BİL-722 ADVANCED TOPICS IN COMPUTER VISION

Çağdaş Baş, N10266943

Paper: Robust Object Tracking with Online Multi-lifespan Dictionary Learning

Authors: Junliang Xing, Jin Gao, Bing Li, Weiming Hu and Shuicheng Yan

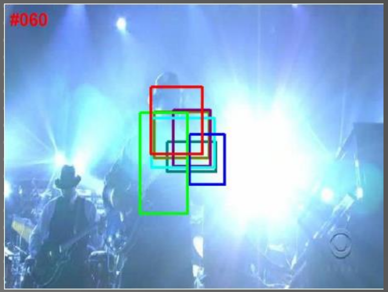
TRACKING

Images from: SUN Dataset

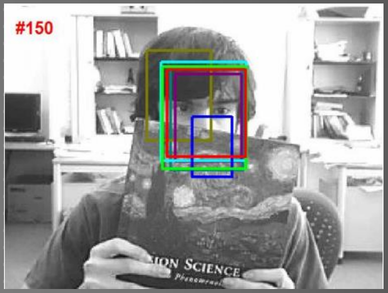
WHY IS TRACKING A DIFFICULT PROBLEM?

- Image noise and background clutter

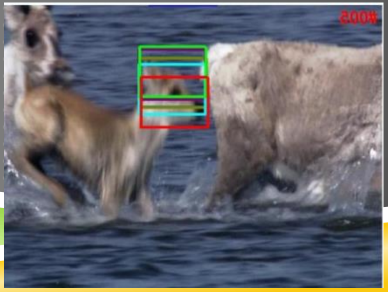
- Illumination changes



- Clutter



- Motion



RELATED WORK

- There are three types of methods in general:
  1. Generative Methods

     Eigentracker, meanshift tracker etc.

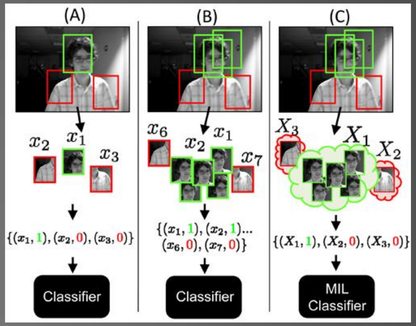
  2. Discriminative Methods

     Ensemble tracker, MIL tracker etc.

  3. Sparse Learning Methods

     This method

MeanShift Tracker [1]



MIL Tracker [2]

DIFFERENT STAGES OF TRACKING

1. Designing good templates
2. Solving optimization problem efficiently
3. Updating the object template.

Many of the papers focus mainly on these two parts.

Present methods use fixed templates or incremental template update.

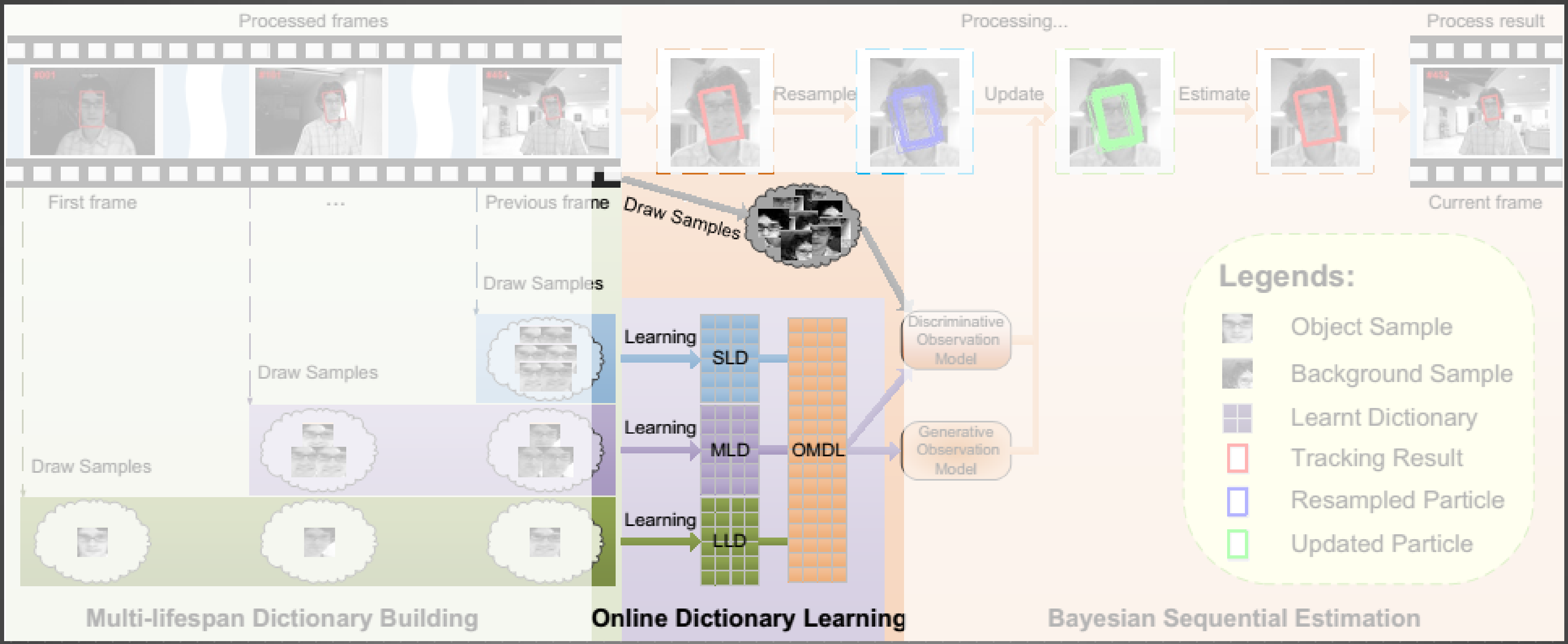
WHAT IS SPARSE LEARNING?

- Represents an instance with a minimum set of dictionary elements.

$$\min_{c} \|Tc - y\|_2^2 + \lambda |c|_1$$

- Find the sparse representation c of y in dictionary T

OVERALL APPROACH



TRACKING AS ONLINE DICTIONARY LEARNING

- Extract candidate regions $\mathcal{Y}$
- Optimize a new object template (dictionary) by minimizing:

$$D^* = \underset{c}{\operatorname{argmin}} \frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} l(y, D)$$

  - Template set is not predifined but learned in time.

ONLINE LEARNING ALGORITHM

Algorithm 1 Online dictionary learning for template update

Input: frame data $\mathbf{I}_t$, tracking results $\mathbf{x}_t$, learned dictionary $\mathbf{D}_{t-1}$, $\mathbf{C}_{t-1}$, $\mathbf{Y}_{t-1}$ in the previous frame, λ (regularization parameter), M (sample drawing number).

Output: learned dictionary $\mathbf{D}_t$ in the current frame.

1: **Initialization**: $\mathbf{D}_t \leftarrow \mathbf{D}_{t-1}, \mathbf{C}_t \leftarrow \mathbf{C}_{t-1}, \mathbf{Y}_t \leftarrow \mathbf{Y}_{t-1}$.

2: **for** $i = 1 \rightarrow M$ **do**

3: **Step 1**: fix $\mathbf{D}_t$ to find the best coefficients,

$$\mathbf{c}_t^{(i)} = \operatorname*{argmin}_{\mathbf{c} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{y}_t^{(i)} - \mathbf{D}_t \mathbf{c}\|_2^2 + \lambda \|\mathbf{c}\|_1.$$

4: **Step 2**: fix $\{\mathbf{c}_t^{(i)}\}$ to update the dictionary,

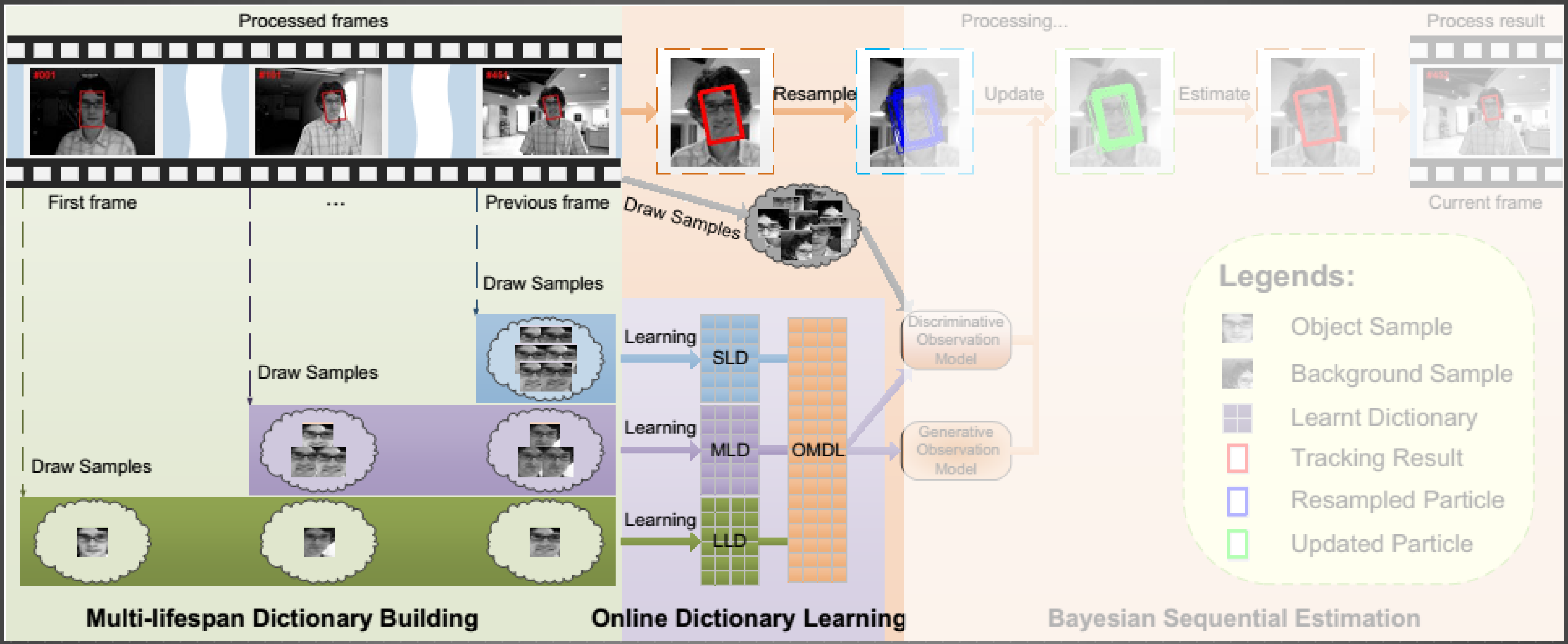
$$\mathbf{C}_t \leftarrow \mathbf{C}_t - \frac{\mathbf{C}_t \mathbf{c}_t^{(i)} \mathbf{c}_t^{(i)\top} \mathbf{C}_t}{1 + \mathbf{c}_t^{(i)\top} \mathbf{C}_t \mathbf{c}_t^{(i)}}, \mathbf{Y}_t \leftarrow \mathbf{Y}_t + \mathbf{y}_t^{(i)} \mathbf{c}_t^{(i)\top},$$

$$\begin{aligned} \mathbf{D}_t &= \operatorname*{argmin}_{\mathbf{D} \in \mathcal{D}} \sum_{j=1}^{i} \frac{1}{2} \|\mathbf{y}_t^{(j)} - \mathbf{D}\mathbf{c}_t^{(j)}\|_2^2 + \lambda \|\mathbf{c}_t^{(j)}\|_1, \\ &= \left(\sum_{j=1}^{i} \mathbf{c}_t^{(j)} \mathbf{c}_t^{(j)\top}\right)^{-1} \left(\sum_{j=1}^{i} \mathbf{y}_t^{(j)} \mathbf{c}_t^{(j)\top}\right), \\ &= \mathbf{C}_t \mathbf{Y}_t. \end{aligned}$$

5: **end for**

6: Save dictionary $\mathbf{D}_t$, intermediate variable $\mathbf{C}_t$ and $\mathbf{Y}_t$.

MULTI-LIFESPAN DICTIONARY LEARNING



MULTI-LIFESPAN DICTIONARY LEARNING

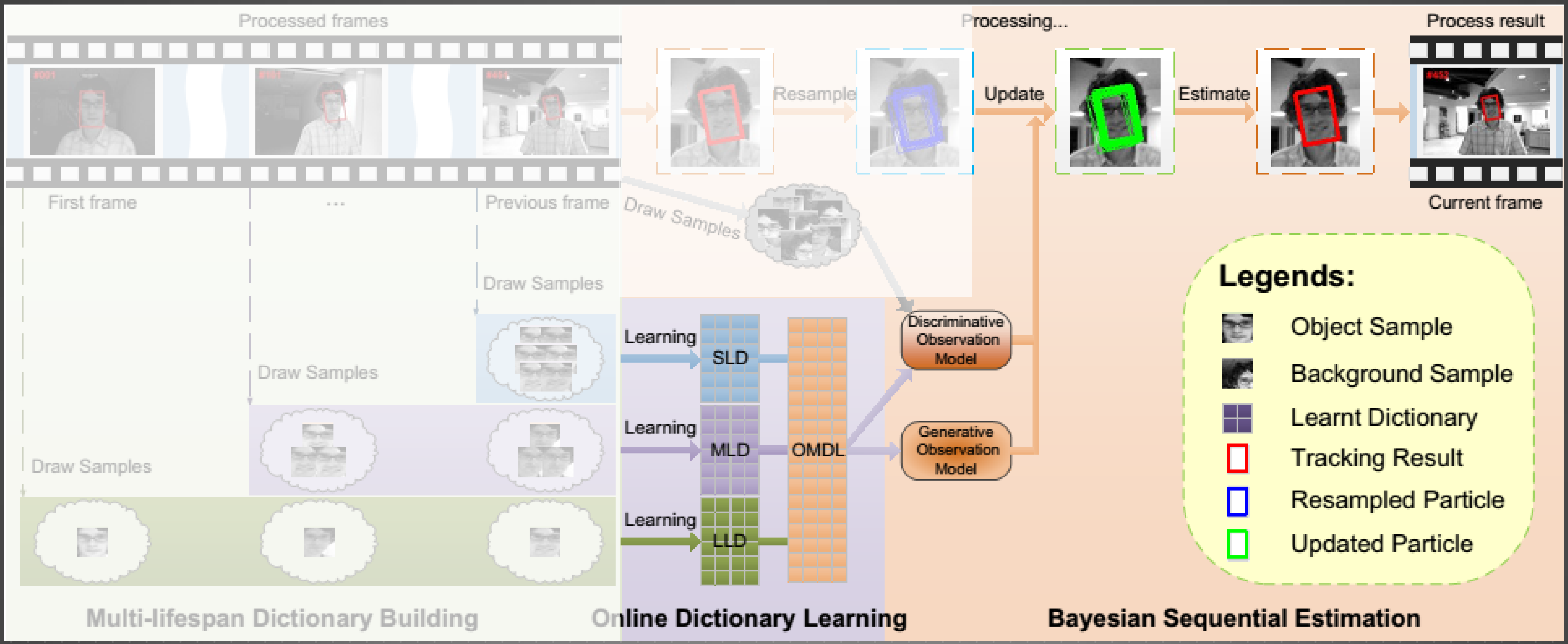
- Sample starting frame changes the lifespan
  1. SLD, Short Life Span Dictionary is learned the samples extracted from only previous frame. (Starting frame:t-1, Ending frame: t)
     - Learned for best adaptation
  2. LLD, Short Life Span Dictionary is learned the samples extracted from all the frames before current (Starting frame:1, Ending frame: t)
     - Learned for robustness
  3. MLD, Short Life Span Dictionary is learned to balance short life span and long life span (Starting frame:t/2, Ending frame: t)
     - Balances trade-off between short term adaptive model and long term robust model.
- Final Template is:

$$D^* = \{D^S, D^M, D^L\}$$

MULTI-LIFESPAN DICTIONARY LEARNING

- Learned examples of different life spanned dictionaries.

BAYESIAN SEQUENTIAL ESTIMATION



PARTICLE FILTER

- *Solving maximum posterior problem:*

$$\hat{x}_t = \underset{x_t}{\operatorname{argmax}}\, p(x_t|y_{1:t})$$

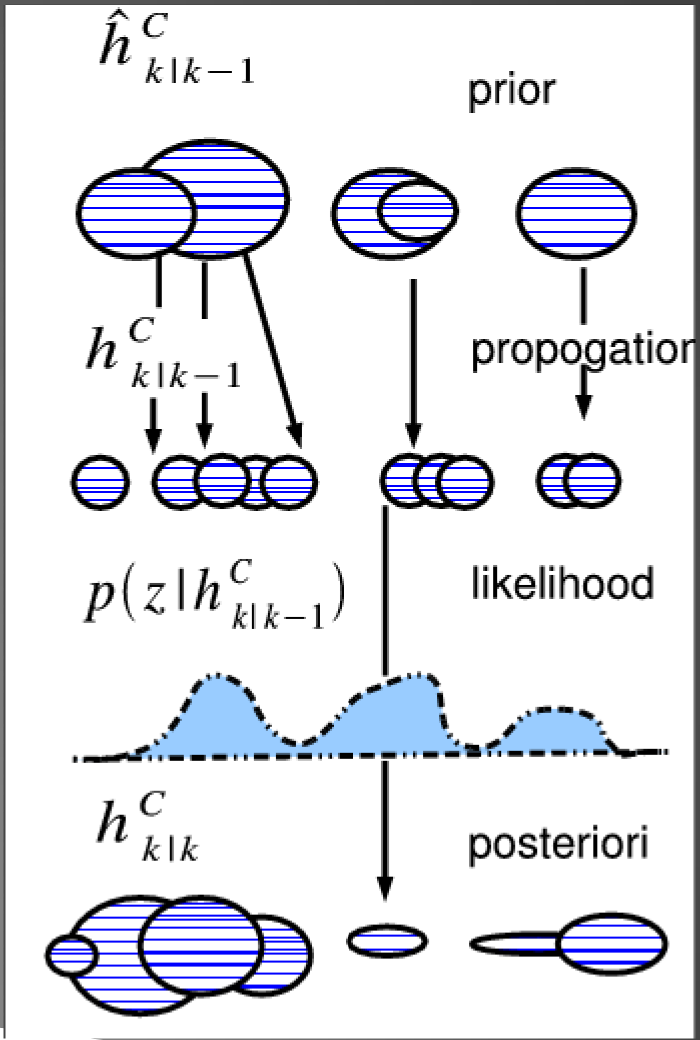
means tracking.

- Use learned OMDL as observation model in Particle Filter:

$$p(y_t|x_t) \propto g(y_t|x_t)d(y_t|x_t)$$

Generative: Fix dictionary and optimize sparsity

Discriminative: Exract negative samples and optimize with labels

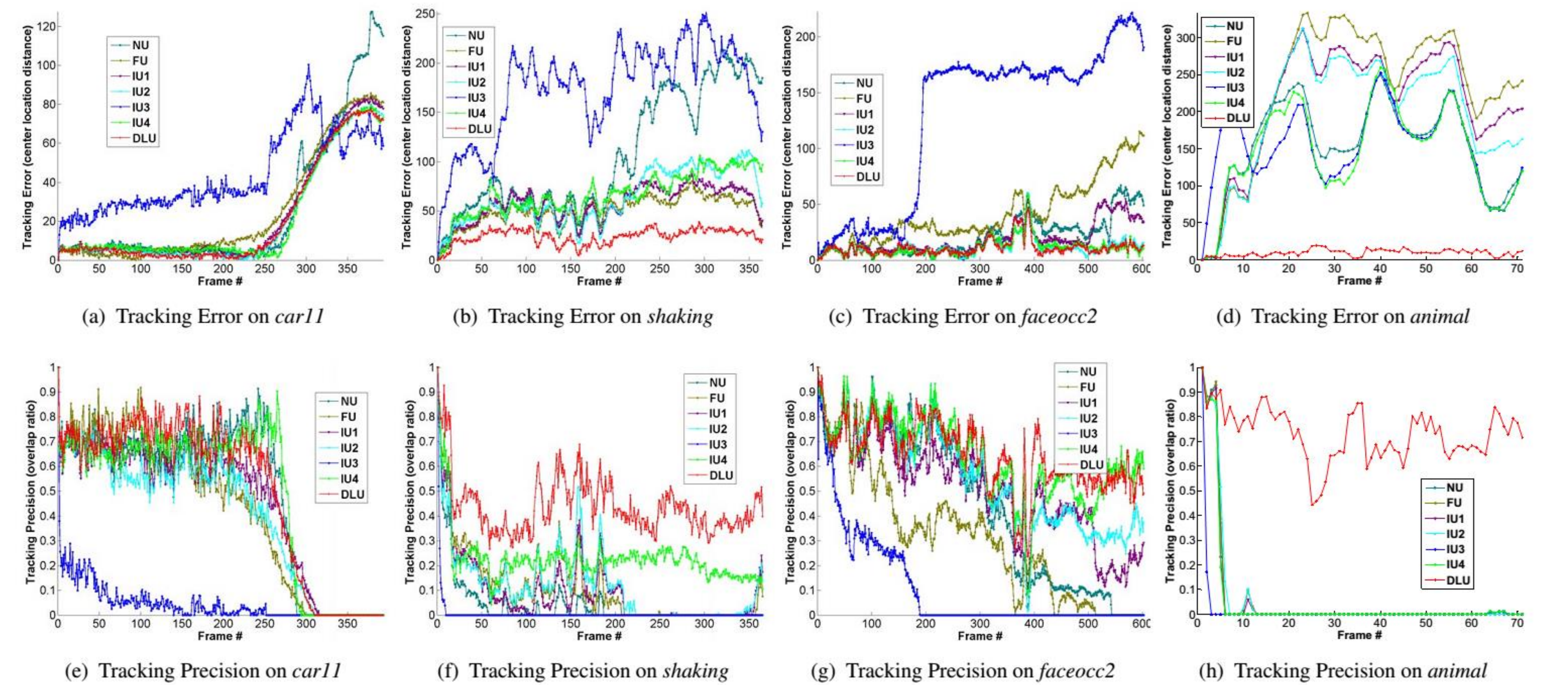


EXPERIMENTS

- Results compared with two different metric:
  - Center location distance
  - Overlap Ratio

EXPERIMENTS

- The template update method is evaluated firstly:



(a) Tracking Error on *car11*
(b) Tracking Error on *shaking*
(c) Tracking Error on *faceocc2*
(d) Tracking Error on *animal*
(e) Tracking Precision on *car11*
(f) Tracking Precision on *shaking*
(g) Tracking Precision on *faceocc2*
(h) Tracking Precision on *animal*

EXPERIMENTS

- Overall tracking error and precision:

Table 2. Average tracking errors (in pixels). The best and second best results are respectively shown in **red** and **blue** colors.

Sequence	Frag	IVT	MIL	VTD	ℓ_1	MTT	Ours
sylv	0.245	0.875	**0.156**	0.220	0.961	0.260	**0.139**
bike	2.109	0.075	0.083	0.086	0.082	**0.070**	**0.054**
car11	1.436	**0.062**	0.848	**0.065**	0.378	0.403	0.161
david	0.946	**0.057**	0.194	0.351	0.210	1.103	**0.110**
woman	1.302	1.590	1.351	**1.126**	1.305	2.281	**0.135**
animal	0.934	0.101	0.182	**0.056**	0.059	**0.047**	**0.047**
coke11	1.247	0.894	0.381	0.759	0.954	**0.338**	**0.178**
shaking	0.704	1.005	**0.222**	0.279	1.286	0.336	**0.161**
jumping	0.169	**0.094**	0.245	1.121	1.417	0.666	**0.081**
faceocc2	0.137	**0.101**	0.252	0.117	0.149	**0.097**	0.113
Average	0.923	0.485	**0.391**	0.418	0.680	0.560	**0.118**

Table 3. Average tracking precision. The best and second best results are respectively shown in **red** and **blue** colors.

Sequence	Frag	IVT	MIL	VTD	ℓ_1	MTT	Ours
sylv	0.617	0.450	0.751	**0.810**	0.323	0.770	**0.833**
bike	0.136	**0.983**	0.917	**1.000**	0.908	**1.000**	**1.000**
car11	0.097	**1.000**	0.102	**0.972**	0.682	0.687	0.781
david	0.089	**0.905**	0.537	0.615	0.435	0.320	**0.779**
woman	0.256	0.204	0.209	**0.309**	0.215	0.198	**0.440**
animal	0.099	0.887	0.747	**0.972**	**0.972**	**1.000**	**1.000**
coke11	0.051	0.119	0.271	0.068	0.085	**0.559**	**0.678**
shaking	0.222	0.025	0.414	**0.784**	0.011	0.099	**0.578**
jumping	0.690	**0.959**	0.233	0.230	0.118	0.198	**0.984**
faceocc2	0.767	0.772	0.537	0.743	0.419	**0.929**	**0.826**
Average	0.302	0.630	0.472	**0.650**	0.417	0.576	**0.790**

EXPERIMENTS

- Speed Analysis:

Algorithm	[19]	[20]	[5]	[27]	[28]	[13]	Ours
Speed	0.01fps	0.05fps	1fps	2fps	2.5fps	2.5fps	2.5fps

VISUAL RESULTS

(e) *woman*, occlusions and viewpoint changes

(i) *jumping*, fast motions and blurs

(h) *shaking*, dynamic illumination changes and scale variations

THANKS

1. D. Comaniciu and P. Meer. Kernel-based object tracking. TPAMI, 25(5):564–77, 2003.
2. B. Babenko, M. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. TPAMI, 33(8):1619–32, 2011.