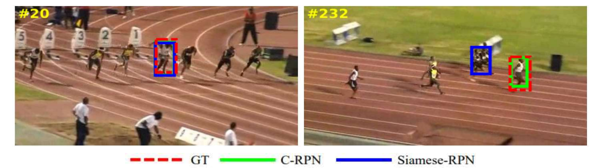
Tracking inference

- Classification for localization, regression for result improvement
- Even faster (160+ fps)

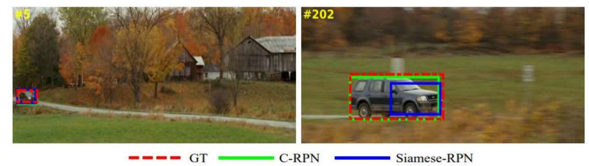
Open Issues

- Distractors and scale variation

Drift in presence **similar distractors** (hard negative)

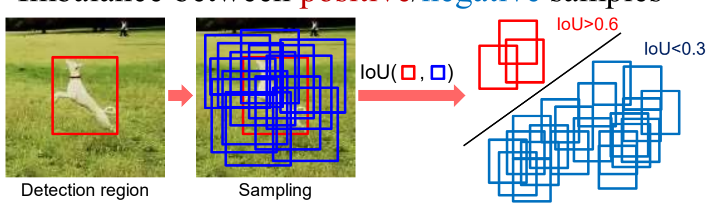


Inaccurate scale estimation when **scale changes heavily**

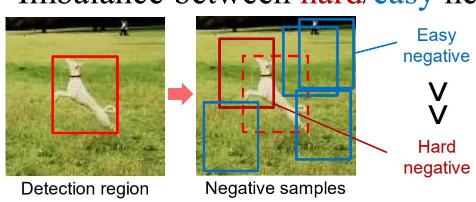


Training Sample Imbalance

- Imbalance between positive/negative samples

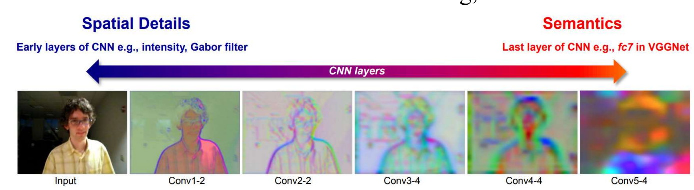


- Imbalance between hard/easy negative samples

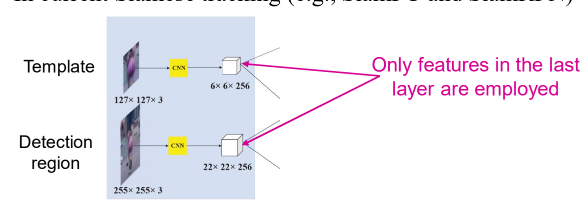


Low-level Feature Under-Exploited

- Hierarchical CNN features for visual tracking, Ma *et al.* ICCV 15

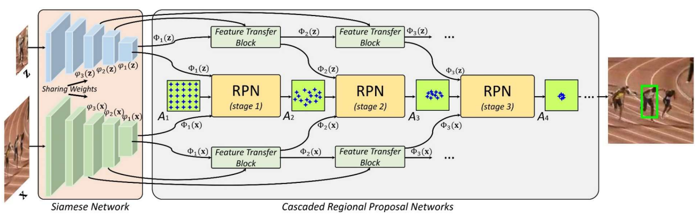


- In current Siamese tracking (e.g., SiamFC and SiamRPN)



Cascaded Siamese Region Proposal Networks

Cascaded Siamese Region Proposal Networks for Real-time Visual Tracking
Fan & Ling, CVPR 2019

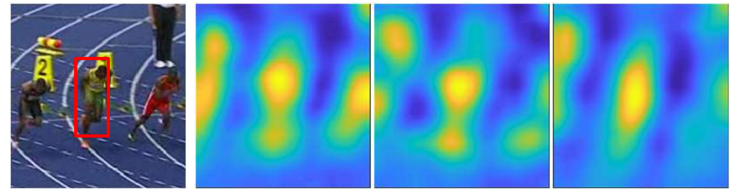


Three key ingredients for improvement

- Filtering out easy negative samples → More balanced training samples → Sequentially more discriminative classifier in RPN
- Multi-layer feature fusion via FTB → stronger classifier and regressor in RPN
- Multi-step regression in C-RPN → more accurate scale estimation

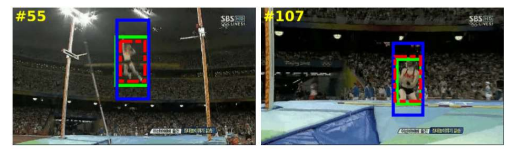
Illustrative Results

- Prediction maps of RPNs of different stages



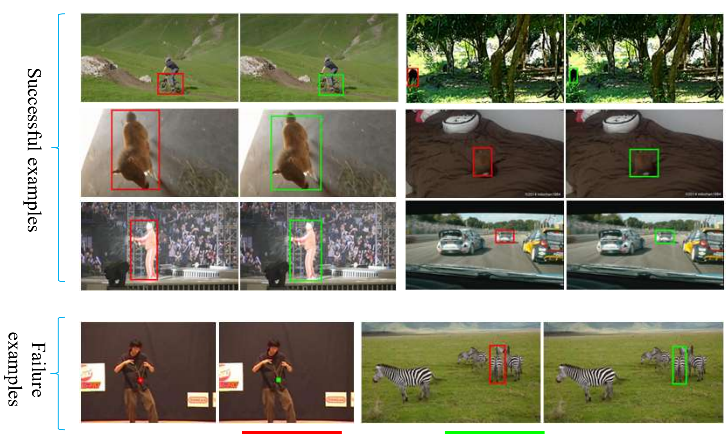
(a) Region of interest (b) From left to right: response maps of stage 1, stage 2 and stage 3

- Multi-stage regression



GT C-RPN Siamese-RPN

Qualitative Results on LaSOT



GT C-RPN

Since 2013

Example benchmarks

- CVPR 2013, Object Tracking Benchmark (OTB-2013)
- PAMI 2014, ALOV 300
- TIP 2015, Temple Color 128 (TC-128)
- PAMI 2015, Object Tracking Benchmark (OTB-2015)
- PAMI 2016, NUS-PRO
- ECCV 2016, UVA123
- ICCVW 2017, Visual Object Tracking (VOT 2017)
- ICCV 2017, Need For Speed (NFS)
- ECCV 2018, long-term tracking in the wild (TLP)
- CVPR 2019, Large scale Single Object Tracking (LaSOT)
- …

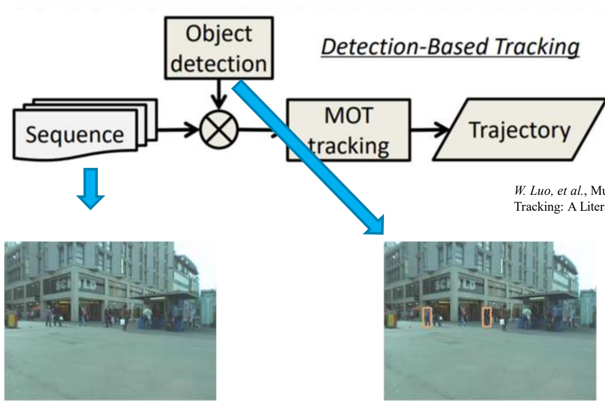
LaSOT Large-Scale Single Object Tracking

Benchmark	Videos	Min frames	Mean frames	Median frames	Max frames	Total frames	Total duration	frame rate	Absent labels	Object classes	Class balance	Num. of attributes	Lingual feature
OTB-2013 [52]	51	71	578	392	3,872	29**K**	16.4 **min**	30 fps	✗	10	✗	11	✗
OTB-2015 [53]	100	71	590	393	3,872	59**K**	32.8 **min**	30 fps	✗	16	✗	11	✗
TC-128 [35]	128	71	429	365	3,872	55**K**	30.7 **min**	30 fps	✗	27	✗	11	✗
VOT-2014 [26]	25	164	409	307	1,210	10**K**	5.7 **min**	30 fps	✗	11	✗	n/a	✗
VOT-2017 [27]	60	41	356	293	1,500	21**K**	11.9 **min**	30 fps	✗	24	✗	n/a	✗
NUS-PRO [28]	365	146	371	300	5,040	135**K**	75.2 **min**	30 fps	✗	8	✗	n/a	✗
UAV123 [39]	123	109	915	882	3,085	113**K**	62.5 **min**	30 fps	✗	9	✗	12	✗
UAV20L [39]	20	1,717	2,934	2,626	5,527	59**K**	32.6 **min**	30 fps	✗	5	✗	12	✗
NfS [14]	100	169	3,830	2,448	20,665	383**K**	26.6 **min**	240 fps	✗	17	✗	9	✗
GOT-10k [22]	10,000	-	-	-	-	1.5M	-	10 fps	✓	563	✗	6	✗
LaSOT	1,400	1,000	2,506	2,053	11,397	3.52M	32.5 **hours**	30 fps	✓	70	✓	14	✓

- ❑ **Large-scale**
  - 1,400 sequences
  - > 3.52M frames
- ❑ **High-quality**
  - Each frame is manually annotated
- ❑ **Diversity**
  - 70 categories in total
- ❑ **Long-term**
  - Average >2,500 frames per seq
- ❑ **Balance**
  - 20 sequence for each category
- ❑ **Comprehensive**
  - Language description provided

LaSOT Large-Scale Single Object Tracking

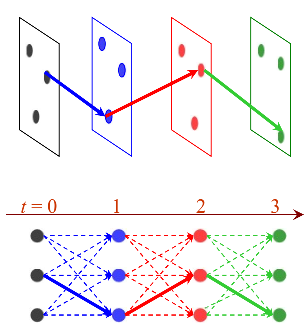
Multiple Object Tracking



W. Luo, et al., Multiple Object Tracking: A Literature Review

Problem Formulation

A toy example: K= 3, N=3

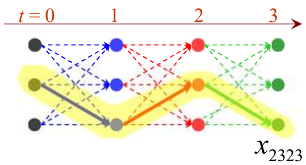


Input:
- K+1 frames
  - "+1" for convenience
- Each with N detections
  - Usually with false alarms
  - N can be frame-dependent

Output:
- Trajectories over time

Multi-Dimensional Assignment

(K+1)-dimensional assignment,
(K+1)-partite problem



$$\min_{X=\{x_{i_0 i_1 \ldots i_K}\}} \sum_{i_0}^{N}\sum_{i_1}^{N}\cdots\sum_{i_K}^{N} c_{i_0 i_1 \ldots i_K}\, x_{i_0 i_1 \ldots i_K}$$

Trajectory indicator

(K+1)-order trajectory cost

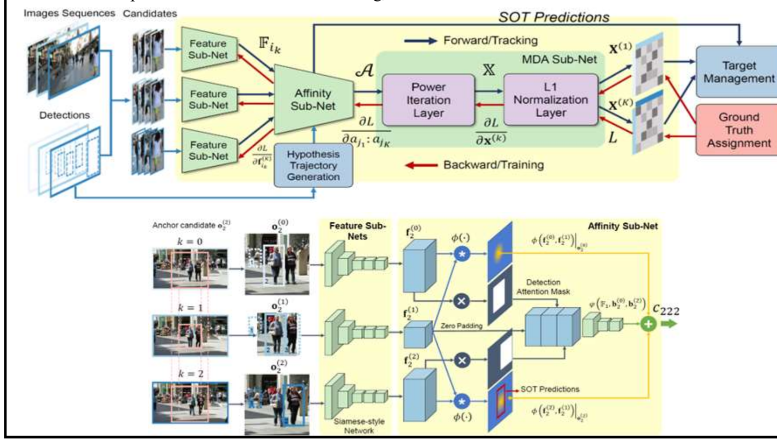
$$s.t.\begin{cases}\displaystyle\sum_{\{i_0,i_1,\ldots,i_K\}/i_k} x_{i_0 i_1 \ldots i_K} = 1, \quad k = 0,1,\ldots,K, \quad i_k = 1,\ldots,N \\ x_{i_0 i_1 \ldots i_K} \in \{0,1\}, \quad i_0,\ldots,i_K = 1,\ldots,N\end{cases}$$

Trajectory disjoint constraints

Binary constraints

End-to-End Multi-Object Tracking

FAMNet: Learning **F**eature extraction, **A**ffinity estimation, and **M**ulti-dimensional assignment in a unified deep neural network. P. Chu & H. Ling, ICCV 2019.



Qualitative Results

MOT2015

Test Set

11 Videos

(First 100 Frames Each)

New Trend in Tracking

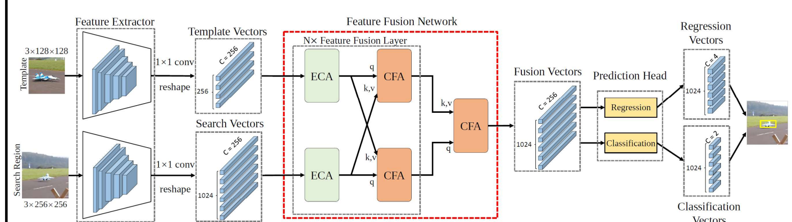


Figure 2. Architecture of our Transformer tracking framework. This framework contains three fundamental components: feature extraction backbone, feature fusion network, and prediction head. The proposed attention-based feature fusion network is naturally applied on the Siamese-based feature extraction backbone.

Transformer modules:

- **Self-attention:** Ego-Context Augment
- **Cross-attention:** Cross-Feature Augment

Transformer Tracking, Chen, Yan, Zhu, Wang, Yang, & Lu, CVPR 2021

New Trend in Tracking

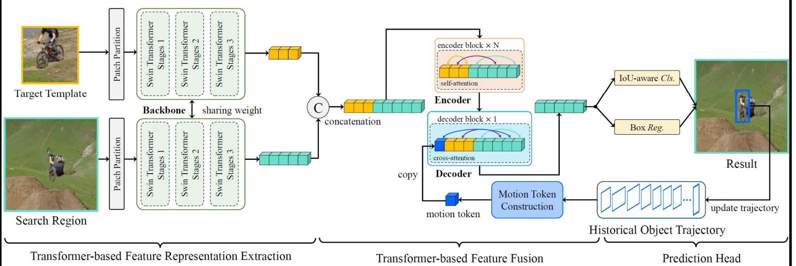


Figure 2: Architecture of SwinTrack, which contains three parts including Transformer-based feature representation extraction, Transformer-based feature fusion and prediction head. Our SwinTrack is a simple and neat tracking framework without complex designs such as multi-scale features or temporal template updating, yet demonstrating state-of-the-art performance. *Best viewed in color.*

Transformer modules:

- **Swin Transformer backbone**
- **Motion token:** dynamically update object trajectory information

SwinTrack, Lin, Fan, Zhang, Xu, & Ling, NeurIPS 2022

New Trend in Tracking

Transformer for Multiple Object Tracking

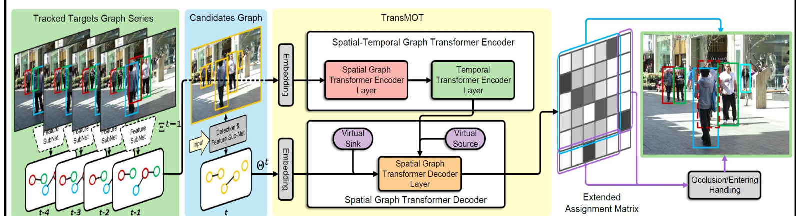


Figure 1. Overview of the proposed TransMOT pipeline for online MOT. The trajectories graph series Ξ^{t-1} till frame $t-1$ and detection candidates graph Θ^t at frame t serve as the source and target inputs, respectively, to the spatial-temporal graph transformer.

TransMOT: Spatial-Temporal Graph Transformer for Multiple Object Tracking
Chu, Wang, You, Ling, & Liu, WACV 2023