Online Discriminative Dictionary Learning for Visual Tracking

***Fan Yang*[1]*, Zhuolin Jiang*[2] *and Larry S. Davis*[1]**

[1]University of Maryland College Park, MD
[2]Noah's Ark Lab, Huawei Technologies

1. Overview

- **Goal**
  - Present a visual tracking framework based on online discriminative dictionary learning (ODDL) which enforces both the reconstructive and discriminative capacity of the dictionary.
- **Approach**
  - The dictionary learning is performed in a joint manner, where the discriminative and reconstructive power are enforced in a unified algorithm.
  - A joint decision measure is presented to evaluate the reliability of candidates to improve tracking accuracy, in contrast to previous work which only relies on reconstruction error.
  - The dictionary can be updated online efficiently with a set of adaptively selected, reliable samples..

2. Related Work

- Many discriminative dictionary learning algorithms [1,2,3,4] have been proposed, but none of them have been applied to tracking effectively and efficiently.
- Sparse representations have been applied to visual tracking [5,6,7,8], but most do not use dictionary learning.
- [7] combines a sparsity-based discriminative classifier with a generative model, but the two parts are independent and combined in a heuristic way.
- [8] incorporates the discriminative power into standard sparse representations by learning a classifier. Nevertheless, the dictionary and classifier are learned separately rather than jointly.

3. Our Approach

- **Problem formulation**

Given training samples $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\} \in \mathbb{R}^{d\times n}$ with class labels $\mathbf{Y} = \{-1, 1\}$, we aim to learn a compact dictionary discriminative to distinguish the object from the background. Given a dictionary $\mathbf{D} = \{\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_k\} \in \mathbb{R}^{d\times k}$, the sparse code $\mathbf{c}_i \in \mathbb{R}^k$ for a sample is computed by

$$\mathbf{c}_i = \arg\min_{\mathbf{c}} \|\mathbf{x}_i - \mathbf{D}\mathbf{c}\|^2 + \lambda\|\mathbf{c}\|_1$$

Motivated by [9], we assign a specific label to each dictionary and learn the classifier and the dictionary simultaneously. We incorporate an ideal sparse coding error and a linear regression loss into the objective function of dictionary learning

$$\begin{aligned} \min_{\mathbf{D},\mathbf{W}} \quad & \sum_i \ell(\mathbf{D}, \mathbf{W}; \mathbf{x}_i, \mathbf{f}_i, \mathbf{l}_i) + \lambda \|\mathbf{W}\|_F^2 \\ s.t. \quad & \mathbf{c}_i = \arg\min_{\mathbf{c}} \|\mathbf{x}_i - \mathbf{D}\mathbf{c}\| + \gamma\|\mathbf{c}\|_1, i = 1, ..., n \end{aligned} \qquad (1)$$

where $\ell(\mathbf{D}, \mathbf{W}; \mathbf{x}_i, \mathbf{f}_i, \mathbf{l}_i) = (1-\mu)\|\mathbf{f}_i - \mathbf{W}\mathbf{c}_i\|_2^2 + \mu\|\mathbf{l}_i - \mathbf{c}_i\|_2^2$ is the loss function.

$\|\mathbf{l}_i - \mathbf{c}_i\|^2$ is the ideal sparse code error, where $\mathbf{l}_i = [l_{i,1}, l_{i,2}, ..., l_{i,k}]^T$ =$[1, ..., \bar{0}, 1, ..., 0]^T \in \mathbb{R}^k$ is an ideal sparse code. $\|\mathbf{f}_i - \mathbf{W}\mathbf{c}_i\|^2$ is the quadratic loss for linear regression. $\mathbf{f}_i = [0, ..., 1, ..., 0]^T \in \mathbb{R}^m$ is the label vector where the non-zero position indicates label.

- **Optimization**

We use stochastic gradient descent to optimize the objective function. The gradient with respect to **W** is

$$\frac{\partial \ell}{\partial \mathbf{W}} = (1-\mu)(\mathbf{W}\mathbf{c}_i - \mathbf{f}_i)\mathbf{c}_i^T + \lambda\mathbf{W}$$

The dictionary **D** is not explicitly defined in ℓ but implicitly defined on the sparse code. To obtain the gradient of **D**, we adopt the implicit differentiation algorithm [10].

- **Classification**

The key idea of classifying a sample **x** using learned dictionary is combining the similarity between **x** and the training samples with the classification score from the classifier.

We define a joint decision measure

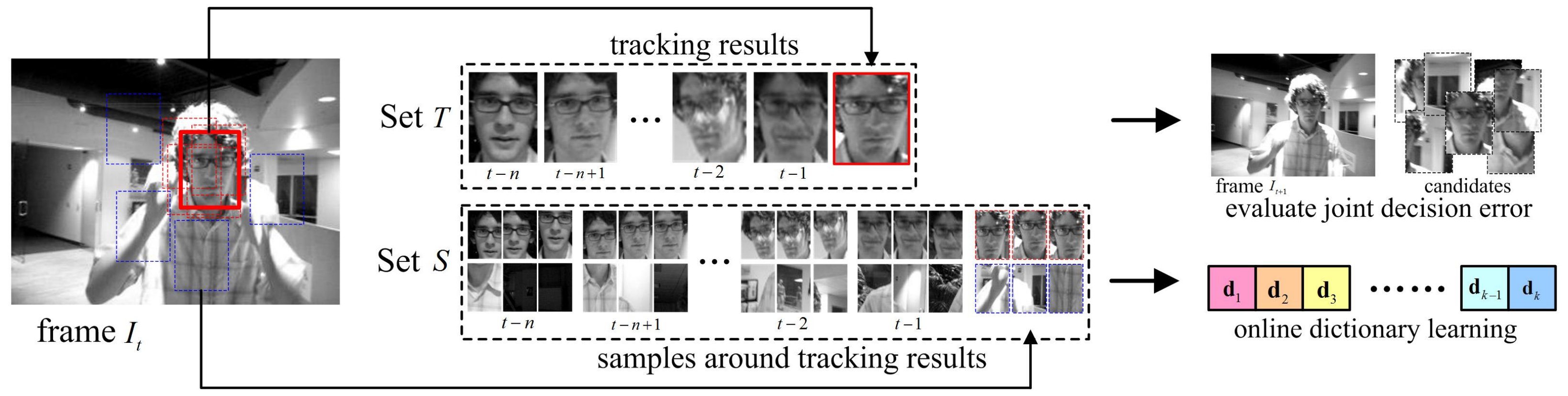
$$\varepsilon(\mathbf{x}) = \|\mathbf{x}_{tr} - \mathbf{D}\mathbf{c}\|^2 + \rho\|\mathbf{f} - \mathbf{W}\mathbf{c}\|^2 \qquad (2)$$

where $\mathbf{x}_{tr}$ is the weighted average of the elements in a set (see below). The two terms measure the quadratic appearance reconstruction error and linear regression loss.

Joint decision leads to more accurate results.

- **Tracking**
  - Positive and negative samples are sampled in the first frame to learn the dictionary.
  - In a new frame, a number of candidates are randomly sampled around the tracking result in the last frame.
  - To compute $\varepsilon_{rec} = \|\mathbf{x}_{tr} - \mathbf{D}\mathbf{c}\|^2$
    - accumulate the feature extracted from the bounding box at the optimal location into a set T.
    - We associate with each element in T the weight $w = e^{-\varepsilon}$.
    - $\mathbf{x}_{tr}$ is computed as the weighted average of the elements in T.
  - To update dictionary
    - construct a set S to store new positive and negative samples.
    - when S reaches a pre-defined size, we apply ODDL algorithm to it to update dictionary.
    - empty S.

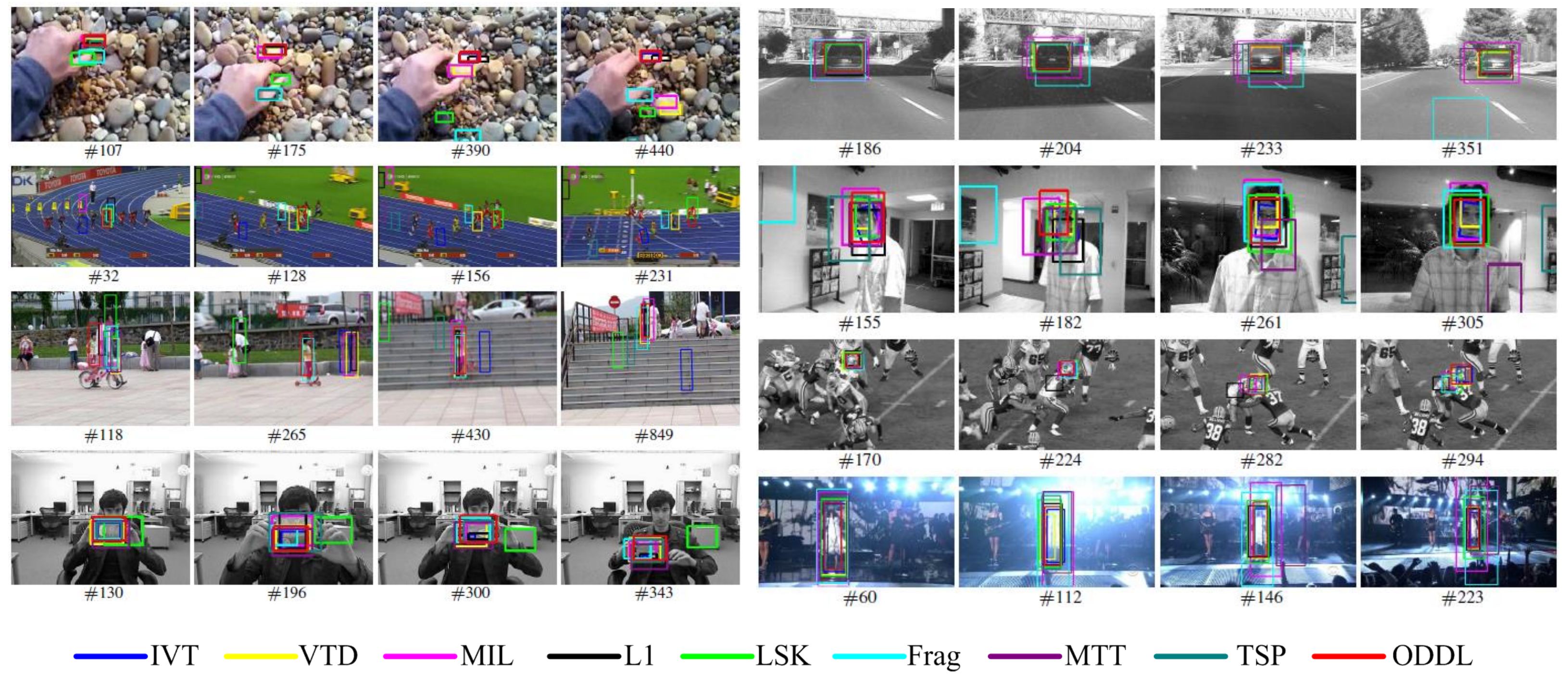
```
Algorithm 2 Tracking by ODDL
Input: Frames $I_1, I_2, ..., I_t$.
Output: Tracking results in each frame $x^1, x^2, ..., x^t$
Initialization $I_t$ $(t = 1)$
    Given initial $x^1 = (c_x^1, c_y^1, s^1)$, sample $N^+$ positive and $N^-$ negative samples;
    Extract features to form $\mathbf{X}$ with label $\mathbf{Y}$;
    Initialize $\mathbf{D}^0$ and $\mathbf{W}^0$;
    Add $\mathbf{X}$ into set $S$, and add the initial state $x^1$ into set $T$;
For each new frame $I_t$ $(t > 1)$
    Sample $P$ candidates around the tracked object $x^{t-1}$ according to distribution $p(x^t|x^{t-1})$ and extract features;
    Compute the sparse code $\mathbf{c}$ for each candidate;
    Apply Equation (2) to each candidate to compute $\varepsilon$ using $\mathbf{D}$, $\mathbf{W}$ and elements in $T$;
    Select the candidate with the smallest $\varepsilon$ as the tracking result $x^t$;
    Sample $N_{new}^+$ positive samples around $x^t$ and $N_{new}^-$ negative samples far away from $x^t$ to obtain new samples $\mathbf{X}_{new}$;
    Add tracking result $x^t$ into $T$ and $\mathbf{X}_{new}$ into $S$;
    If $length(T) > U_1$, remove the oldest element from $T$;
    If $length(S) = U_2$, apply Equation (1) to all elements in $S$ to update $\mathbf{D}$ and $\mathbf{W}$; then empty $S$ for future samples;
    Output $x^t$ and proceed to the next frame $I_{t+1}$.
```



4. Experiments

- **Qualitative comparisons**

Compared with 8 state-of-the-art trackers on 9 public sequences. Some tracking results are shown below.



- **Quantitative comparisons**

Results of our tracker are averaged from 5 runs on each sequence. For quantitative results, we use the average center location error (CLE) and successful tracking rate (STR). In computing STR, we employ the PASCAL score which is obtained by $s = \frac{area(R_{GT} \cap R_T)}{area(R_{GT} \cup R_T)}$ **, where R_{GT} and R_T are groundtruth region and tracked result.**

Table 1. Average center location error (CLE). Trackers using the reconstruction error only, using the linear regression loss only, without online update and with blind update are denoted as ε_{rec}, ε_{cls}, ODDL- and ODDL+. Red is the best and blue is the second best.

Sequence	IVT	VTD	MIL	ℓ_1	LSK	Frag	MTT	TSP	ODDL	ε_{rec}	ε_{cls}	ODDL-	ODDL+
animal	127.5	12.0	66.5	26.6	70.0	92.1	9.2	216.8	14.9	16.3	145.5	16.4	131.0
car4	2.9	12.3	60.1	4.1	3.3	179.8	37.2	3.6	4.3	16.2	54.8	9.4	6.2
david	3.6	13.6	16.2	7.7	4.9	76.7	36.5	73.4	7.5	10.8	59.3	18.4	8.2
football	6.3	4.0	13.7	29.9	14.6	16.3	8.0	14.6	6.3	9.2	40.2	6.3	4.2
singer	8.6	4.1	15.2	3.2	14.6	22.1	41.3	6.3	9.3	16.1	35.3	9.4	43.9
stone	2.3	31.4	32.3	19.2	68.7	65.9	2.5	97.8	4.8	10.3	17.1	4.7	33.2
bolt	189.2	38.6	376.5	377.8	10.3	96.8	386.1	317.0	9.5	371.2	9.8	117.0	372.1
girl	154.8	50.6	53.1	50.2	244.9	67.2	576.6	201.2	14.7	267.2	73.0	44.9	37.8
twinnings	14.1	8.6	6.4	10.9	61.9	11.3	11.6	4.1	8.2	20.4	27.3	41.0	18.0

Table 2. Successful tracking rate (STR). Trackers using the reconstruction error only, using the linear regression loss only, without online update and with blind update are denoted as ε_{rec}, ε_{cls}, ODDL- and ODDL+. Red is the best and blue is the second best.

Sequence	IVT	VTD	MIL	ℓ_1	LSK	Frag	MTT	TSP	ODDL	ε_{rec}	ε_{cls}	ODDL-	ODDL+
animal	0.28	0.78	0.14	0.58	0.41	0.02	0.79	0.31	0.86	0.61	0.21	0.82	0.32
car4	0.99	0.99	0.27	0.99	0.98	0.27	0.38	0.99	0.99	0.95	0.44	0.99	0.99
david	0.91	0.37	0.42	0.86	0.98	0.16	0.40	0.29	0.92	0.71	0.03	0.45	0.73
football	1.00	0.98	0.69	0.52	0.77	0.67	0.86	0.32	0.98	0.94	0.18	0.99	0.99
singer	0.94	0.95	0.23	0.98	0.49	0.23	0.34	0.98	0.95	0.68	0.22	0.95	0.45
stone	0.67	0.60	0.50	0.29	0.10	0.24	0.84	0.09	0.87	0.59	0.20	0.77	0.52
bolt	0.02	0.35	0.04	0.06	0.43	0.10	0.02	0.01	0.68	0.02	0.64	0.31	0.02
girl	0.09	0.61	0.36	0.09	0.09	0.53	0.09	0.09	0.76	0.06	0.46	0.34	0.61
twinnings	0.28	0.85	0.53	0.34	0.23	0.39	0.57	0.80	0.80	0.65	0.44	0.40	0.38

The average CLE and STR of our tracker is 8.8 and 0.868, which are the best.

5. Key References

1. M. Yang, L. Zhang, X. Feng, *et al.* Fisher **Discrimination Dictionary Learning for sparse representation**. ICCV, 2011.
2. Q. Zhang and B. Li. **Discriminative k-svd for dictionary learning in face recognition**. CVPR, 2010.
3. J. Mairal, F. Bach, J. Ponce *et al.* **Discriminative learned dictionaries for local image analysis**. CVPR, 2008.
4. J. Mairal, M. Leordeanu, F. Bach, *et al*. **Discriminative sparse image models for class-specific edge detection and image interpretation**. ECCV, 2008.
5. X. Mei and H. Ling. **Robust visual tracking using L1 minimization**. ICCV, 2009.
6. B. Liu, J. Huang, L. Yang *et al*. **Robust tracking using local sparse appearance model and k-selection**. CVPR, 2011.
7. W. Zhong, H. Lu, and M.-H. Yang. **Robust object tracking via sparsity-based collaborative model**. CVPR, 2012.
8. Q.Wang, F. Chen, W. Xu, *et al*. Online discriminative object tracking with local sparse representation. WACV, 2012.
9. Z. Jiang, Z. Lin, and L. S. Davis. **Learning a discriminative dictionary for sparse coding via label consistent k-svd**. CVPR, 2011.
10. J. Yang, K. Yu, Y. Gong *et al.* Linear spatial pyramid matching using sparse coding for image classification. CVPR, 2009