Parallel Tracking and Verifying (PTAV): A Framework for Real-Time and High Accuracy Visual Tracking

Heng Fan and Haibin Ling

Meitu HiScene Lab, HiScene Information Technologies, Shanghai, China
Department of Computer and Information Sciences, Temple University, Philadelphia, PA USA

Project & Code: ***http://www.dabi.temple.edu/~hbling/code/PTAV/ptav.htm***

ICCV, 2017

Visual Tracking

Video *Basketball*

Video *Coke*

time

- Goal: to locate an arbitrary target in a video with its initial position.
  - *Model-free:* agnostic to the object's class
  - *Single-object tracking*
- Applications: video surveillance, robotics, human-computer interactions, etc.
  - *Accuracy*
  - *Efficiency:* real-time requirement

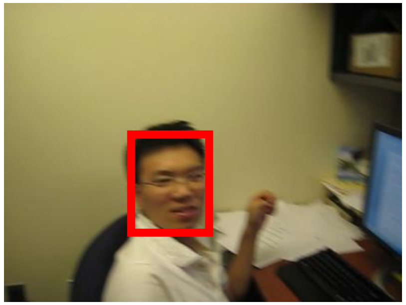
Challenges

- Challenge I: appearance variations

Occlusion

Scale & Illumination changes

Rotation



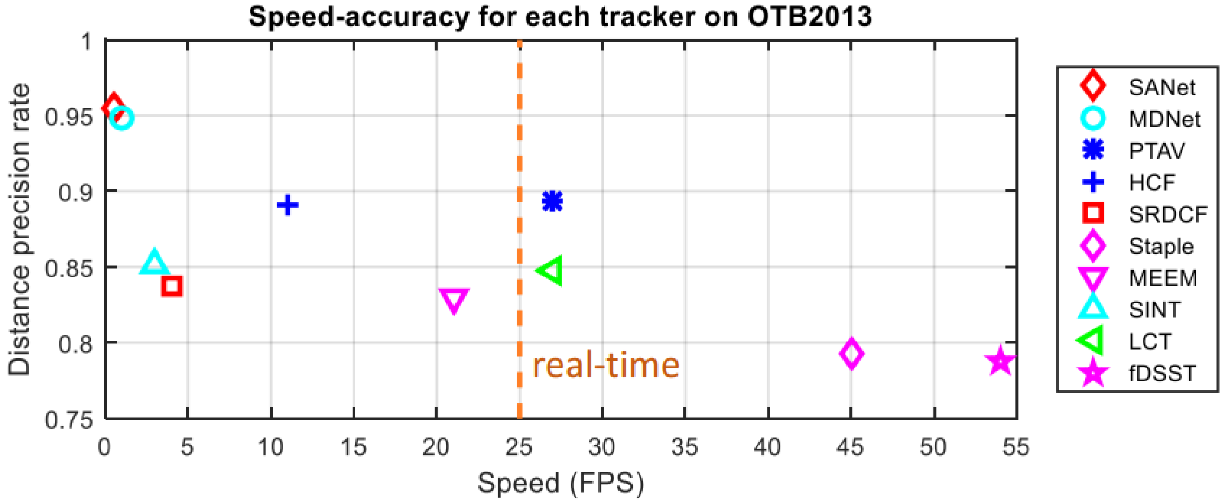
Motion blur

Scale changes

Deformation & Background Clutter

- Challenge II: real-time requirement

Motivations



Speed and accuracy plot on OTB2013 (CVPR'13). For better illustration, only those trackers with accuracy higher than 0.75 are reported. The proposed PTAV algorithm achieves the best accuracy among all real-time trackers.

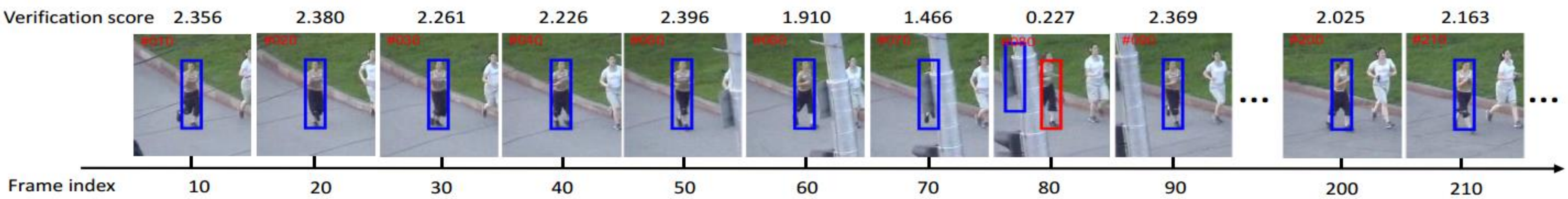
- Existing tracking approaches

Challenges	Solution	Pros & Cons	Representatives
Challenge I: appearance variations	robust deep features	Pros: robust to appearance variations; Cons: high computational burden;	SANet (CVPRW'17), MDNet (CVPR'16), HCF (ICCV'15), SINT (CVPR'16), etc.
Challenge II: real-time requirement	simple hand-crafted features	Pros: efficient computation; Cons: sensitive to appearance variations;	KCF (TPAMI'15'), MOSSE (CVPR'10), fDSST (TPAMI'17), Staple (CVPR'16), etc.

- Real-time & high quality trackers remain scarce
  - Trade off between accuracy and speed

Motivations

● Motivation I: smooth object appearance changes in video



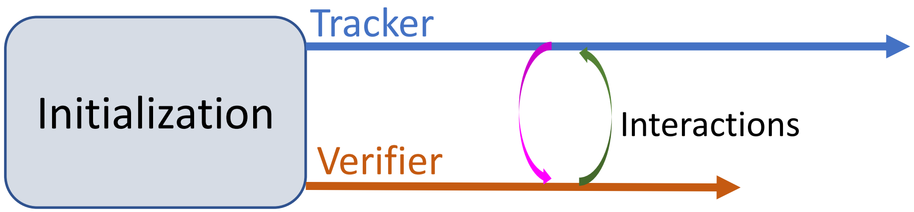
Verifying tracking results on a typical sequence. Verifier validates tracking results every 10 frames. Most of the time the tracking results are reliable (showing in blue). Occasionally, *e.g*. frame #080, the verifier finds the original tracking result (showing in blue) unreliable and the tracker is corrected and resumes tracking based on detection result (showing in red).

◆ Decompose video into alternative 'easy' and 'hard' video clips

- For the 'easy' ones, simple but efficient tracker usually works fine
- For the 'hard' ones, complex operations (verifications) are required to deal with these cases
- Based on smooth assumption, the number of 'easy' cases may be larger than that of 'hard' cases
- Intuitively, verifications are only needed occasionally instead of for each frame (see the above figure)

Motivations

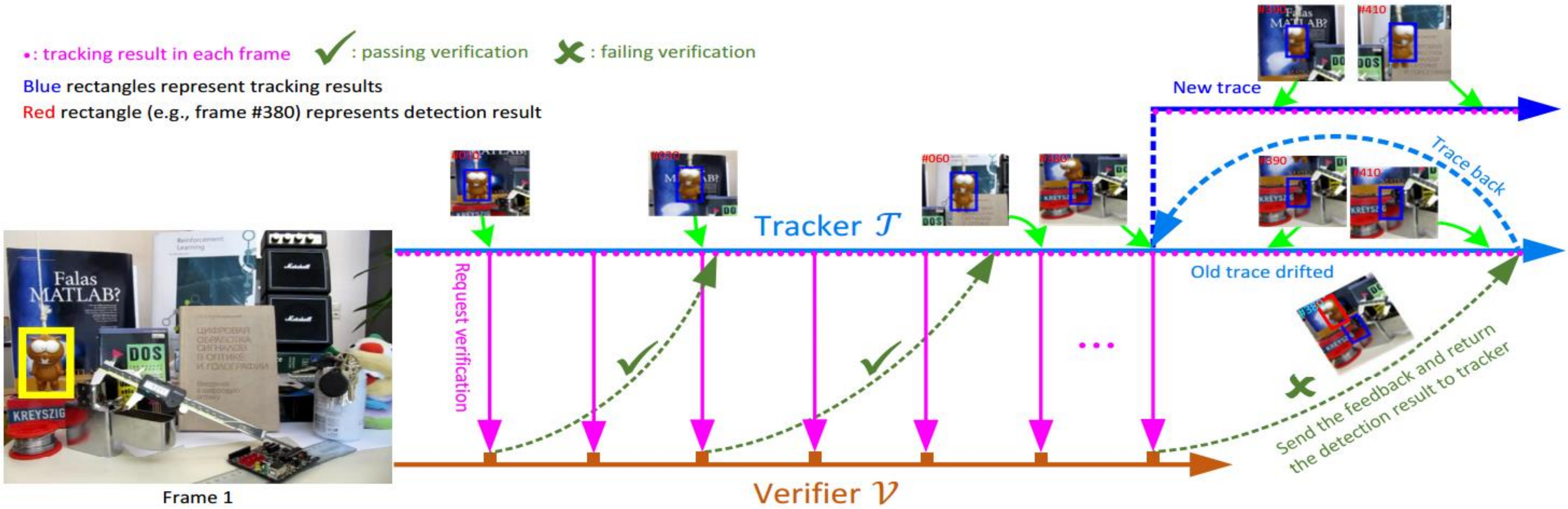
- Motivation II: asynchronism between tracker and verifier



- Tracker and verifier run in an asynchronism mechanism
  - Tracker $\mathcal{T}$: locate object in each frame
  - Verifier $\mathcal{V}$: validate tracking result every T frames, and return feedback to $\mathcal{T}$
  - $\mathcal{T}$ and $\mathcal{V}$ work independently except for necessary interactions (see the above figure)
- Visual SLAM (*simultaneous localization and mapping*)
  - PTAM (parallel tracking and mapping) splits tracking and mapping into two parallel threads
  - In PTAM, mapping is not needed for every frame; nor does verification in our task

Parallel Tracking and Verifying (PTAV)

● Framework of PTAV

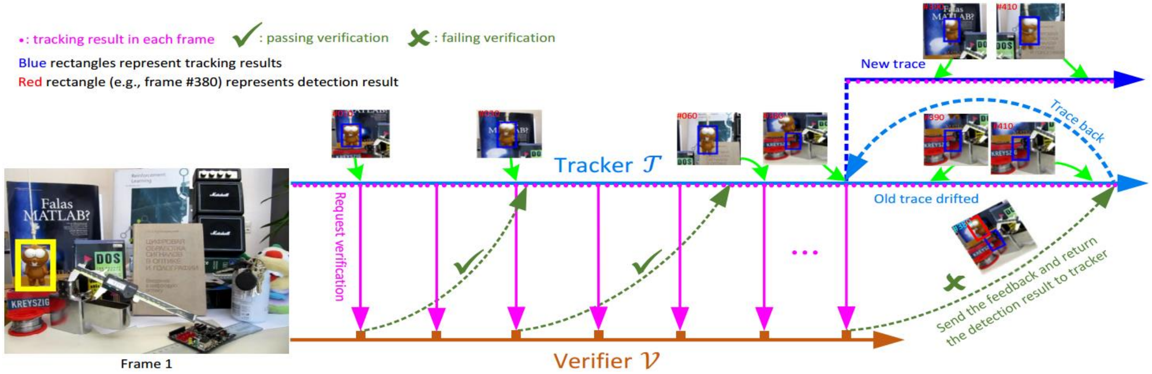


◆ Components in PTAV

- ➢ A (fast) tracker $\mathcal{T}$
- ➢ A (reliable) verifier $\mathcal{V}$

Parallel Tracking and Verifying (PTAV)

Framework of PTAV



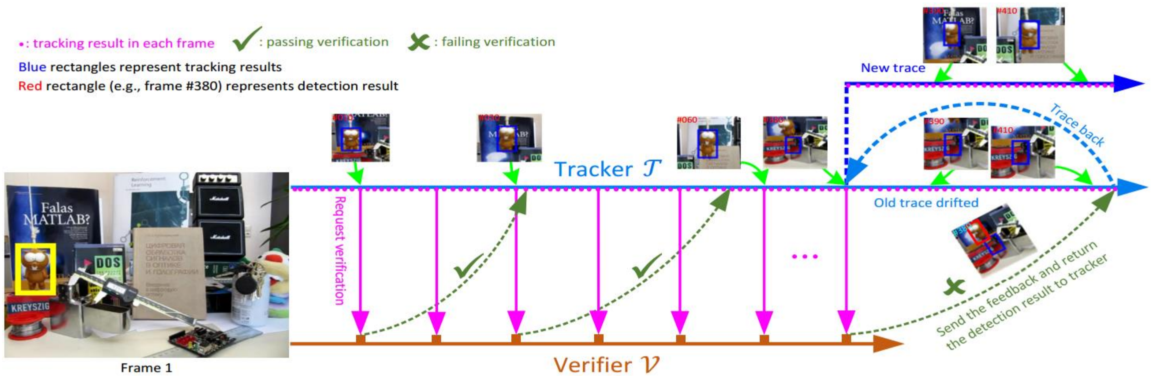
Tasks of each component

- Tracker $\mathcal{T}$
  - Perform efficiently (real-time)
  - Send verification request to $\mathcal{V}$
  - Allowed to make mistakes
  - Respond to feedback from $\mathcal{V}$ by adjusting tracking model
  - Remain all intermediate status for fast rolling back
- Verifier $\mathcal{V}$
  - Perform relatively slowly but accurately
  - Receive request from $\mathcal{T}$
  - Return feedback to $\mathcal{T}$, and correct it (if necessary)

Detailed working flow is shown in next slide.

Parallel Tracking and Verifying (PTAV)

Framework of PTAV



Workflow

Algorithm 1: Parallel Tracking and Verifying (PTAV)

Initialize the tracking thread for tracker $\mathcal{T}$;
Initialize the verifying thread for verifier $\mathcal{V}$;
Run $\mathcal{T}$ (Alg. 2) and $\mathcal{V}$ (Alg. 3) till the end of tracking;

Algorithm 3: Verifying Thread $\mathcal{V}$

while *not ended* **do**
  if *received request from* $\mathcal{T}$ **then**
    Verifying the tracking result;
    if *verification failed* **then**
      Provide correction information;
    end
    Send verification result s to $\mathcal{T}$;
  end
end

Algorithm 2: Tracking Thread $\mathcal{T}$

while *Current frame is valid* **do**
  if *received a message* s *from* $\mathcal{V}$ **then**
    if *verification passed* **then**
      Update tracking model (optional);
    else
      Correct tracking;
      Trace back and reset current frame;
      Resume tracking;
    end
  else
    Tracking on the current frame;
    if *time for verification* **then**
      Send the current result to $\mathcal{V}$ to verify;
    end
  end
  Current frame $\leftarrow$ next frame;
end

Parallel Tracking and Verifying (PTAV)

- Implementation of tracker $\mathcal{T}$
  - fDSST (TPAMI'17)
    - Tracking model

      Correlation filter h learning by optimizing

$$\epsilon(f) = \left\| \sum_{l=1}^{d} h^l * f^l - g \right\|^2 + \lambda \sum_{l=1}^{d} \| h^l \|^2 \quad (1)$$

      Using FFT, the solution for (1) is derived with

$$H^l = \frac{\bar{G} F^l}{\sum_{k=1}^{d} \bar{F}^k F^k + \lambda}, \quad l = 1, 2, \cdots, d \quad (2)$$

    - Model update

      Simple linear update for numerator A_t^l and denominator B_t^l of H_t^l

$$\begin{aligned} A_t^l &= (1-\eta) A_{t-1}^l + \eta \bar{G}_t F_t^l \\ B_t &= (1-\eta) B_{t-1} + \eta \sum_{k=1}^{d} \bar{F}_t^k F_t^k \end{aligned} \quad (3)$$

    - Tracking

      The response map y for a new image patch is computed by

$$y = \mathcal{F}^{-1} \left\{ \frac{\sum_{l=1}^{d} \bar{A}^l Z^l}{B + \lambda} \right\} \quad (4)$$

      The location of maximal value of y is the position of target object.

    - PCA is used to reduce feature dimension for efficient computation, and fDSST runs around 54 fps

For fast rolling back, in PTAV, we store all intermediate status (i.e., H_t^l in (2)) of tracker.

Parallel Tracking and Verifying (PTAV)

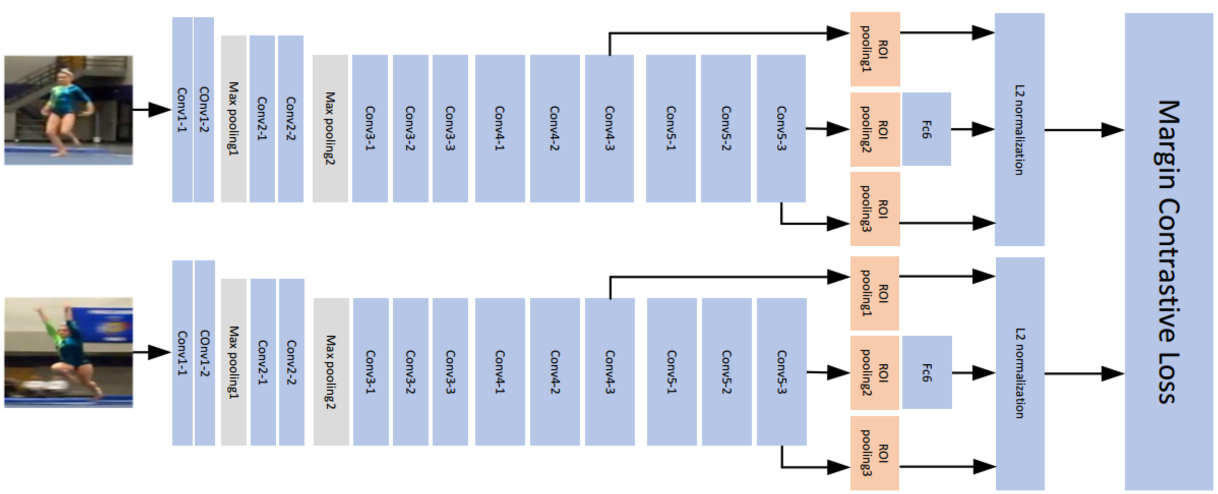
- Implementation of verifier $\mathcal{V}$
  - Siamese networks (similar to SINT (CVPR'16))
    - Formulate verification problem as a matching problem,

$$s = v(x_{obj}, x_{candidate}) \qquad (5)$$

Where s represents the verification score. Verification result is decided by

$$\begin{cases} s \geq \tau_1, & passing\ verification \\ s < \tau_1, & failing\ verification \end{cases} \qquad (6)$$

- Network architecture



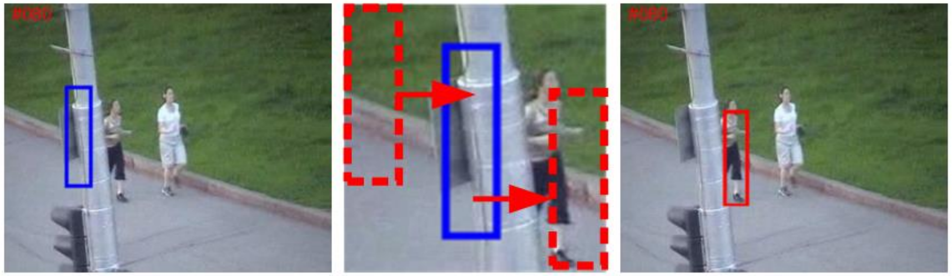
Detailed layer parameters are shown in the supplementary material.

Parallel Tracking and Verifying (PTAV)

- How to correct tracker $\mathcal{T}$
  - Detection
    - Verify tracking results every T frames using Eq. (5)
    - When finding unreliable result (i.e., failing verification using Eq. (6)), the verifier returns a feedback with correct object target position obtain via detection
    - Detection can be formulated as a (batch) verification problem as follows

$$\widehat{c} = \underset{c_i}{\operatorname{argmax}}\, \nu(x_{obj}, c_i), \quad i = 1, 2, \cdots, N \qquad (7)$$

Where $\{c_i\}_{i=1}^{N}$ denotes candidate set.



(a) (b) (c)

Parallel Tracking and Verifying (PTAV)

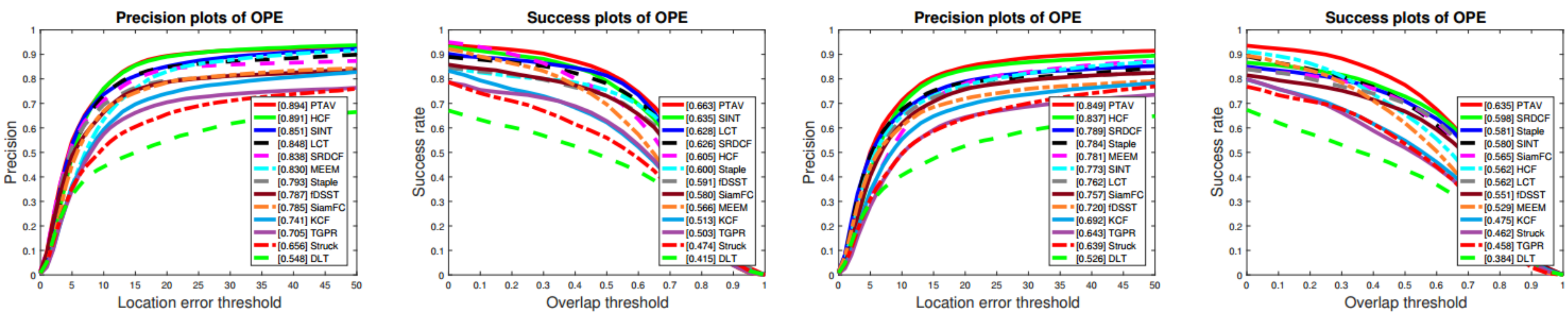
- PTAV V.S. other ensemble trackers
  - Ensemble tracking methods
    - Combine different components for tracking
      - TLD (PAMI'12)
      - LCT (CVPR'15)
      - MUSTer (CVPR'15)
    - Could be implemented in multi-threads
  - Differences with PTAV
    - Working principle
      - Other ensemble methods: different components work **simultaneously** to determine the tracking result
      - PTAV: different components work almost **independently** to determine the tracking result
    - Multi-thread implementation
      - Other ensemble methods: **synchronous** parallel multi-threads
      - PTAV: **asynchronous** parallel multi-threads

Parallel Tracking and Verifying (PTAV)

- Experimental Setting
  - Datasets: OTB2013, OTB2015, TC128, and UAV20L
    - Yi Wu et al, Online Object Tracking: A Benchmark, in CVPR, 2013
    - Yi Wu et al, Object Tracking Benchmark, TPAMI, 2015.
    - Pengpeng Liang et al, Encoding Color Information for Visual Tracking: Algorithms and Benchmark, TIP, 2015.
    - Matthias Mueller et al, A Benchmark and Simulator for UAV Track, in ECCV, 2016.
  - Metrics
    - Distance precision (DP) rate
    - Overlap success (OS) rate
  - Validation scheme
    - OPE: one-pass evaluation

Parallel Tracking and Verifying (PTAV)

● Overall Results on OTB2013 and OTB2015



(a) Comparisons on OTB2013 (b) Comparisons on OTB2015

			Deep trackers				Correlation-filters based trackers					Representative trackers		
		PTAV (Ours)	HCF [28]	SINT [37]	DLT [38]	SiamFC [4]	SRDCF [8]	Staple [3]	LCT [29]	fDSST [7]	KCF [16]	MEEM [42]	TGPR [11]	Struck [14]
OTB2013	DPR (%)	89.4	89.1	85.1	54.8	78.5	83.8	79.3	84.8	78.7	74.1	83	70.5	65.6
	OSR (%)	82.7	74	79.1	47.8	74.0	78.2	75.4	81.2	74.7	62.2	69.6	62.8	55.9
	Speed (*fps*)	27	11	3	9	46	4	45	27	54	245	21	1	10
OTB2015	DPR (%)	84.9	83.7	77.3	52.6	75.7	78.9	78.4	76.2	72	69.2	78.1	64.3	63.9
	OSR (%)	77.6	65.6	70.3	43	70.9	72.9	70.9	70.1	67.6	54.8	62.2	53.5	51.6
	Speed (*fps*)	25	10	2	8	43	4	43	25	51	243	21	1	10

Among all real-time trackers, PTAV achieves the best performance!

PTAV performs better than deep learning based HCF tracker, in both accuracy and efficiency!

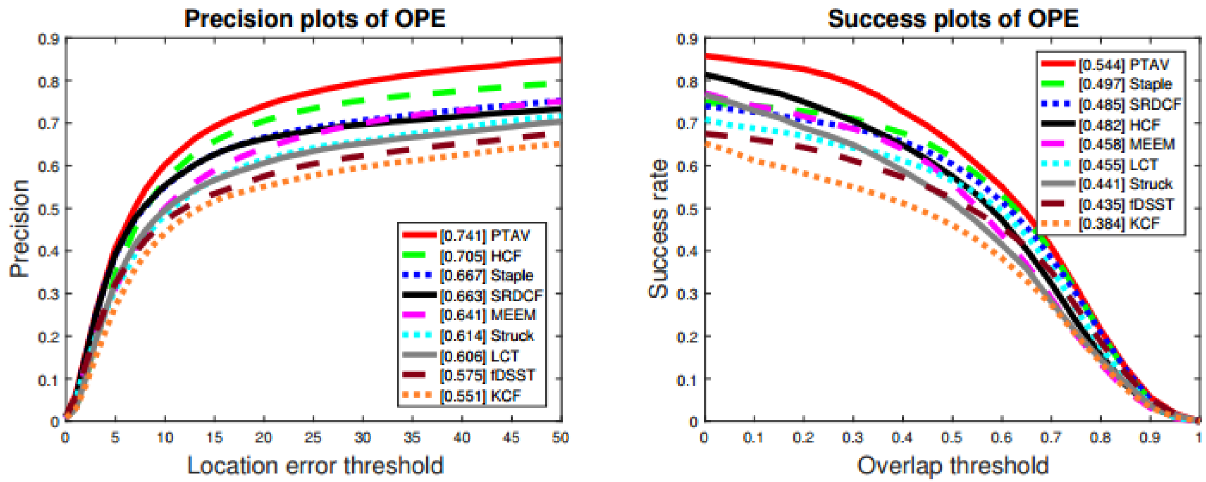
Parallel Tracking and Verifying (PTAV)

- Attribute evaluation on OTB2015

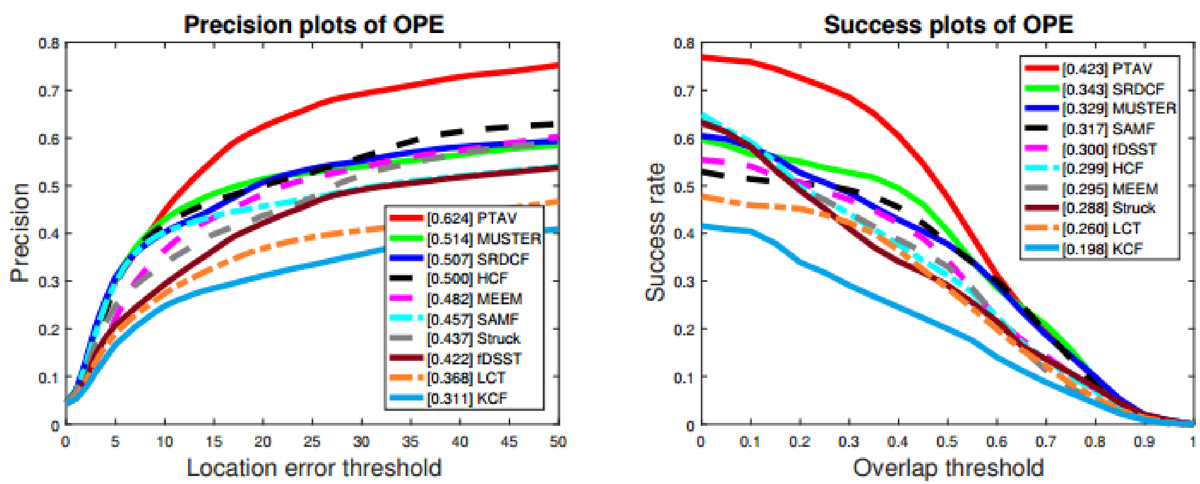
	Distance precision rate (%) on eleven attributes						Overlap success rate (%) on eleven attributes					
Attribute	PTAV	HCF [28]	SRDCF [8]	Staple [3]	MEEM [42]	SINT [37]	PTAV	HCF [28]	SRDCF [8]	Staple [3]	MEEM [42]	SINT [37]
BC	87.9	84.7	77.6	77.0	75.1	75.1	64.9	58.7	58.0	57.4	52.1	56.7
DEF	81.3	79.1	73.4	74.8	75.4	75.0	59.7	53.0	54.4	55.4	48.9	55.5
FM	77.7	79.7	76.8	70.3	73.4	72.5	60.8	55.5	59.9	54.1	52.8	55.7
IPR	83.0	85.4	74.5	77.0	79.3	81.1	60.7	55.9	54.4	55.2	52.8	58.5
IV	86.0	81.7	79.2	79.1	74.0	80.9	64.3	54.0	61.3	59.8	51.7	61.8
LR	78.9	78.7	63.1	60.9	60.5	78.8	56.3	42.4	48.0	41.1	35.5	53.9
MB	81.0	79.7	78.2	72.6	72.1	72.8	62.9	57.3	61.0	55.8	54.3	57.4
OCC	83.2	76.7	73.5	72.6	74.1	73.1	62.3	52.5	55.9	54.8	50.3	55.8
OPR	82.8	81.0	74.6	74.2	79.8	79.4	61.1	53.7	55.3	53.8	52.8	58.6
OV	73.6	67.7	59.7	66.1	68.3	72.5	57.0	47.4	46.0	48.1	48.4	55.9
SV	79.7	80.2	74.9	73.1	74.0	74.2	59.0	48.8	56.5	52.9	47.3	55.8
Overall	84.9	83.7	78.9	78.4	78.1	77.3	63.5	56.2	59.8	58.1	52.9	58.0

Parallel Tracking and Verifying (PTAV)

- Overall Results on TC128 and UAV20L



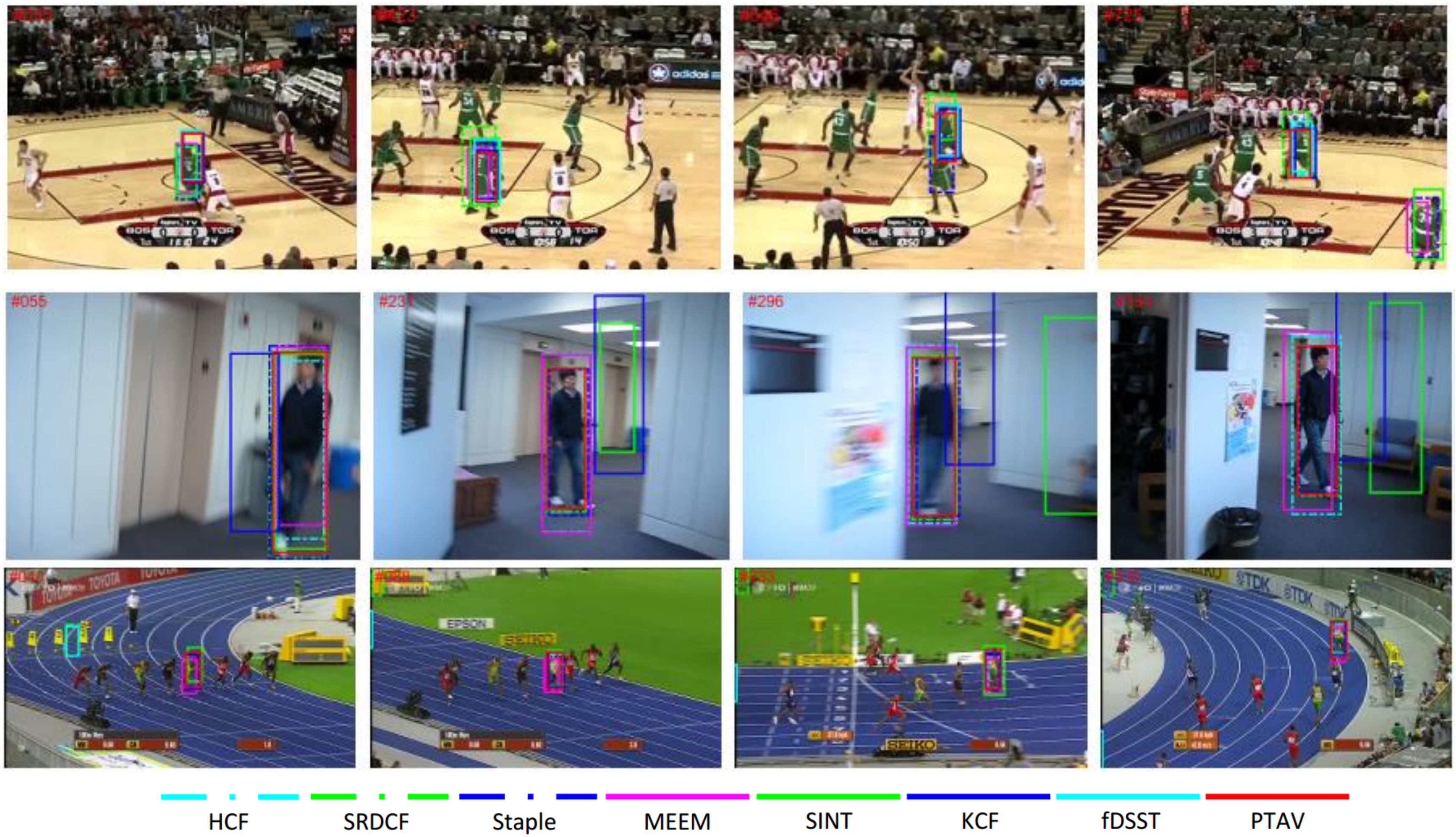
(a) Comparisons on TC128



(b) Comparisons on UAV20L

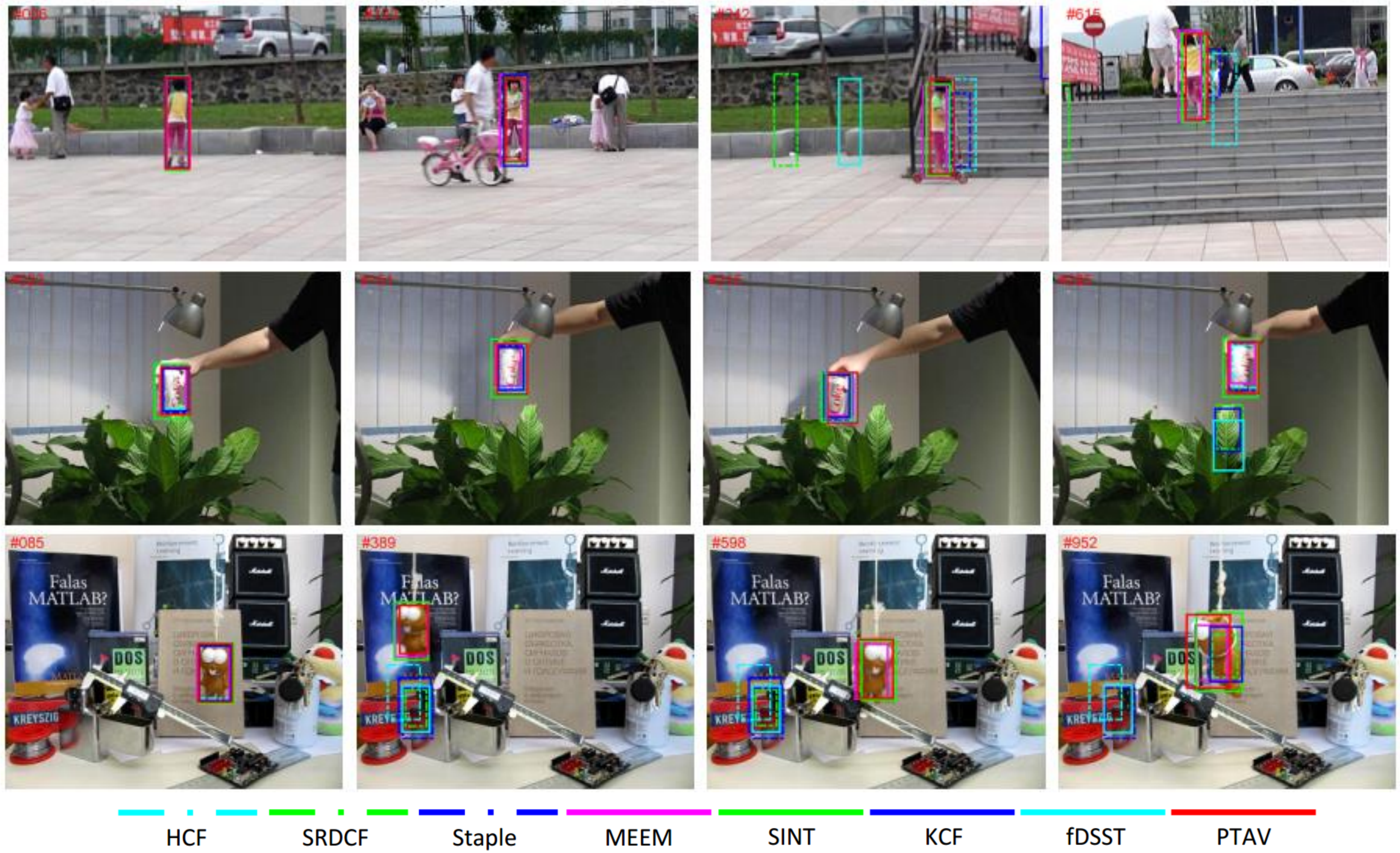
Parallel Tracking and Verifying (PTAV)

- Qualitative Results I



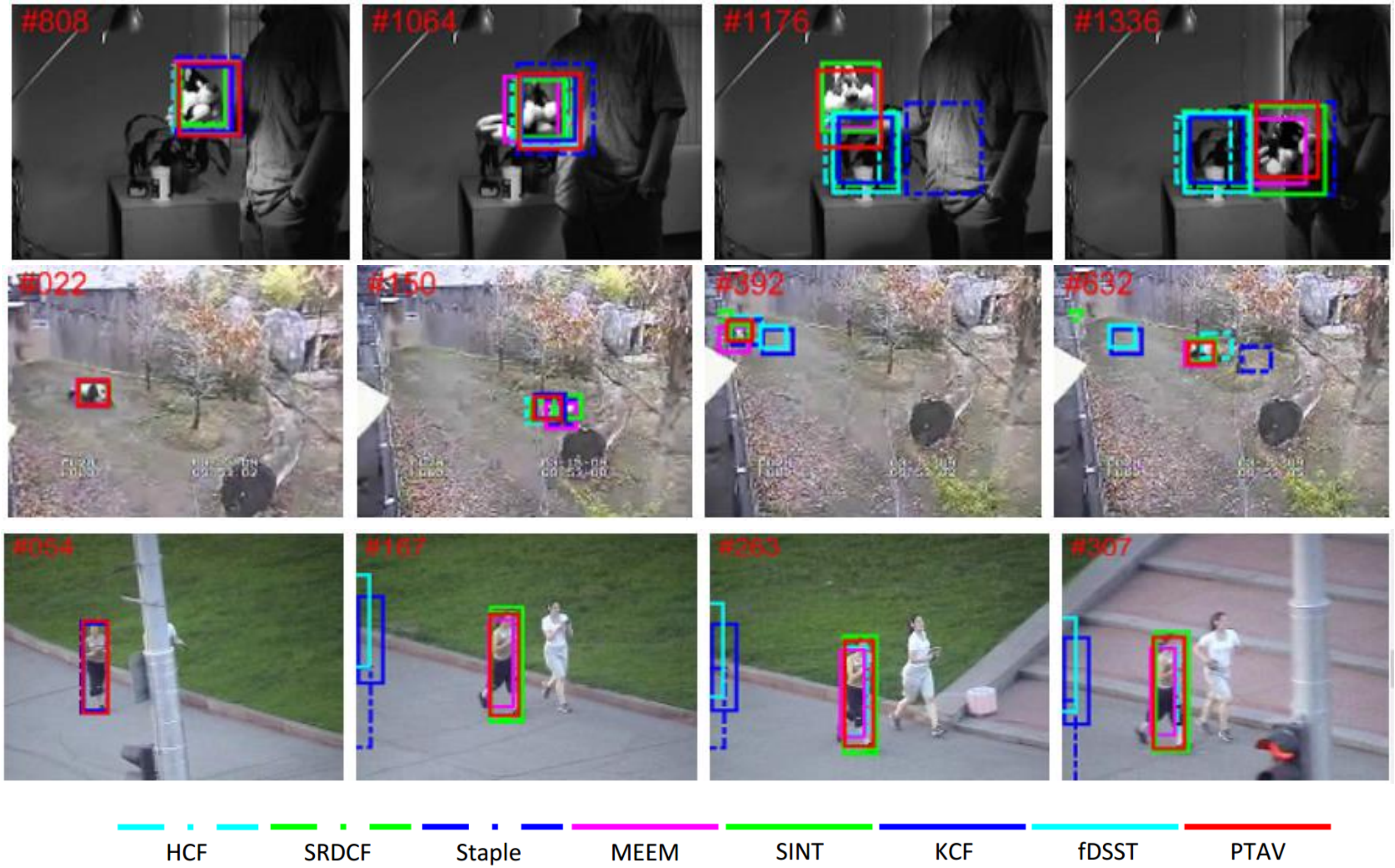
Parallel Tracking and Verifying (PTAV)

- Qualitative Results II



Parallel Tracking and Verifying (PTAV)

- Qualitative Results III



Parallel Tracking and Verifying (PTAV)

- Detailed analysis of PTAV
  - Different verification interval V

	$V = 5$	$V = 10$	$V = 15$
DPR (%)	89.7	89.4	87.9
Speed (*fps*)	23	27	29

  - Two threads V.S. one

Threads	OTB2013 [39]	OTB2015 [40]	TC128 [26]	UAV20L [31]
One	16	14	11	15
Two	27	25	21	25

  - Different tracker $\mathcal{T}$: fDSST V.S. KCF

		PTAV with fDSST	PTAV with KCF
OTB2013	DP (%)	89.4	80.4
	OS (%)	82.7	66.3
	Speed (*fps*)	27	24
OTB2015	DP (%)	84.9	73.5
	OS (%)	77.6	57.9
	Speed (*fps*)	25	21

Parallel Tracking and Verifying (PTAV)

- Conclusions

  - Decompose tracking into two separate tasks (i.e., fast tracking and slow verifying)

  - The (fast) tracking and (slow) verifying work asynchronously

  - PTAV enjoys both high efficiency provided by $\mathcal{T}$ and the strong discriminative power provided by $\mathcal{V}$

  - PTAV is a very flexible framework with great rooms for improvement and generalization.

Project & code: http://www.dabi.temple.edu/~hbling/code/PTAV/ptav.htm

Parallel Tracking and Verifying (PTAV)

Thank you!