Comp-LOP: Complex Form of Local Orientation Plane for Object Tracking

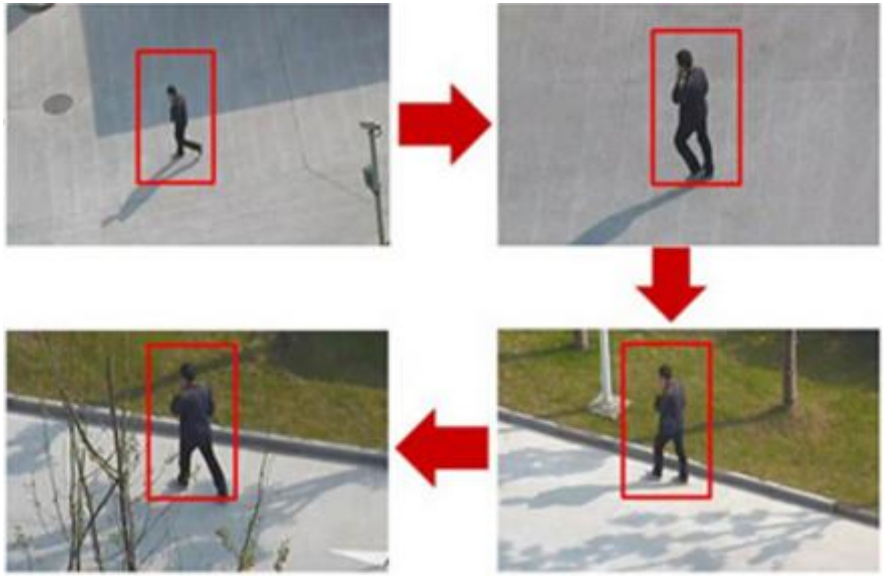
Miaobin Cen and Cheolkon Jung
School of Electronic Engineering
Xidian University, China

Object Tracking

- **Object tracking**: Process of locating a moving object (or multiple objects) over time in video, the ground-truth object is given in the first frame.

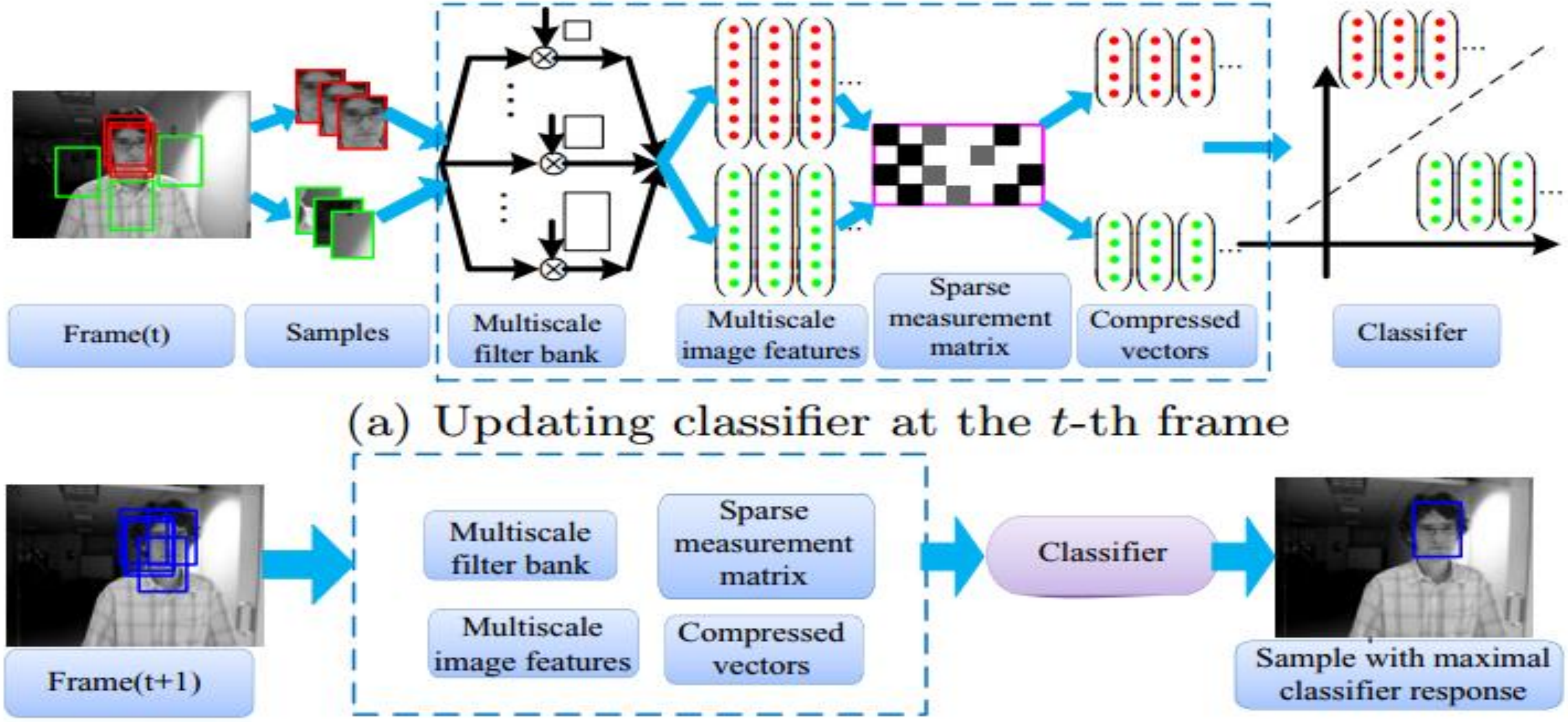
- **Challenges**: Occlusion, illumination changes, and background clutter

- **Applications**: Traffic monitoring, video surveillance



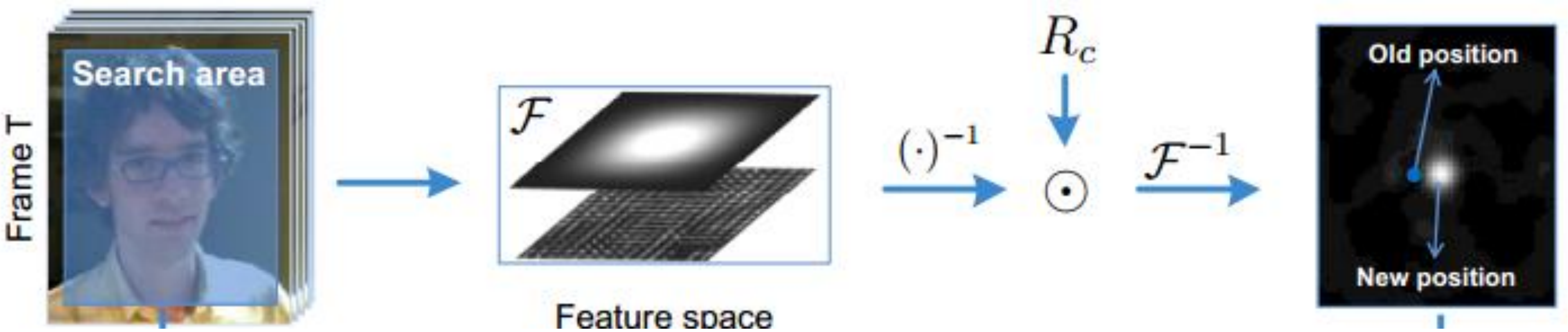
Fast Compressive Tracking (TPAMI 2014)

- **Basic idea**: Dividing the region into target and background, then extract features of samples in the target and background. Finally, use the Bayesian classifier to find the target in the new frame.
- HAAR-like feature + Bayesian classifier



Kernelized Correlation Filters (KCF, TPAMI 2015)

- **Basic idea**: Extract HOG and perform regression using Gaussian distribution response, finally use **correlation filters** and find **the position with the maximum response** as the target.



Correlation Filters: Circulant Matrix

$$C(\mathbf{u}) = \begin{bmatrix} u_0 & u_1 & u_2 & \cdots & u_{n-1} \\ u_{n-1} & u_0 & u_1 & \cdots & u_{n-2} \\ u_{n-2} & u_{n-1} & u_0 & \cdots & u_{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u_1 & u_2 & u_3 & \cdots & u_0 \end{bmatrix}$$

$$C(\mathbf{u})\mathbf{v} = \mathcal{F}^{-1}\left(\mathcal{F}^*(\mathbf{u}) \odot \mathcal{F}(\mathbf{v})\right)$$

An nxn circulant matrix $C(\mathbf{u})$ is obtained from the $nx1$ vector $\mathbf{u}$ by concatenating all possible cyclic shifts of $\mathbf{u}$

$C(\mathbf{u})\mathbf{v}$ represents convolution of vectors **u** and **v**. $\odot$ is the element-wise product, F and F^{-1} denote the Fourier transform and its inverse, $*$ is the complex-conjugate.

+30 +15 Base sample -15 -30

Examples of vertical cyclic shifts of a base sample.

Correlation filters—How to get α

$$\min_{\mathbf{w}} \sum_i (f(\mathbf{x}_i) - y_i)^2 + \lambda \|\mathbf{w}\|^2 \longrightarrow \boldsymbol{\alpha} = (K + \lambda I)^{-1} \mathbf{y} \longrightarrow f(\mathbf{x}_i) = \sum_i \alpha_i \kappa(\mathbf{x}_i, \mathbf{z})$$

where K is the kernel matrix with elements $Kij = \kappa(\mathbf{x}i, \mathbf{x}j)$, I is the identity matrix, and the vector $\mathbf{y}$ has elements y_i.

Depending on the formula: $$C(\mathbf{u})\mathbf{v} = \mathcal{F}^{-1}\left(\mathcal{F}^*(\mathbf{u}) \odot \mathcal{F}(\mathbf{v})\right)$$

Then, we get:

$$\boldsymbol{\alpha} = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(\mathbf{y})}{\mathcal{F}(\mathbf{k}) + \lambda}\right)$$

$$f(\mathbf{x}_i) = \sum_i \alpha_i \kappa(\mathbf{x}_i, \mathbf{z})$$

The x and x' is two samples

$$\mathbf{k}^{\text{gauss}} = \exp\left(-\frac{1}{\sigma^2}\left(\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2 - 2\mathcal{F}^{-1}\left(\mathcal{F}(\mathbf{x}) \odot \mathcal{F}^*(\mathbf{x}')\right)\right)\right)$$

Correlation Filters: Pre-processing

Since Fourier transform is periodic, it does not consider the **image boundaries**. The large discontinuity between opposite edges of a non-periodic image results in a noisy Fourier representation. Thus, it uses pre-processing as follows:

$$x_{ij} = \left(x_{ij}^{\text{raw}} - 0.5\right)\sin(\pi i/n)\sin(\pi j/n), \quad \forall i,j = 0,\ldots,n-1$$

Moreover, the output will be 1 near the target location (i_0, j_0), and decay to 0 as the distance increases, with a bandwidth of σ

$$y_{ij} = \exp\left(-\left((i-i')^2 + (j-j')^2\right)/\sigma^2\right), \quad \forall i,j = 0,\ldots,n-1$$

Problem Formulation

Correlation filters for object tracking in KCF framework:

1. Fixed search range ⇒ **Adaptive search range** based on entropy

   Find a good search range with entropy that can make tracker stronger.

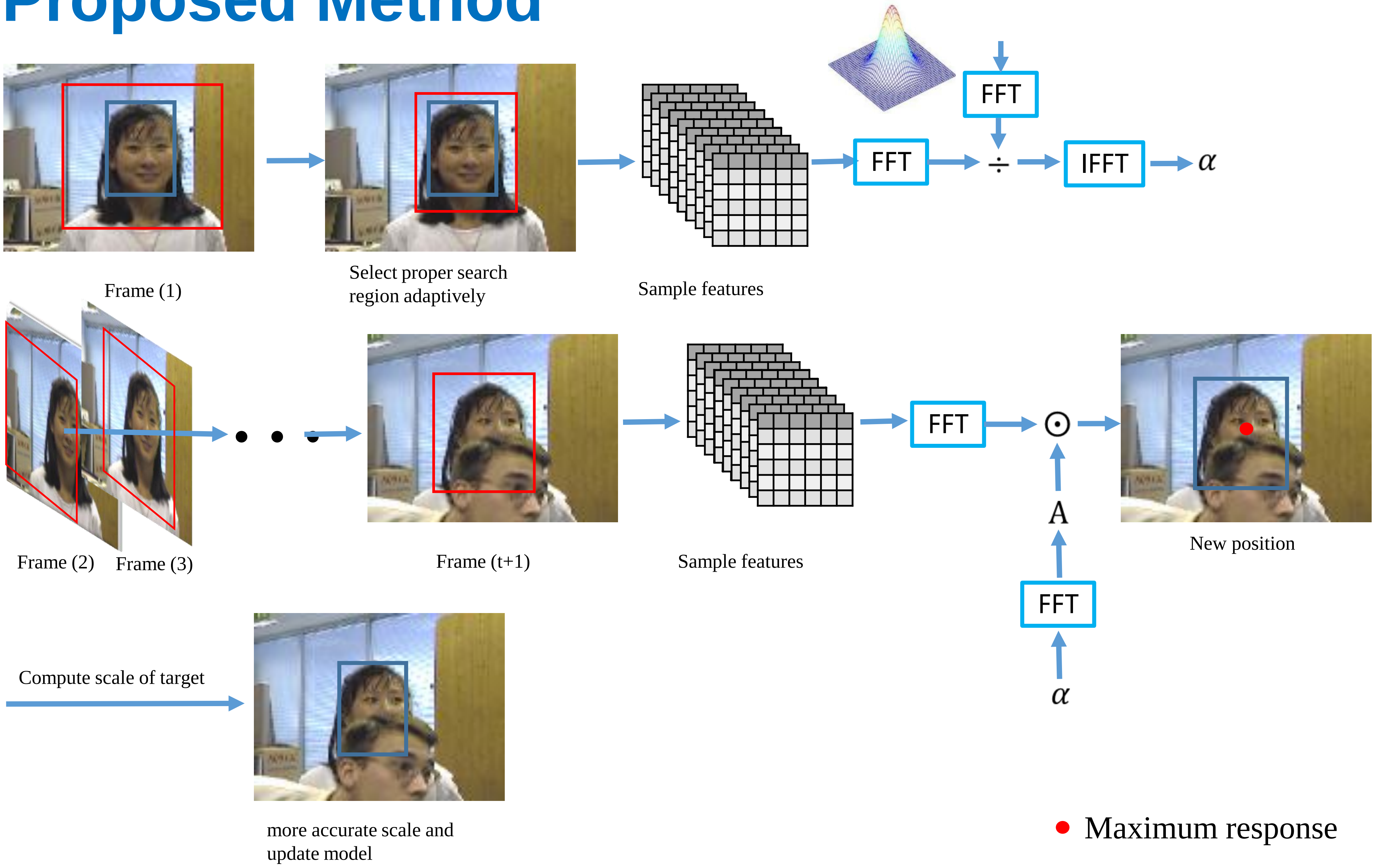
2. Features (HOG, Haar-like, LBP, …) ⇒ **Comp-LOP**

   Previous features are only suitable for specific objects.

   Comp-LOP considers the relationship between pixels, thus more general

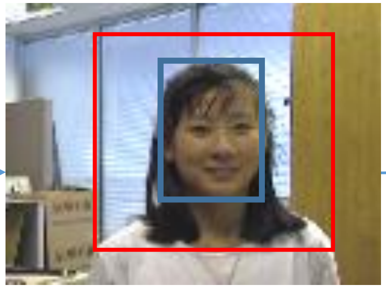
3. Fixed scale ⇒ **Scale invariant** using adaptive sigma

Proposed Method



Adaptive Search Range based on Entropy

Make the search region as much as possible, e.g. search region s is 3 times larger than target t.



Original search region

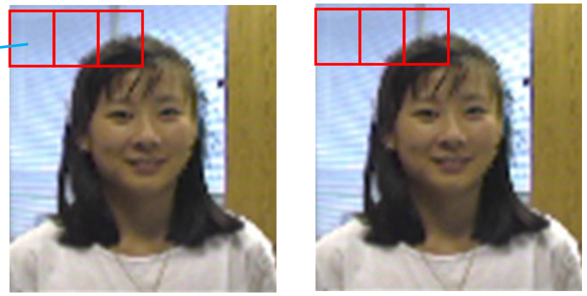
In the first frame, compute entropy of s and t as follows:

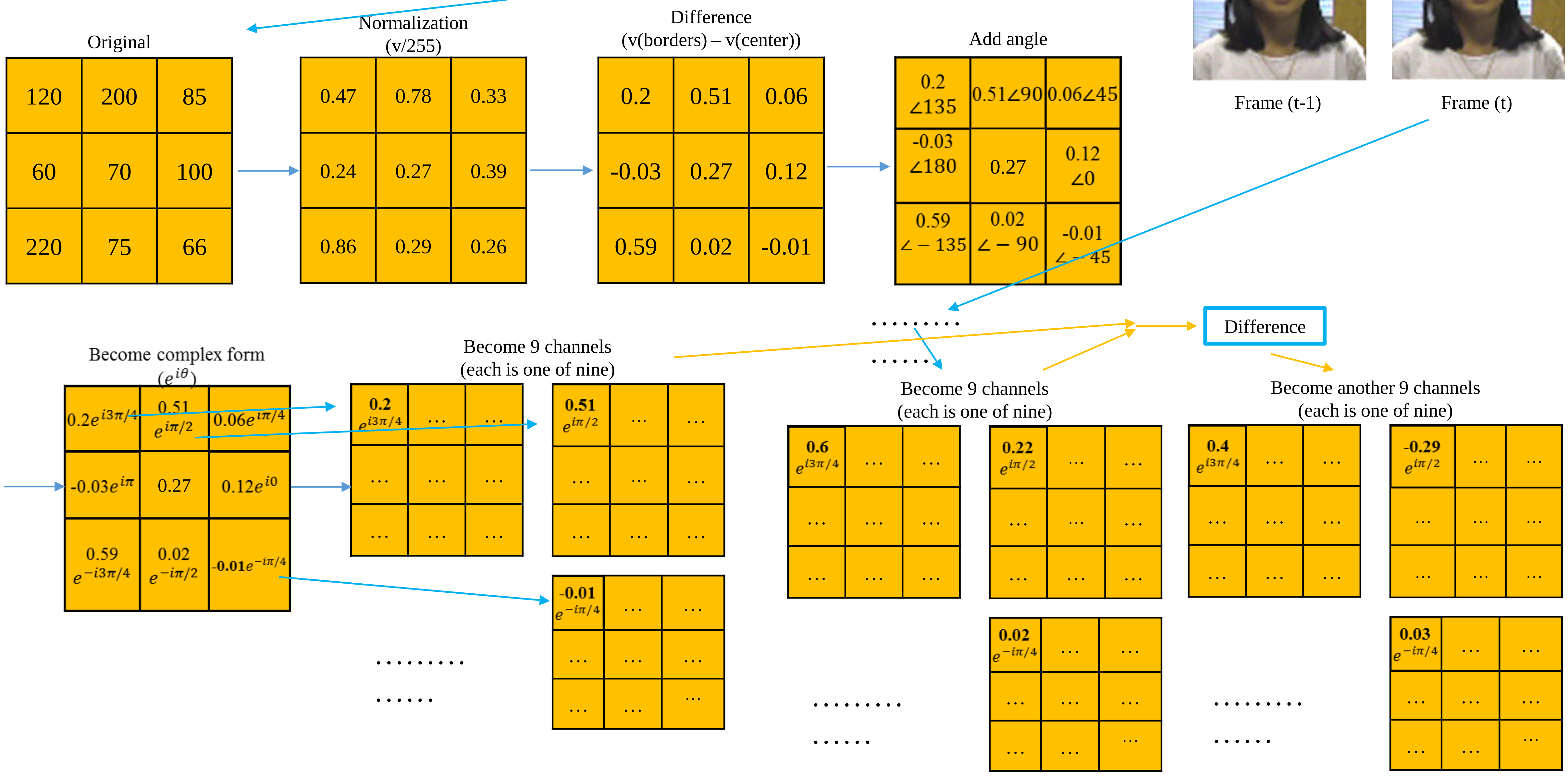
Search region

Repeat k times
$E(t) = \text{entropy}(t)$
$E(s) = \text{entropy}(s)$
$\text{Ratio}(k) = E(t) / E(s)$
$s = s - 0.2$
Until $s = 0$
$M = \text{average}(\text{Ratio})$
$U = \text{abs}(M - \text{Ratio})$
Find k that $U(k)$ is minimum
$s = 3 - (k - 1) * 0.2$

Bigger ratio means more information of target and less information of background, and vice versa, thus finding a **good balance**.

Comp-LOP (Complex form-local orientation plane)





Scale-Invariant Model Update

The scale of the target often changes over time. Therefore, the scale parameter σ in k and y should be updated accordingly. I propose the scale update scheme as follows:

$$\theta_t' = \left(\frac{y(t) + s_t}{y(t-1) + s_{t-1}}\right)^{0.5}$$

$$\bar{\theta}_t = \frac{1}{n}\sum_{i=1}^{n} \theta_{t-i}'$$

$$\theta_{t+1} = (1 - \lambda)\theta_t + \lambda\bar{\theta}_t$$

$$\sigma_{t+1} = \sigma_t \theta_t$$

$$\mathbf{k}^{\text{gauss}} = \exp\left(-\frac{1}{\sigma^2}\left(\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2 - 2\mathcal{F}^{-1}\left(\mathcal{F}(\mathbf{x}) \odot \mathcal{F}^*(\mathbf{x}')\right)\right)\right)$$

$$y_{ij} = \exp\left(-\left((i-i')^2 + (j-j')^2\right)/\sigma^2\right)$$

Frame 1-th

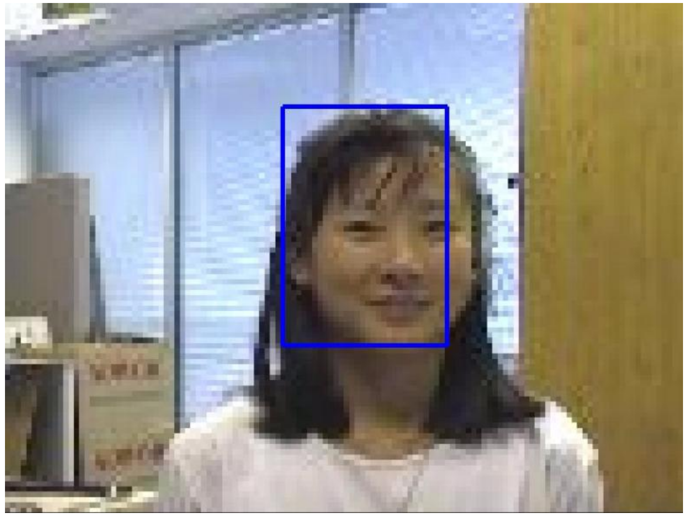
Frame 416-th

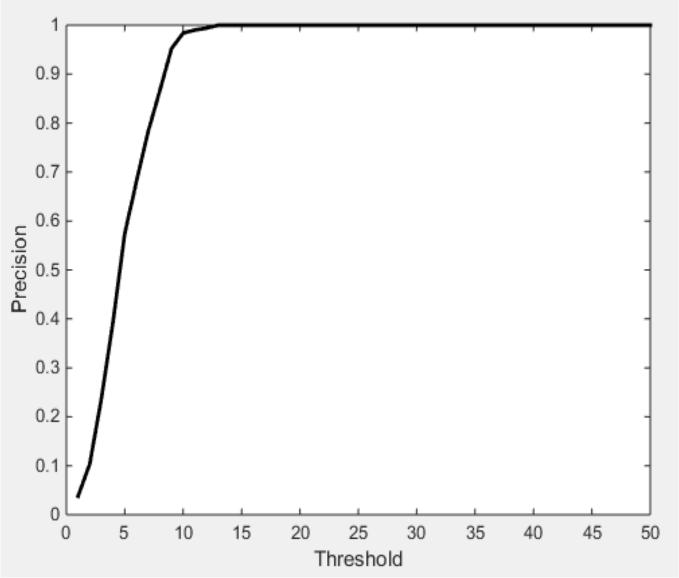
where $y(\cdot)$ is the response at the t-th frame, and θ_t' is the estimated scale between two consecutive frames. To avoid oversensitive and to reduce noise, the estimated target scale θ_{t+1} is obtained that cannot change too fast in which $\bar{\theta}_t$ is the average of the estimated scales from n consecutive frames, and $\lambda > 0$ is a fixed parameter.

Experimental Results

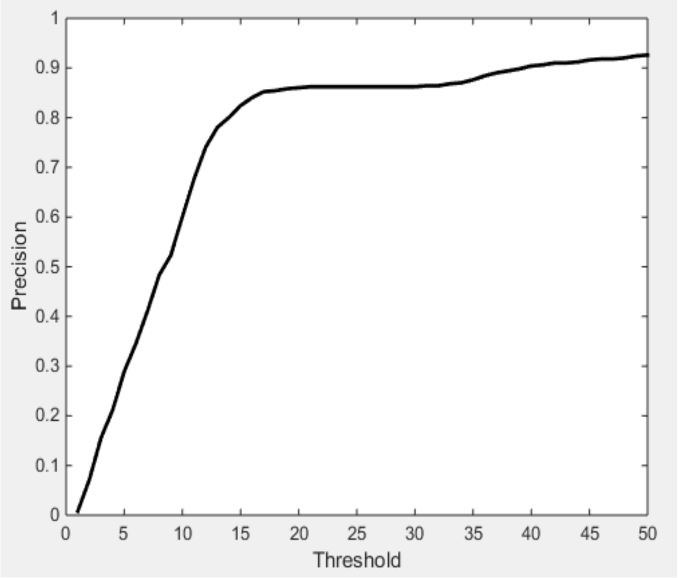
- Hardware: PC with Intel Pentium CPU G3260 3.30GHz and 4 GB RAM
- Software: Windows 7 and Matlab 2013
- Database: OTB 50
- Evaluation measures: DP (Distance Precision) OS (Overlap success rate), runtime (frame/sec)
- Compared methods: CSK, STC, TLD, Struck, SCM , CT, KCF, LOT, ORIA, MTT, ASLA.

Experimental Results: Visual Comparison

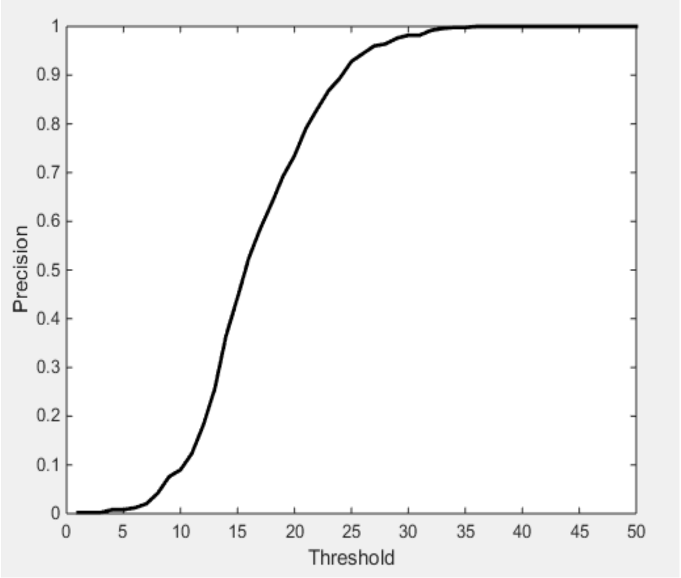




Proposed

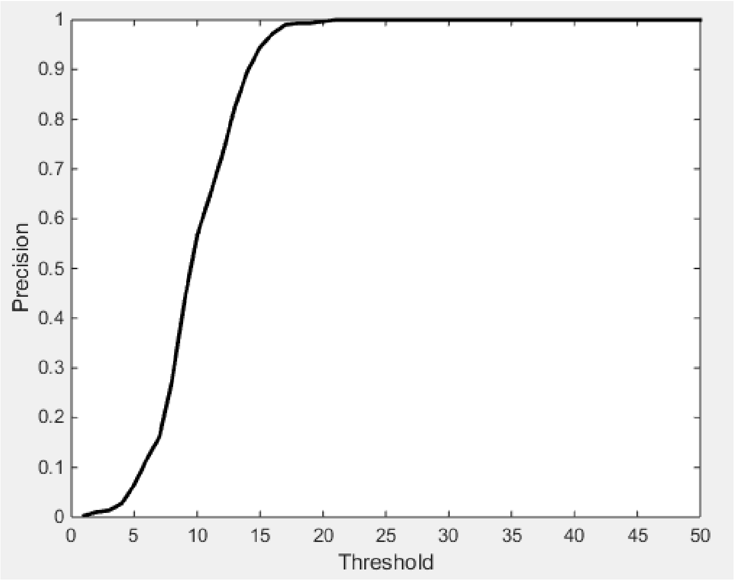


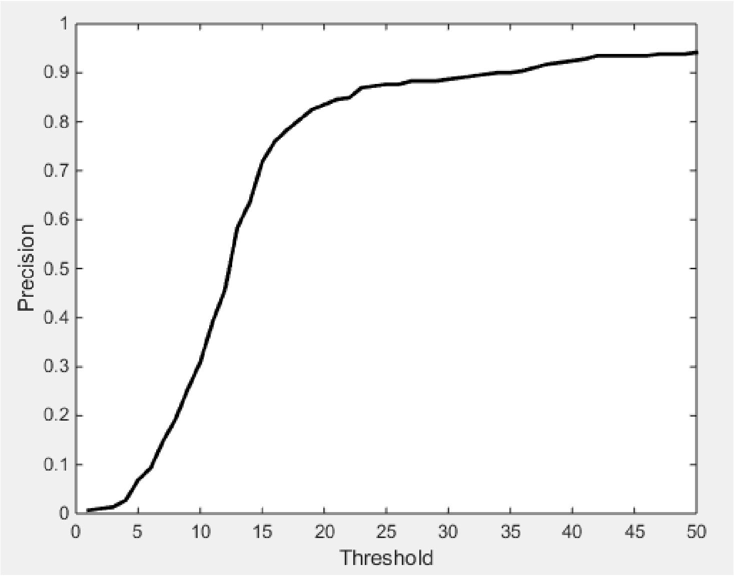
KCF

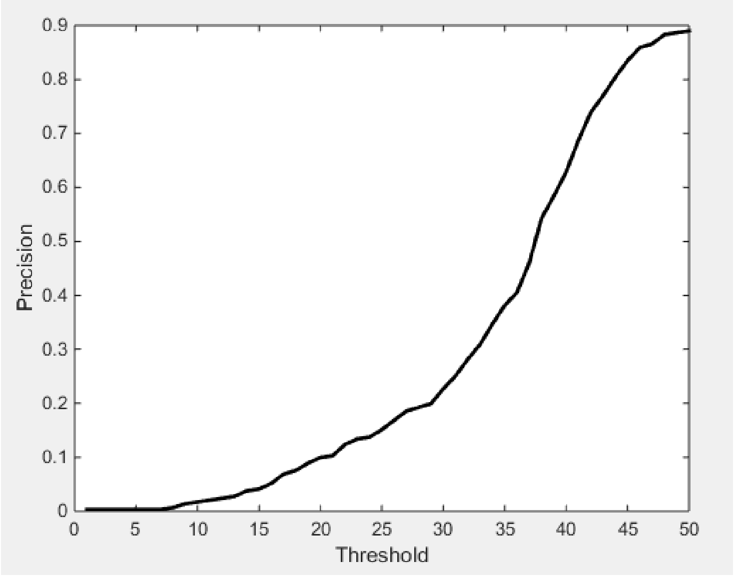


CT

Experimental Results: Visual Comparison





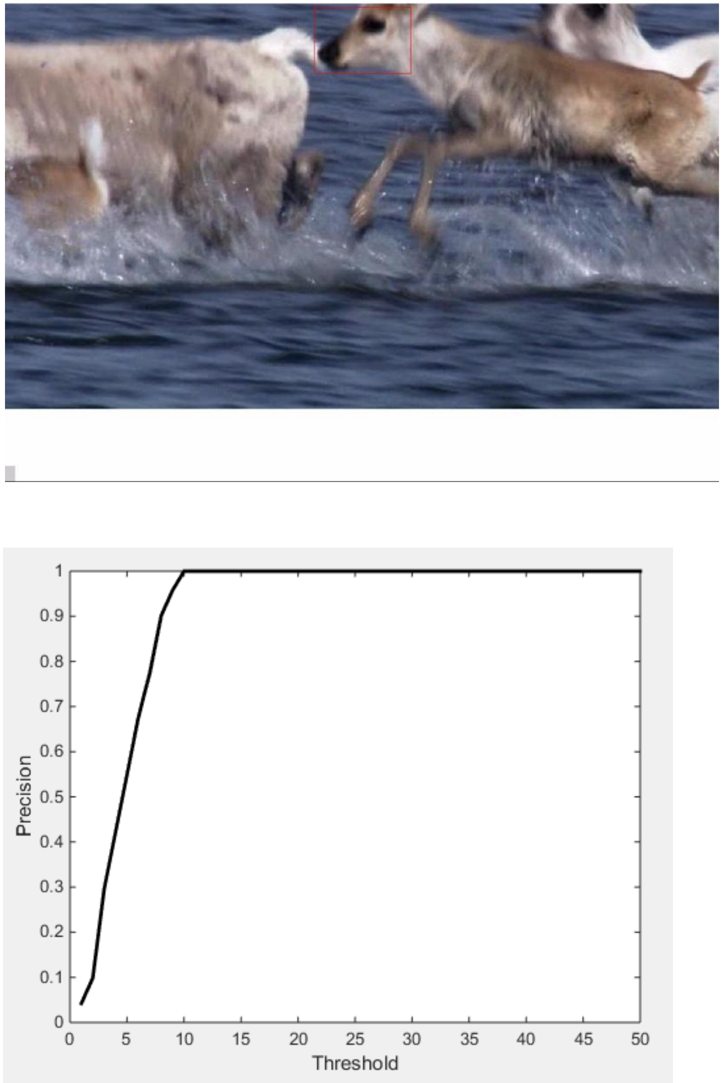


Proposed

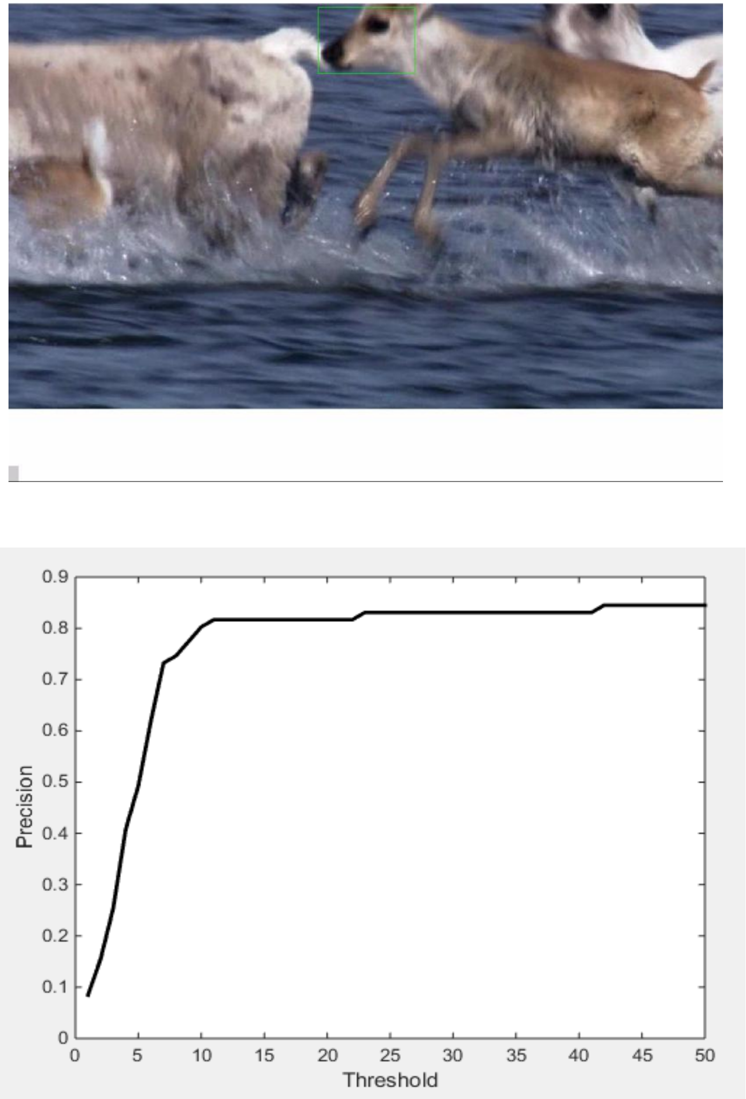
KCF

CT

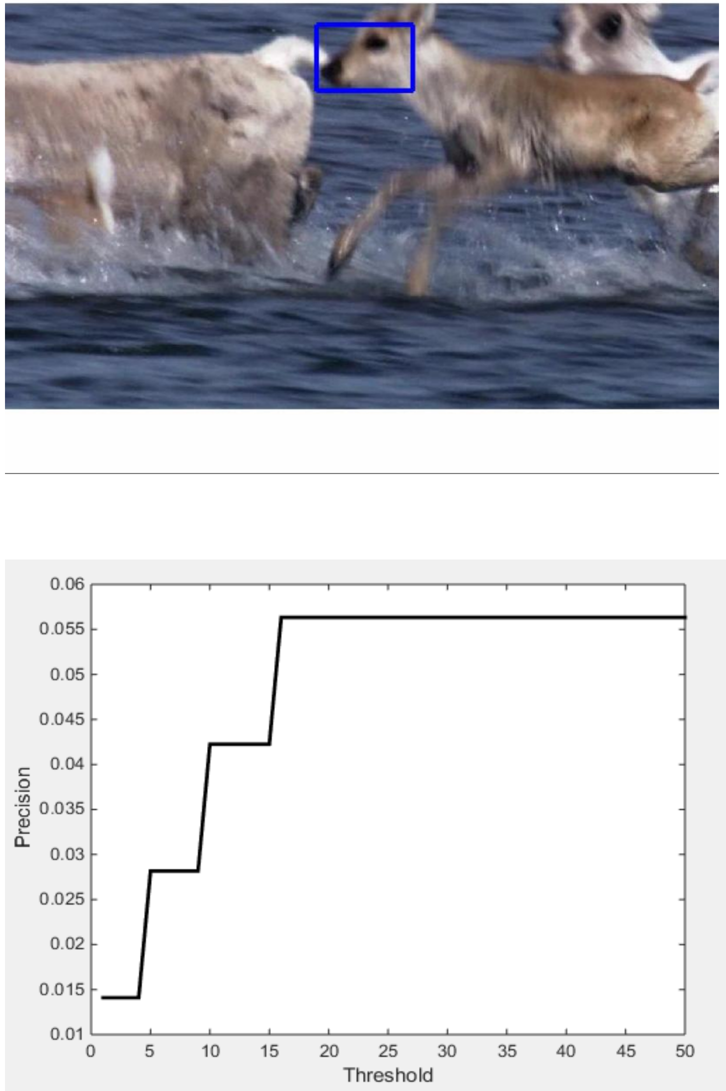
Experimental Results: Visual Comparison



Proposed



KCF



CT

Experimental Results: DP, OS and Speed

Table 1. Performance comparison between the proposed method and other 11 state-of-the-art trackers

Metrics	Proposed	CSK	Struck	MTT	CT	KCF	STC	ORIA	TLD	ASLA	LOT	SCM
DP(%)	**69.5**	58.6	<u>66.4</u>	53.2	38.0	*67.8*	57.6	49.5	57.2	55.9	46.6	61.3
OS(%)	<u>51.5</u>	46.6	*53.0*	44.9	31.3	46.9	40.4	38.9	45.1	48.7	46.6	**56.5**
Speed(fps)	67.6	*150*	12.2	2.1	36.5	<u>86.3</u>	**286**	8.2	21.5	6.4	0.6	0.82

The bold numbers indicate the best performance, the italic ones indicate the second performance, and the underline ones indicate the third performance.

Table 2. Performance comparison of the proposed method and other 11 state-of-the-art trackers under occlusion

Metrics	Proposed	CSK	Struck	MTT	CT	KCF	STC	ORIA	TLD	ASLA	LOT	SCM
DP(%)	**70.0**	56.4	<u>63.9</u>	51.2	45.1	58.4	53.1	46.1	56.8	49.0	50.2	*67.0*
OS(%)	**55.4**	45.0	*53.3*	43.7	38.5	45.8	37.7	36.8	44.3	43.0	40.7	<u>52.7</u>

The bold numbers indicate the best performance, the italic ones indicate the second performance, and the underline ones indicate the third performance.

Conclusions

- We have proposed **Comp-LOP for object tracking**.

- We have utilized **entropy** to compute a appropriate search region.
- We have introduced **complex form** to get a novel and simple feature for object tracking.
- We have provided a **scale update** scheme for target scale-invariant tracking.

- Experimental results show that the proposed method outperforms state-of-the-art trackers on large benchmark data sets (**DP: 69.5%, OS: 51.5%**). Its processing speed is **67.6fps**, i.e. **real-time**.

THANK YOU!