Video target tracking with fisher discriminant dictionary learning

Jian-Feng Zheng, Ji Zhang*

School of Information Science and Engineering, ChangZhou University, ChangZhou 213164, China

Received 1 July 2014, www.cmnt.lv

Abstract

As one of the state-of-the-art tracking methods based on sparse coding, l1-tracker finds the target with the minimum reconstruction error from the target template subspace. But the high computational costs restrict its application in practical terms heavily. In this paper, we incorporate the discriminant information into original l1-tracker, and introduce it into the tracking framework, called FD2LT. In our framework, tracking is considered as a problem consisting of object location with dictionary learned in the last frame in generative tracking framework, training samples selection, and dictionary learning with fisher discriminant dictionary learning (FDDL). With our method, the dictionary is much smaller than that in original one, moreover, without loss of tracking performance (and even better in some scenarios). The discriminant power explored from the dictionary is used in generative tracking. Experimental results demonstrate the effectiveness and efficiency of the proposed tracking algorithm.

Keywords: Fisher discrimination dictionary learning, generative and discriminant tracking, sparse coding, video object tracking

1 Introduction

Recently, computer vision is widely used in many fields. As one of the most exciting fields, target tracking looks for some specified objects pre-definited in video streams artificially. Targets change dynamically and uncertainly in video sequence, because of occlusion, noisy, varying and so on. Many tracking algorithms have been proposed, such as *IVT*, *TLD*, *CovTrack*, l_1-*tracker* [1, 2].

Based on *sparse coding* (*SC*) [3], Mei proposed l_1-tracker [4], where many challenging problems presented in tracking are addressed seamlessly. However, computational cost of l_1-*tracker* is quite expensive to achieve efficient tracking. Moreover, the discriminant ability of the dictionary is not explored. An alternative way is to construct the dictionary with rich representation ability and few atoms, which is called *dictionary learning* (*DL*) [5]. Many *DL* algorithms have been proposed in last several years. *K-SVD* [6] learns the dictionary from training sets, which is suitable for reconstruction, rather than discrimination. Mairal introduces discriminant constraint into *K-SVD* for classification [7], which is not convex; Tosic proposes a new method for learning the over-complete dictionary to represent the stereo images [8], but not for classification like *K-SVD*; Yang's *Fisher discriminant dictionary learning* (*FDDL*) aims to learn a structured dictionary for face recognition, whose sub-dictionaries have specific class labels[5]. Our method is motivated by Yang's *FDDL*. The rest of this paper is organized as follows: sparse coding, l_1-*tracker* and *FDDL* are introduced in section 2. In section 3, we analysis the shortcoming of *FDDL*, then improves and introduce it into tracking, called *FD*^{2}*LT*. The convergence of *FD*^{2}*LT* is demonstrated numerically. Experimental results with *FD*^{2}*LT* and some competitive algorithms are reported in section 4. Finally, we will conclude our work and propose future work.

2 Related Work

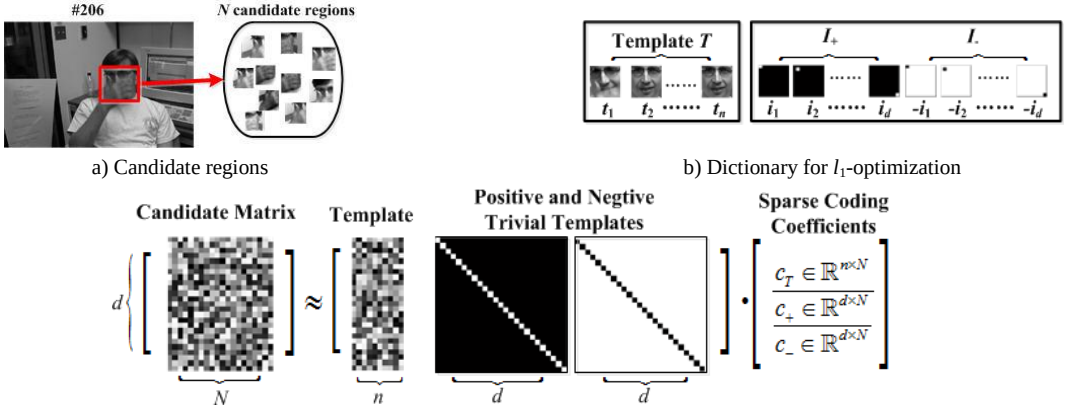
2.1 SPARSE CODING FORMULATION AND SPARSE CODING BASED TRACKING (L_1-TRACKER)

SC is an attractive signal reconstruction method, and the main task of which is to reconstruct a query signal $y \in \mathbb{R}^{d\times 1}$ over the over-complete dictionary $D \in \mathbb{R}^{d\times n}$ with a sparse coefficient vector $x \in \mathbb{R}^{n\times 1}$:

$$\min_x \|y - Dx\|_F^2 + \lambda \|x\|_1 . \quad (1)$$

where, $\|.\|_F$ and $\|.\|_1$ are the Frobenius-norm and l_1-norm, respectively. l_1-*tracker* is proposed based on *SC* [4], as shown in Figure 1. Suppose that, the target for tracking has been located in #205 (where the red box indicated and l_1-*tracker* initialized in #206), and *N* candidate regions are generated with Bayesian inference around it. With *n* templates learned from the last frame and 2*d* trivial templates (*d* positive ones and *d* negative ones, *d* is the dimension of 1-D stretched image) in Figure 1b), Equation (1) can be solved like Figure 1c). Furthermore, with these trivial templates, Mei adds *non-negative constraint* (*NNC*) $x \geq 0$ into Equation (1). Reconstruction errors of all candidates with *SC* coefficients can be used to determine the weights for each candidate, the target in #206 can be located with the sum of weighted candidates, and the updating strategies of dictionaries can be seen in [4].

* ***Corresponding author*** e-mail: zhangji@cczu.edu.cn



a) Candidate regions

b) Dictionary for l_1-optimization

c) SC of Candidates over dictionary

FIGURE 1 Original l_1-tracker

2.2 FISHER DISCRIMINANT DICTIONARY LEARNING (*FDDL*)

Mei's experiments show that, l_1-*tracker* has excellent performance when comparing with some state-of-the-art trackers [4]. But it is inefficiency, caused by the number of candidates (particles) N and the size of over-complete dictionary D, affords its application in real-time tracking severely. In original l_1-*tracker*, in order to achieve robust tracking, N must be very large, while the dimension of dictionary D is $d\times(2d+n)$ in Equation (1). In our experiment setting, N=200, and D for l_1-*tracker* is 1600×3210. It is quite nature that, how to reduce the number of candidates and the size of dictionary without (or with a little) loss of tracking accuracy, are two important issues in l_1-*tracker*. The former depends on the improvement of PF tracking framework [1], which is not mentioned in this paper; and the latter, specifically, how to construct dictionary which not only contains few atoms, but also has good ability of representation, is exactly the main task of our algorithm.

FDDL is proposed for face recognition, which learns c structured dictionaries $D=[D_1,D_2,\ldots, D_c]$ for each class of facial images, instead of a whole shared dictionary for all images, where D_i is the class-specified sub-dictionary associated with class-i, and c is the class number. Let $Y=[Y_1,Y_2,\ldots, Y_c]$ and $X=[X_1,X_2,\ldots,X_c]$ denote the set of training samples and the coding coefficient matrix of Y over D, respectively, where Y_i is the sub-set of the training samples from class i, X_i is the sub-matrix containing the coding coefficients of Y_i over D and $Y\approx DX$. *FDDL* can be formulated as following:

$$J_{(D,X)} = \min_{(D,X)} \left\{ r(Y,D,X) + \lambda_1 \|X\|_1 + \lambda_2 f(X) \right\}, \quad (2)$$

where, $f(X)=tr(S_W(X))-tr(S_B(X))-\eta\|X\|_F^2$ is a discriminative constraint imposed on X, which makes D discriminative for the samples in Y; $S_W(X)$ and $S_B(X)$ are within- and between-class scatters of X, respectively; λ_1, λ_2 are used to tune the influences of each term; $r(Y,D,X)=\sum_{i=1,\ldots c} r(Y_i,D,X_i)$ is the discriminative fidelity term, and:

$$r(Y_i,D,X_i) = \|Y_i - DX_i\|_F^2 + \|Y_i - D_i X_i^i\|_F^2 + \sum_{j=1,j\neq i}^{c} \|D_j X_i^j\|_F^2, \quad (3)$$

The first two terms ensure that Y_i can be represented by D and D_i approximately with X_i and X_i^i, respectively; the last one ensures the representation of Y_i over $D_j(i\neq j)$ is small. Some important terms in $r(Y_i,D,X_i)$ is shown in Figure 2. Consider some $y\in\mathbb{R}^{d\times 1}$ (e.g. a stretched face image), $\tilde{y}=Dx$ and $\hat{y}=D_i x_i$ are reconstruction results of y over the whole dictionary D and the class-i dictionary D_i, respectively. We denote the first two terms in Equation (3) as $\tilde{e}$ and $\hat{e}$ in Figure 2. The minimization of Equation (3) can be divided into two sub-problems: updating X by fixing D, and updating D by fixing X [9].

3 Our tracking framework with FDDL

3.1 IMPROVED FDDL

As mentioned above, Equation (3) minimizes $\tilde{e}$, $\hat{e}$ and $\sum_i y_j$ $(i\neq j,\ j=1,\ldots,c)$ in Figure 2. But we find that, it is not sufficient for reconstructing signal y based on Equation (3). Denote by $y'=D_i x'$ the approximation of $\tilde{y}$ over D_i, and $e'=\hat{y}-y'$, $e=y-y'$, $e^*=\tilde{y}-y'$. Here, we use *AR* face database to validate the insufficient. *AR* contains 700 face images from 100 individuals (7 images for each one). In our experiment, 100 images are selected as query singles randomly, and the rest 600 images are treated as dictionary atoms. For each selected query image, $\|e\|_F$, $\|e^*\|_F$, $\|e'\|_F$, $\|\tilde{e}\|_F$ and $\|\hat{e}\|_F$ are calculated on 600 labelled training images, and plotted in Figure 3. It is clear to see that:

1) Minimization of $\|\tilde{e}\|_F$ and $\|\hat{e}\|_F$ cannot guarantee minimization of $\|e\|_F$, $\|e^*\|_F$ and $\|e'\|_F$;

2) As $\|e'\|_F>0$ (unless the total dictionary D is consisting of i^{th} class dictionary D_i merely, which is not practical), $\|\hat{e}\|_F<\|e\|_F$. And minimization of $\|\tilde{e}\|_F$ and $\|\hat{e}\|_F$ in Equation (3) has nothing to do with minimization of $\|e^*\|_F$;

3) Beside $\|\tilde{e}\|_F$ and $\|\hat{e}\|$, $\|e\|_F$, $\|e^*\|_F$ and $\|e'\|_F$ also

play important roles in face recognition. They report the difference between i^{th} class information contained in D_i and that in D. The smaller these three terms, the more positive information contained in D_i and the less in $D\backslash D_i$;

4) For each image, $||e||_F$ is maximum among all five residual terms, and the minimization of $||e||_F$ can be considered as the upper bound of all five representation residual terms. The similar results can be obtained on *Yale*, *ORL* database. Therefore, we can rewrite Equation (3) as:

$$r'(Y,D,X)=\sum_{i=1}^{c} r'(Y_i,D,X_i)= \sum_{i=1}^{c}\left(\left\|Y_i-D_iX_i'^{i}\right\|_F^2+\sum_{j=1,j\neq i}^{c}\left\|D_jX_i'^{j}\right\|_F^2\right), \tag{4}$$

where, Y_i is the class-i subset of the training samples, X'^{j}_i is the coding coefficient matrix of $\tilde{Y}_i$ (reconstruction of Y_i over D_j). Note that, we use X'^{j}_i in Equation (4) instead of X_i^j in Equation (3). Denote $f(X')=tr(S_W(X'))-tr(S_B(X'))-\eta||X'||_F^2$, thus Equation (2) can be rewritten as:

$$J_{(D,X')}=\min_{(D,X')}\left\{r'(Y,D,X')+\lambda_1\|X'\|_1+\lambda_2 f(X')\right\}. \tag{5}$$

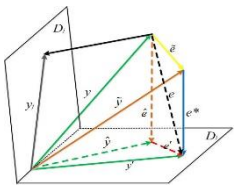


FIGURE 2 Residual terms

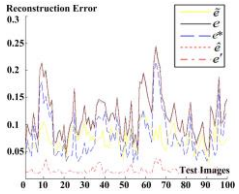
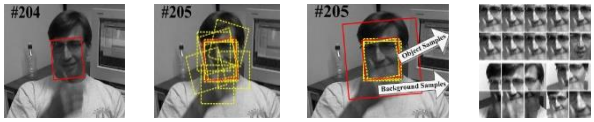


FIGURE 3 Residual terms



(a)Target in #204 (b)Candidates (c)Sampling range (d)Training set

FIGURE 4 Object location and training samples selection

3.2 FDDL BASED TRACKING

In this subsection, our tracking framework called *FDDL based tracking* (*FD*^{2}*LT*) is proposed, which includes three components: target location with the dictionary learned in the last frame, training samples selection for dictionary learning, and discriminant dictionary learning.

3.2.1 Target Location and Training Samples Selection

According to the target location in the last frame (red box in Figure 4a), a number of candidate regions can be extracted with Bayesian inference (dotted boxes in Figure 4b), then target can be distinguished from those candidates. Many *SC* methods have been proposed for classification [5-7]. Most of them (no matter supervised, unsupervised and semi-supervised classification) work well based on a huge number of training samples, but as discussed above, *DL* with so many samples (e.g. Y in Equation (2-5)) lead to the tremendous computational costs. It is not critical for classification, as training and updating of classifiers are off-line in advance and classifying the new-arrival sample is very fast with the classifiers; but it is impatient in tracking, as the latter is real-time.

In order to achieve robustness and efficiency, under the framework of PF framework, we seek the most likely candidate region as target in current frame, then generate object/positive samples and background/negative samples as following. Select 10 regions extremely nearby the object (small red rectangle in Figure 4c) as positive samples Y_O, and 10 regions (including up-left, up, up-right, left, right, left-down, down, left-right, 1/3 bigger and 1/3 smaller than the small red box) as negative samples Y_B. Notice that, as shown in Figure 4d), in order to remain information of target during tracking, we always fix the last one in Y_O with the target selected artificially in the first frame; and the last two background regions are used to deal with the scale changing of target.

3.2.2 Dictionary Learning

Most of discriminant tracking methods are based on the assumption that, the appearances of target and background (near the object) change slightly frame by frame. Thus, we can represent candidate regions selected in current frame using the dictionary D_{old} learned in the last frame, and locate target as shown in the last section; afterwards, update D_{old} with 20 selected samples to help tracking in the successive frame. So, updating the dictionary is also a critical problem here.

Suppose that, $D_{old}=[D_{old\_O}, D_{old\_B}]$ is the last learned dictionary, and labeled training samples $Y=[Y_O,Y_B]$ are selected as Figure 4d). The problem of *DL* is how to update D_{old}, such that Y can be represented by new dictionary D_{new} with as less error as possible. According to improved *FDDL* proposed above, objective function for tracking is:

$$J_{(D',X')}=\min_{(D',X')}\left\{\left\|Y_O-D'_OX'^{O}_O\right\|_F^2+\left\|Y_B-D'_BX'^{B}_B\right\|_F^2+\left\|D'_BX'^{B}_O\right\|_F^2+\left\|D'_OX'^{O}_B\right\|_F^2+\lambda_1\|X'\|_1+\lambda_2\left(S_W(X')-S_B(X')-\eta\|X'\|_F^2\right)\right\}, \tag{6}$$

where, $\tilde{Y}=\left[\tilde{Y}_O,\tilde{Y}_B\right]$ are approximation of $Y=\left[Y_O,Y_B\right]$ over D_{old}; $X'=\left[X'_O,X'_B\right]$; $X'_O=\left[X'^{O}_O,X'^{B}_O\right]$ and $X'_B=\left[X'^{O}_B,X'^{B}_B\right]$ are SC coefficients of $\tilde{Y}_O$ and $\tilde{Y}_B$ over updated D', respectively.

3.2.3 Solving modified FDDL

Optimization of $J(D',X')$ in Equation (6) is not convex with respect to D' and X' simultaneously, and we can divided it into two sub-problems: updating X' by fixing D'; and updating D' by fixing X'. The procedures are iteratively implemented in our previous work, the reader can refer to [10] for detail.

3.3 REPRESENTATIVE/DISCRIMINANT ABILITIES OF DICTIONARY

Figure 5 shows some tracking results with our modified and original *FD²LT* in #205 (Figures 5a and 5b) and #355 (Figure 5c) of *Dudek* sequence, and the further and detail results can be seen in Figure 8b). Experimental settings are the same as those in section 4.1. It is clear to see that:

1) Tracking results of our two *FD²LT* methods are similar in #205, both of them deal with deformations and occlusions of target steadily;

2) *FD²LT* with our improved *FDDL* remains object information well after dictionary updating, while the latter destroys almost all object information, which is shown in 10 object atoms (updated dictionary in current frame) in the lower-left corner of Figures 5a and 5b, respectively;

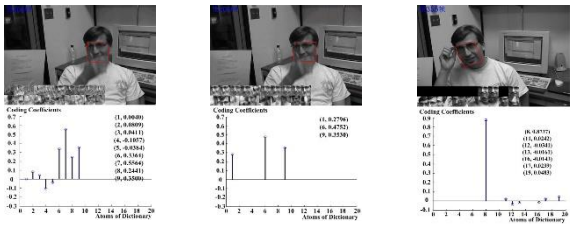
(3) We also assert that, coding coefficients used to represent target in #205 with improved *FD²LT* is sparser than that with original *FD²LT*. The button row of Figures 5a and 5b shows the coding results of these two methods.

With experiments in section 4, we get similar conclusions and list as following:

1) Our improved *FDDL* is much sparser than original *FDDL*, when coding the same signal;

2) Dictionary learning with our improved *FDDL* has stronger representation ability than original *FDDL*.

The second conclusion can be expressed in Figure 5c. With our method, after a few successive frames, some atoms in dictionary are almost close to zeros. As shown in Figure 5c, only 3 object atoms and 10 background atoms are used to represent the object with our improved *FDDL*, while it appears rarely when we use original *FDDL* in tracking. But it does not always benefit for tracking, especially when object changes heavily, as if the number of dictionary atoms is too small, they cannot remain the object information and adapt the change of object, simultaneously. In order to maintain rich ability of representation, we set those atoms with all zero elements as mean of all non-zero atoms.



(a) Original *FD²LT* (b) Improved *FD²LT* (c) Special Example

FIGURE 5 Results of *FD²LT* on Dudek

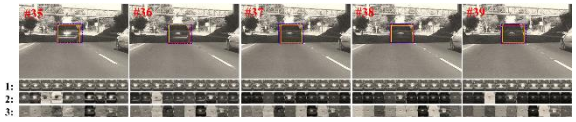


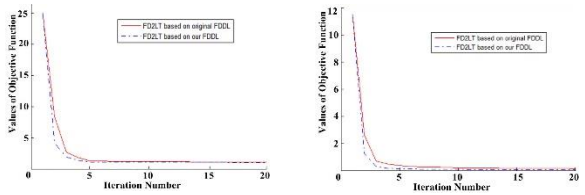
FIGURE 6 NNC. Top: Tracking results of successive five frames with 1: l_1-tracker(Bule), 2: original *FD²LT* (Red) and 3: improved *FD²LT* (Yellow); Bottom: dictionary templates of three algorithms

3.4 WITH OR WITHOUT NON-NEGATIVE CONSTRAINTS?

In original l_1-*tracker*, *non-negative constraints* (*NNC*) for coding coefficients are not only used to restrict them to be positive, but also used to filter out clutter that is similar to target templates as reversed intensity patterns, as shown in Figure 6. In these successive five frames in *Car4* sequence, the car for tracking is driven from brightness to darkness. With *NNC*, original l_1-*tracker* remains the template information of the car, no matter it is in the shadow or not. But we have referred that, l_1-*tracker* costs heavily, because of the large dictionary(almost half of the dictionary, i.e. the negative trivial template, is designed for *NNC*) and l_1-*optimization*, as shown in Figure 1 and section 2.2, respectively.

In our *FD²LT*, which can be considered as the discriminant extended version of l_1-*tracker*, *NNC* is omitted, because the changes of reversed intensity patterns can be incorporated into dictionary after learning. We find that, in Figure 1, the dictionaries learned with original and improved *FDDL* include not only positive templates, but also negative ones, which are used to deal with the problem of reversed intensity patterns. According to this, our two *FD²LT* algorithms are much faster than original l_1-*tracker*, and the performances of them are nearly the same, which can be seen in Section 4.

To compare the convergence we verify that for our two *FD²LT* trackers with *Dudek* sequence quantification ally. In #206, almost the entity face of the man (target for tracking) is blocked by his hand, which is slightly occurred in #205, see Figures 1a and Figures 5a, respectively. This is still a heavy representative change for modern trackers. There is reason to regard that, dictionaries learned in #205 with above two *FDDL* methods are not very suitable for tracking the object in #206. We get the same result in Figure 7a that, initial energies (values of objective function, calculated with Equation (7) are very large. During the iterative procedure, energies decayed quickly, and *FD²LT* based on our improved *FDDL* convergences faster than the other one. Figure 7b shows the same result with the average energies convergence of the whole tracking procedure. In our experiment, we set number of iterations as 5, in order to keep our tracker working fast.



a) Energy of Dudek #206 b) Average Energy of Deduk

FIGURE 7 Convergence Curves of Energies (Values of Objective Function)

4 Experiment

4.1 EXPERIMENT SETTING

In order to evaluate our trackers, we conduct experiments on eight challenging video sequences, including *Surfer*, *Dudek*, *Faceocc2*, *Animal*, *Girl*, *Car11*, *Stone*, *Jumping*, *Car4* and *Pktest02* with 375, 1145, 819,71,500,393,200,100,400 and 120 frames, respectively (see Figure 8). These sequences cover almost all challenges in tracking, including occlusion (even heavy), motion blur, rotation, scale variation and complex background, etc. For comparison, we select four state-of-the-art trackers, including *incremental learning based tracker* (*IVT*, a familiar discriminant tracking method [11]), *Tracking -Learning-Detection* (*TLD*, a real-time long-term discriminant tracking method [12]), *CovTrack*(a generative tracker on Lie-group [13]) and l_1-*tracker*(a generative tracking methods[4])[1].

4.2 EXPERIMENTAL RESULTS

We evaluate the above-mentioned algorithms using the centre location errors, as shown in Figure 8. Overall, our *FD*^{2}*LT* performs well against the other state-of-the-art algorithms.

For occlusion, five algorithms except *IVT* work steadily roughly, especially at #206, #366 of the *Dudek* sequence in Figure 8b (head for tracking is covered by hand and glasses) and at #85, #108, #433 of the *Girl* sequence in Figure 8e (head for tracking turns right, turns back and blocks by someone else). After the target recovers from occlusion, these five trackers can seek it quickly. *IVT* works poorly, even lost the target from #10 in the *Girl* sequence, because of the number of positive and negative samples are limited (in consideration of the learning efficiency). Incremental updating of classifier in *IVT* is less effective; *CovTrack* has large size of candidates (with the definition of integral image, feature extraction of these candidates is so fast, and the costs of which can be ignored), which makes it robust for occlusion, scale variation and blur. Thanks to *P-N* expert learning and detection when loses the target, *TLD* often performs good when confronts occlusion. When occlusion happens, *TLD* abandons tracking, see the yellow regions in Figure 8. When target appears again, *TLD* can obtain it. But there are also some exceptions, see Figure 8c from #377 to the end of the *Faceocc2* sequence. *TLD* loses the head for tracking from #377 because of the occlusion and rotation of the target, and after that, *TLD* never detects the target again; while original l_1-*tracker* and the derivative two *FD*^{2}*LT* trackers, which have strong representative abilities based on the large size of dictionary and good performances of dictionary learning, respectively, performance excellently.

For motion blur, our two trackers work better than *IVT* and l_1-*tracker*, moreover, *CovTrack* and *TLD* also reveal their abilities for blur, see #4, #9 and #38 in Figure 8d and #16, #29 and #53 in Figure 8h. The animal runs and jumps fast(motion blur) with splashing a lot of water splashing (occlusion). *IVT* and l_1-*tracker* fail both from #4, and never recover after that. Original and improved *FD*^{2}*LT* lost target at #28 and recover at #38, Figure 8d. And at #12,#21 and #43,#71, improved *FD*^{2}*LT* works better than original *FD*^{2}*LT*, *CovTrack* and *TLD*. *TLD* loses target from #24 to #33, from #53 to #71 in *Animal* sequence and from #33 to #36, from #41 to #48, from #56 to #70, from #73 to #90 in *Jumping* sequence.

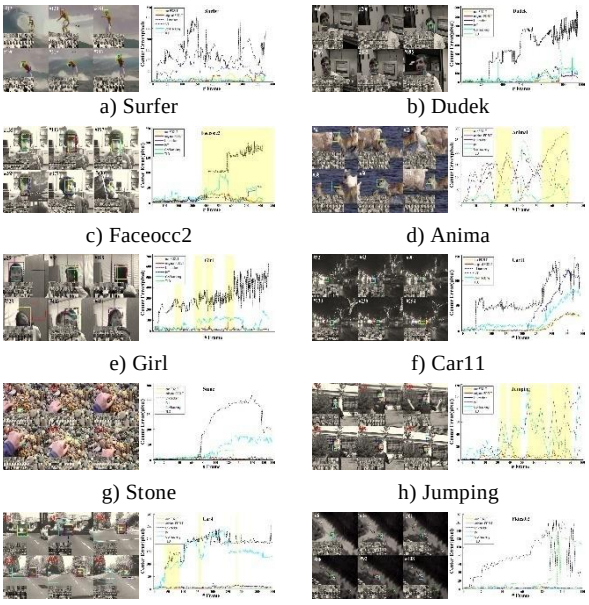
For rotation and scale variation, our trackers still work robustly, see Figures 8a, 8c and 8e. The surfer staggers forward and back in the *Surfer* sequence, the girl turns left, turns right, zoom in and zoom out in the *Girl* sequence, and the man turns left, turns right and occludes by book in the *Faceocc2* sequence, four trackers except *IVT* and *TLD* perform nice for these challenges. Especially, *TLD* loses the target in #377 in *Faceocc2*, and never recovers again.

For complex background, as shown in Figures 8f and 8g the car for tracking is driven in the dark with bright lamplight and car light affecting the tracking, and the stone for tracking scatters around lots of similar stones. *TLD* and our two trackers work well before #220 in *Car11* sequence but *IVT* loses the target from #50. And in the *Stone* sequence, *TLD*, l_1-*tracker* and our two trackers work better than other two trackers as before. *CovTrack* fail in these two sequences, because it extracts edge information of targets as one dimension of features, and in these two sequences, edge of targets are ambiguous and hard to distinct. l_1-*tracker* fails after #220 in *Car11*, because of it is short of the discriminant ability of foreground and background; On the other hand, when there exists a candidate region which is like the target for tracking, it is likely for *TLD* to detect the former instead of the latter, as shown in #50 in Figure 8d and from #85 to the end in Figure 8j. This is because of the excessive strong detective ability of *TLD*, when losing the target.

In general, from above analysis, we can find that, owing to powerful representative and discriminant capabilities, our original and improved *FD*^{2}*LT* trackers work nearly the same, and the latter is slightly better, especially in the *Faceocc2*, *Dudek*, *Girl*, *Car11*, *Stone* and

[1] Readers can download codes of *IVT*(Matlab version) and *TLD*(C++ version) on www.cs.toronto.edu/ dross/ivt/ and info.ee.surrey.ac.uk/Personal/Z.Kalal/, respectively. The other programs are coded with Matlab7.0 ourselves, and experiments are running on computer with 2.67GHz CPU and 2GB memory.

Car4 sequences, see Figures 8c, 8b, 8e, 8f, 8g, 8i; Original l_1-*tracker*[4] performances good in most frames, but fail to track sometimes; *TLD* has high performance in most of situations, which is worthy of "*long term tracker*"[12]. But it gives up tracking when facing heavily occlusion and rotation, and cannot recover when target appears again with large changes in appearance; *CovTrack* is suitable for occlusion and rotation, but fails when facing with complex backgrounds; *IVT* is sensitive, when the occlusion, rotation, motion blur of target are appeared in tracking. We also find that, our improved *FD*^{2}*LT* remains object information in updated dictionary, which are shown in button-two rows of lower-left of each frame in Figures 8a and 8b, while original *FD*^{2}*LT* destroys almost all object information, which are shown in top-two rows of lower-left of each frames. The same conclusions can be obtained when investigating the rest sequences in Figure 8. Moreover, the fixed 10^{th} object training samples also prevent the procedure of *DL* from degeneration. See #28 in Figure 8d, all six trackers lost the target, and most of the dictionary atoms after updating (used to track in the successive frame) are confused, except few atom. With our selection of training samples for *DL* in section 3.1, our two trackers retrieve the animal's head in #38, but the other two methods fail. Meanwhile, *CovTrack* also preserves the original information of target, so it recovers in #34; and *TLD* uses detection module(not tracking or learning module) to search for the object and regain it in #38. All these mean that, remaining the original information of targets, instead of updating the whole templates frame by frame, is beneficial for tracking, which is taken more and more attention in tracking community.



a) Surfer b) Dudek c) Faceocc2 d) Anima e) Girl f) Car11 g) Stone h) Jumping i) Car4 j) Pktest02

FIGURE 8 Tracking results with six tracking methods. Yellow regions means that, in these frames, trackers lose the targets

Table 1 is the tracking cost comparison of six algorithms used in our experiments. One can see that, our two *FD*^{2}*LT* frameworks work much faster(average 400 times faster under the same experimental settings described in Section 4.1) than l_1-*tracker*, because of the much smaller but well-trained dictionary; the improved *FD*^{2}*LT* (*FD*^{2}*LT*$_1$ in Table 1) is slightly fast than the original one (*FD*^{2}*LT*$_2$ in Table 1), because of the simpler optimization in Equation (5) than original one in Equation (2). *TLD*, coded with C++, is the fastest algorithm in our comparative experiments; and our improved and original *FD*^{2}*LT* ranking third and fourth in these algorithms. But our methods work much better than *IVT* (ranking second).

Table 1 The tracking cost comparison of six algorithms used in our experiments.

	Surfer	Dudek	Faceocc2	Animal	Girl	Car11	Stone	Jumping	Car4	Pktest02
IVT	**2.8694**	**3.3211**	**2.7886**	**1.8979**	1.6548	**2.7901**	1.2903	1.2682	0.8479	**1.3503**
CovTrack	1.5707	1.2454	1.1278	1.2534	1.2209	1.7041	**1.8890**	**1.4333**	1.3632	1.1906
TLD	**3.0124**	**2.5078**	**3.4763**	**4.1023**	**3.8792**	**3.8145**	**4.6451**	**3.9561**	**5.7258**	**6.7894**
l_1-*track*	0.0040	0.0016	0.0020	0.0020.5	0.0023	0.0023	0.0031	0.0029	0.0045	0.0033
FD^2*LT*$_1$	2.0886	1.8748	1.7103	1.3596	1.6909	1.7473	1.0794	.0.8927	1.1323	1.1247
FD^2*LT*$_2$	2.3469	2.3154	1.8979	1.4620	**1.7392**	2.3624	1.1501	1.1084	**1.6092**	1.1931

5 Conclusion and future works

In this paper, we analysis the reasons for the inefficiency of l_1-*tracker* and the insufficient of original *FDDL* proposed by Mei [4] and Yang [5] firstly. Then, in order to overcome these drawbacks, we present a modified version of *FDDL* and validate the numerical convergence of improved *FDDL*, then introduce the original/modified *FDDL* into video tracking, called *FD*^{2}*LT*.In our framework, three important components (object location, training samples selection, and dictionary learning) are introduced and discussed detail in Section3. Our framework combines generative tracking (i.e., *PF* [1]) with discriminant information (i.e., *FDDL*), and experiments demonstrate the effectiveness and efficiency of our trackers. But there are also some aspects required to be studied in future, including:

1) some conclusions proposed in this paper without strict proof, instead of numerical validation, e.g. theoretic convergence of *FDDL*, why improved *FD*^{2}*LT* has stronger discriminant ability than original one;

2) our *FD*^{2}*LT* framework is much faster than the latter, but it is still far away from real-time tracking(more than 20 *fps* in common video sequence). How to accelerate the tracking efficiency is one the further goals of us and even all computer vision researchers.

Acknowledgements

This work was supported by the Creative Fund on the Integration of Industry, Education and Research of Jiangsu Province(BY2013024-18).

References

[1] Yilmaz A, Javed O, Shah M 2006 *ACM Computing Surveys (CSUR)* **38**(4) 1-45
[2] Feng C, Wang Q, Wang S, Zhang W, Xu W 2011 *Image And Vision Computing* **29**(11) 787-796
[3] Cands E J, Wakin M B 2008 *Signal Processing Magazine* **25**(2) 21-30
[4] Mei X, Ling H B 2011 *IEEE Transactions on Pattern Analysis and Machine Intelligence* **33**(11) 2259-72
[5] Yang M, Zhang L, Feng X, Zhang D 2011 *IEEE 13th International Conference on Computer Vision (ICVV)* Barcelona 543-550
[6] Aharon M, Elad M, Bruckstein A 2006 *IEEE Transactions on Signal Processing* **54**(1) 4311-22
[7] Benoît L, Mairal J, Bach F, Ponce J 2011 *IEEE Conference on Computer Vision and Pattern Recognition* Colorado Springs USA 2913-20
[8] Tosic I, Frossard P 2011 *IEEE Transactions on Image Processing* **20**(4) 921-34
[9] Rosasco L, Mosci M, Santoro S, Verri A, Villa S 2009 *Technical Report MIT-CSAIL-TR-2009-050* MIT
[10] Zhang J, Wang H Y, Chen F H 2013 *Lecture Notes in Computer Science* **7751** 700-10
[11] Ross D A, Lim J, Lin R S 2008 *International Journal of Computer Vision* **77**(1-3) 125-41
[12] Kalal Z, Mikolajczyk K, Matas J 2012 *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(7) 1409-22
[13] Porikli F, Tuzel O, Meer P 2006 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR2006)* **1** 728-35

Authors

Jian-Feng Zheng, born in March, 1978, Changzhou, China

Current position, grades: lecturer at the School of Information Science & Engineering, Changzhou University.
University studies: M.S. degree in Master of Computer Applications at Nanjing University of Science and Technology, China in 2011.
Scientific interests: intelligent instruments, computer vision, image processing.
Publications: 3 Patents, 6 Papers.

Ji Zhang, born in November, 1981, Changzhou, China

Current position, grades: lecturer in School of Information Science & Engineering, Changzhou University.
University studies: M.S. degree in Control Science and Engineering at Nanjing University of Science and Technology, China in 2006.
Scientific interest: computer vision, image processing, pattern recognition.
Publications: 10 Papers.