Real Time Robust L1 Tracker Using Accelerated Proximal Gradient Approach

Chenglong Bao[1], Yi Wu[2], Haibin Ling[2], and Hui Ji[1]
[1]Department of Mathematics, National University of Singapore, Singapore,119076
[2]Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA,19122
{baochenglong,matjh}@nus.edu.sg, {wuyi,hbling}@temple.edu

Abstract

Recently sparse representation has been applied to visual tracker by modeling the target appearance using a sparse approximation over a template set, which leads to the so-called L1 trackers as it needs to solve an ℓ_1 norm related minimization problem for many times. While these L1 trackers showed impressive tracking accuracies, they are very computationally demanding and the speed bottleneck is the solver to ℓ_1 norm minimizations. This paper aims at developing an L1 tracker that not only runs in real time but also enjoys better robustness than other L1 trackers. In our proposed L1 tracker, a new ℓ_1 norm related minimization model is proposed to improve the tracking accuracy by adding an ℓ_2 norm regularization on the coefficients associated with the trivial templates. Moreover, based on the accelerated proximal gradient approach, a very fast numerical solver is developed to solve the resulting ℓ_1 norm related minimization problem with guaranteed quadratic convergence. The great running time efficiency and tracking accuracy of the proposed tracker is validated with a comprehensive evaluation involving eight challenging sequences and five alternative state-of-the-art trackers.

1. Introduction

Visual tracking has been an active research topic in the computer vision community as it is widely applied in the automatic object identification, automated surveillance, vehicle navigation and many others. Despite great progresses in last two decades, due to numerous factors in real life, many challenging problems still remain when designing a practical visual tracking system. For example, sophisticated object shape or complex motion, illumination changes and occlusions all may cause serious stability issues for a visual tracker (see a more detailed discussion in [26]).

Recently, sparse representation and compressed sensing techniques (*e.g.* [5, 7]) for finding a sparse solution of an under-determined linear system have drawn a great deal of attention in both mathematics and many applied fields, including visual tracking [15, 16, 11, 14, 24]. Similar to sparsity-based approach for face recognition developed in [22], these tracking methods express a target by a sparse linear combination of the templates in the template space, i.e., the target is well approximated by the linear combination of only a few templates. Benefitting from the stable recovery capability of sparse signal using the ℓ_1 norm minimization (*e.g.* [5]), these trackers have demonstrated good robustness in various tracking environments.

In the L1 tracker first proposed by [15], hundreds of ℓ_1 norm related minimization problems need to be solved for each frame during the tracking process. The solver for the ℓ_1 norm minimizations used in [15] is based on the interior point method which turns out to be too slow for tracking. A minimal error bounding strategy is introduced [16] to reduce the number of particles, equal to the number of the ℓ_1 norm minimizations for solving. A speed up by four to five times is reported in [16], but it is still far away from being real time. An efficient solver for the ℓ_1 norm related problems has been the key to use the L1 tracker in practice.

Moreover, in the existing L1 tracker, trivial templates are included in the template dictionary such that its sparse linear combination will present the occlusions and image noise in the target. However, as we empirically observed, the sparse linear combination of the trivial templates sometimes include parts of the object in the target, which will result in a loss of tracking accuracy in some sequences.

Built upon the same framework of the L1 tracker [15, 16], this paper aims at developing a more robust L1 tracker which runs in real time. There are two main contributions in the proposed approach. One is the introduction of a new ℓ_1 norm related minimization model which empirically showed improvements on the tracking accuracy over the model used in [15]. The other more significant contribution is the introduction of a very fast numerical method to solve the resulting ℓ_1 norm minimization problems which leads to a real time L1 tracker. It is noted that the ℓ_1 minimization problem shown in [15] is just a special case of our ℓ_1 minimization problem. Thus, the proposed numerical method can also be applied to the original L1 tracker to make it a real-time tracker.

2. Related Work

Among many approaches for real world visual tracking problem, discriminative tracking and generative tracking are two different categories with different formulations. Tracking problem is formulated as a binary classification problem in discriminative tracking methods. Discriminative trackers locate the object region by finding the best way to separate object from background; see *e.g.* [1, 2, 21, 27]. In [1], a feature vector is constructed for every pixel in the reference image and an adaptive ensemble of classifiers is trained to separate pixels that belong to the object from the ones in the background. Online multiple instance learning is used in [2] to achieve robustness to occlusions and other image corruptions. Sparse Bayesian learning is used in [21]. Global mode seeking is used in [27] to detect the object after total occlusion and reinitialize the local tracker.

Generative tracking method is based on the appearance model of target object. Tracking is done via searching target location with best matching score by some metric; see *e.g.* eigentracker [3], mean shift tracker [6], incremental tracker [18] and covariance tracker [17]. To adapt to pose and illumination changes of the object, appearance model is often dynamically updated during the tracking.

Sparse representation have been applied to tracking problem in [15], and later exploited in [14, 13]. In [15], a tracking candidate is sparsely represented by target templates and trivial templates. In [14], group sparsity is integrated and very high dimensional image features are used for improving tracking robustness. In these approaches, the sparse representation is obtained via solving a ℓ_1-norm related minimization problem [15] or ℓ_0-norm related minimization in [14, 13]. It is well known that ℓ_0-norm related minimization is an NP-hard problem. The large-scale ℓ_1-norm related minimization is also a challenging problem due to the non-differentiability of ℓ_1 norm. The numerical methods for solving ℓ_1-norm related minimization in [15] is based on the interior point method [10], which is very slow when solving large-scale ℓ_1-norm minimizations.

In recent years, there have been great progresses on fast numerical methods for solving large-scale ℓ_1-norm related minimization problems arising in image science, such as Linearized Bregman iteration [4], Split Bregman method [8] etc. Meanwhile, Yang *et al.* [25] has done a comprehensive study of the ℓ_1 norm related minimization on robust face recognition. Among all these methods, one promising approach is the so-called accelerated proximal gradient (APG) method introduced by [20] for minimizing the summation of one smooth function and one non-differential function. The APG method is used in [19] to solve a unconstrained ℓ_1 norm related problem related to image restoration.

3. Introduction to L1 Tracker

Our tracker is closely related to the L1 tracker proposed by Mei and Ling [15]. The main differences lie in a different minimization model and a much faster numerical solver for the resulting ℓ_1 norm minimization problems. We first give a brief review on the L1 tracker within the particle filter framework proposed in [16, 15].

Particle Filter: The particle filter provides an estimate of posterior distribution of random variables related to Markov chain. In visual tracking, it gives an important tool for estimating the target of next frame without knowing the concrete observation probability. It consists of two steps: prediction and update. Specially, at the frame t, denote $\mathbf{x}_t$ which describes the location and the shape of the target, $\mathbf{y}_{1:t-1} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_{t-1}\}$ denotes the observation of the target from the first frame to the frame $t-1$. Particle filter proceeds two steps with following two probabilities:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1}, \quad (1)$$

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})}. \quad (2)$$

The optimal state for the frame t is obtained according to the maximal approximate posterior probability: $\mathbf{x}_t^* = \arg\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{y}_{1:t})$.

The posterior probability (2) is approximated by using finite samples $\mathbf{S}_t = \{\mathbf{x}_t^1, \mathbf{x}_t^2, \cdots, \mathbf{x}_t^N\}$ with different weights $\mathrm{W} = \{\mathbf{w}_t^1, \mathbf{w}_t^2, \cdots, \mathbf{w}_t^N\}$ where N is the number of samples. The samples are generated by sequential importance distribution $\mathbf{\Pi}(\mathbf{x_t}|\mathbf{y_{1:t}}, \mathbf{x_{1:t-1}})$ and weights are updated by:

$$\mathbf{w}_t^i \propto \mathbf{w}_{t-1}^i \frac{p(\mathbf{y}_t|\mathbf{x}_t^i)p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)}{\mathbf{\Pi}(\mathbf{x_t}|\mathbf{y_{1:t}}, \mathbf{x_{1:t-1}})}. \quad (3)$$

In the case of $\mathbf{\Pi}(\mathbf{x_t}|\mathbf{y_{1:t}}, \mathbf{x_{1:t-1}}) = \mathbf{p}(\mathbf{x_t}|\mathbf{x_{t-1}})$, the equation (3) has a simple form $\mathbf{w}_t^i \propto \mathbf{w}_{t-1}^i p(\mathbf{y}_t|\mathbf{x}_t^i)$. Then, the weights of some particles maybe keep increasing and fall into the degeneracy case. To avoid such a case, in each step, samples are re-sampled to generate new sample set with equal weights according to their weights distribution.

Sparse Representation: The sparse representation model aims at calculating the observation likelihood for sample state $\mathbf{x}_t$, i.e. $p(\mathbf{z}_t|\mathbf{x}_t)$. At the frame t, given the target template set $\mathbf{T}_t = [\mathbf{t}_t^1, \mathbf{t}_t^2, \cdots, \mathbf{t}_t^n]$, let $\mathbf{S}_t = \{\mathbf{x}_t^1, \mathbf{x}_t^2, \cdots, \mathbf{x}_t^N\}$ denote the sampled states and let $\mathbf{O}_t = \{\mathbf{y}_t^1, \mathbf{y}_t^2, \cdots, \mathbf{y}_t^N\}$ denote the corresponding candidate target patch in target template space. The sparse representation model is then:

$$\mathbf{y}_t^i = \mathbf{T}_t\mathbf{a}_T^i + I\mathbf{a}_I^i, \quad \forall \mathbf{y}_t^i \in \mathbf{O}_t, \quad (4)$$

where I is the trivial template set (identity matrix) and $\mathbf{a}_t^i = [\mathbf{a}_T^i; \mathbf{a}_I^i]$ is sparse. Additionally, nonnegative constraints are imposed on $\mathbf{a}_T^i$ for the robustness of the L1 tracker [15]. Consequently, for each candidate target patch $\mathbf{y}_t^i$, the sparse representation of $\mathbf{y}_t^i$ can be found via solving the following ℓ_1-norm related minimization with nonnegative constraints:

$$\min_{\mathbf{a}} \frac{1}{2}\|\mathbf{y}_t^i - A\mathbf{a}\|_2^2 + \lambda\|\mathbf{a}\|_1,\ \mathbf{a} \succcurlyeq 0, \tag{5}$$

where $A = [\mathbf{T}_t, I, -I]$.

Finally, the observation likelihood of state $\mathbf{x}_t^i$ is given as

$$p(\mathbf{z}_t|\mathbf{x}_t^i) = \frac{1}{\Gamma}\exp\{-\alpha\|\mathbf{y}_t^i - \mathbf{T}_t\mathbf{c}_T^i\|_2^2\}, \tag{6}$$

where α is a constant controlling the shape of the Gaussian kernel, Γ is a normal factor and $\mathbf{c}_T^i$ is the minimizer of (5) restricted to $\mathbf{T}_t$. Then, the optimal state $\mathbf{x}_t^*$ of frame t is obtained by

$$\mathbf{x}_t^* = \arg\max_{\mathbf{x}_t^i \in \mathbf{S}_t} p(\mathbf{z}_t|\mathbf{x}_t^i). \tag{7}$$

In addition, a template update scheme is adopted in [15] to overcome pose and illumination changes.

Minimal Error Bound: In [15], the ℓ_1-norm related minimization problem (5) is solved by the interior point method which is very slow. A minimal error bounding method is then proposed in [16] to reduce the number of needed ℓ_1 minimizations. Actually, their method is based on the following observation:

$$\|\mathbf{T}_t\mathbf{a} - \mathbf{y}\|_2^2 \geq \|\mathbf{T}_t\hat{\mathbf{a}} - \mathbf{y}\|_2^2, \quad \forall \mathbf{a} \in \mathbb{R}^N, \tag{8}$$

where

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} \|\mathbf{T}_t\mathbf{a} - \mathbf{y}\|_2^2. \tag{9}$$

Consequently, for any samples $\mathbf{x}_t^i$, its observation likelihood has the following upper bound:

$$p(\mathbf{z}_t|\mathbf{x}_t^i) \leq \frac{1}{\Gamma}\exp\{-\alpha\|\mathbf{T}_t\hat{\mathbf{a}} - \mathbf{y}_t^i\|_2^2\} \triangleq q(\mathbf{z}_t|\mathbf{x}_t^i), \tag{10}$$

where $q(\mathbf{y}_t^i|\mathbf{x}_t^i)$ is the probability upper bound for state $\mathbf{x}_t^i$. It is seen that if $q(\mathbf{z}_t|\mathbf{x}_t) < \frac{1}{2N}\sum_{j=1}^{i-1} p(\mathbf{z}_t|\mathbf{x}_t^j)$, then the sample $\mathbf{x}_t^i$ will not appear in the resample set. In other words, $\mathbf{x}_t^i$ can be discarded without being processed. Thus, a two-stage resample method is proposed in [16] to significantly reduce the number of samples needed in tracking.

4. Real Time L1 Tracker

Even though the minimal error bound [16] was proposed to reduce the computation load for L1 tracker, there are still many ℓ_1-norm related minimizations for solving during the tracking process, For example, in the sequence *car* with 620 frames, around 80,000 ℓ_1-norm related minimizations (5) needs to be solved with minimal error bound resampling scheme in [16]. Therefore, the speed bottleneck in the L1 tracker is how to solve the ℓ_1-norm related minimization (5) much faster, in the scale of hundreds of times.

Also, as seen in the model (5), the trivial templates are included in the template dictionary such that its sparse linear combination will represent the occlusions and image noise in the target. However, as we observed in the experiments, the sparse linear combination will sometimes include parts of the object in the target which may lead to a loss of tracking accuracy in some sequences.

In this section, we first proposed a modified version of the minimization problem (5) such that the sparse linear combination of trivial templates can represent the occlusions and image noise more accurately. Then, based on the accelerated proximal gradient approach [20], we proposed a fast numerical method for solving the resulting ℓ_1 norm related minimization problem such that the tracker runs in real time. It is noticed that the developed method is also applicable to original minimization problem in (5).

4.1. A modified ℓ_1 norm related minimization model

There are two types of templates in the template dictionary used by (5): target templates and trivial templates. The target templates are updated dynamically for representing target objects during the tracking process. The trivial templates (identity matrix I) is for representing occlusions, background and noise. However, since parts of objects may also be represented by the trivial templates, the region detected by the original tracker sometimes does not fit the target very accurately.

We take a modified version of (5) for improving tracking accuracy. The new model is based on the following observation. When there are no occlusions, the target in the next frame should be well approximated by a sparse linear combination of target templates with a small residual. Thus, the energy of the coefficients in **a** associate with trivial templates, named *trivial coefficients*, should be small. On the other hand, when there exist noticeable occlusions, the target in the next frame cannot be well approximation by any sparse linear combination of target templates, the large residual (corresponding to occlusions, background and noise in an ideal situation) will be compensated by the part from the trivial templates, which leads to a large energy of the trivial coefficients. The minimization (5) is obviously not optimal since it does not differentiate these two cases.

In other words, to optimize the usage of the trivial templates in the tracking, we need to adaptively control the energy of the trivial coefficients. That is, when occlusions are negligible, the energy associated with trivial templates should be small. When there are noticeable occlusions, the energy should be allowed to be large. This motivation leads

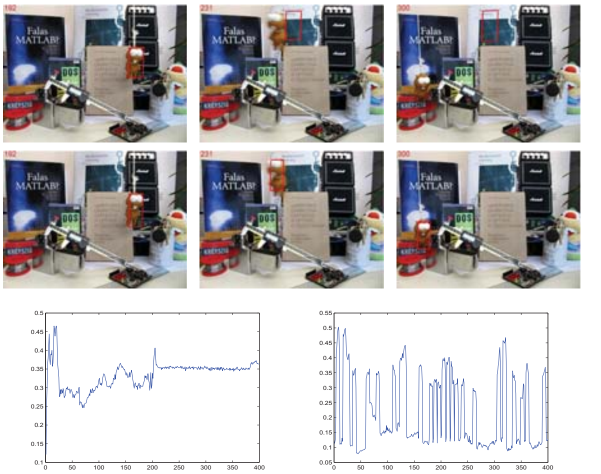

Figure 1. Illustration of the L1 tracker on the sequence *lemming* using the model (5) and the L1 tracker using the proposed model (11). The first and the second row: results using (5) and using (11) respectively. Last row: the energy ratio $\|\mathbf{a}_I\|_2/\|\mathbf{a}\|_2$. The left graph is from (5) and the right is from (11).

to the following minimization model for L1 tracker

$$\min_{\mathbf{a}} \frac{1}{2}\|\mathbf{y}-A'\mathbf{a}\|_2^2+\lambda\|\mathbf{a}\|_1+\frac{\mu_t}{2}\|\mathbf{a}_I\|_2^2, \quad \text{s.t. } \mathbf{a}_T \succcurlyeq 0, \quad (11)$$

where $A' = [\mathbf{T}_t, I]$, $\mathbf{a} = [\mathbf{a}_T; \mathbf{a}_I]$ are the coefficients associated with target templates and trivial templates respectively, and the parameter μ_t is a parameter to control the energy in trivial templates. In our implementation, the value of μ_t for each state is automatically adjusted using the occlusion detection method [16]. That is, if occlusions are detected, $\mu_t = 0$; otherwise μ_t is set as some pre-defined constant.

The benefit of the additional ℓ_2 norm regularization term $\|\mathbf{a}_I\|_2^2$ is illustrated in Fig. 1. In Fig. 1, about 30 percent of object energy is contained in trivial templates from minimization (5). In other words, trivial templates can not distinguish the object and background. On the other hand, we can see the trivial templates coefficients from minimization (11) are small and lead to better tracking results. At last, we note that the original minimization (5) is a special case of the minimization (11) by setting $\mu_t = 0$.

4.2. Fast numerical method for solving (11)

The proposed method for solving the minimization problem (11) is based on the accelerated proximal gradient (APG) approach [20].

APG approach. The APG method is originally designed for solving the following unconstrained minimization:

$$\min F(\mathbf{a}) + G(\mathbf{a}), \quad (12)$$

where $F(\mathbf{a})$ is a differentiable convex function with Lipschitz continuous gradient[1] and $G(\mathbf{a})$ is a non-smooth but convex function. The outline of the APG method is given in Algorithm 1. The efficiency of the APG method is justified by its quadratic convergence; see Theorem 4.1. However, we emphasize here that the APG method is fast only for particular type of function G. During each iteration of Algorithm 1, we need to solve a minimization in Step 2. So, the quadratic convergence of APG is materialized only when the sub-problem in Step 2 has an analytic solution.

Theorem 4.1 ([20]) *Let $\{\alpha_k\}$ is the sequence generated by Algorithm 1. Then within $K = O(\sqrt{L/\epsilon})$ iterations, $\{\alpha_k\}$ achieves ϵ-optimality such that $\|\alpha_K - \alpha^*\| < \epsilon$, where α^* is one minimizer of* (12).

Algorithm 1 the generic APG approach in [20]

```
(i) Set $\alpha_0 = \alpha_{-1} = \mathbf{0} \in \mathbb{R}^N$ and set $t_0 = t_{-1} = 1$.
(ii) For $k = 0, 1, \ldots$, iterate until convergence
```

$$\begin{cases} \beta_{k+1} := \alpha_k + \frac{t_{k-1}-1}{t_k}(\alpha_k - \alpha_{k-1}); \\ \alpha_{k+1} := \arg\min\limits_{\mathbf{a}} \frac{L}{2}\|\mathbf{a} - \beta_{k+1} + \frac{\nabla F(\beta_{k+1})}{L}\|_2^2 + G(\mathbf{a}); \\ t_{k+1} := \frac{1+\sqrt{1+4t_k^2}}{2}. \end{cases} \quad (13)$$

Reformulation of (11) **for applying APG method**. As we see, the original APG method is designed for unconstrained minimization problem which can not be directly applied to (11). Thus, we need to convert the constrained minimization model into an unconstrained problem. Let $\mathbf{1} \in \mathbb{R}^N$ denote the vector with all entries are equal to 1 and let $\mathbf{1}_{\mathbb{R}_+^N}(\mathbf{a})$ denote the indicator function defined by

$$\mathbf{1}_{\mathbb{R}_+^N}(\mathbf{a}) = \begin{cases} 0, & \mathbf{a} \succeq 0; \\ +\infty, & \text{otherwise}. \end{cases} \quad (14)$$

It is easy to see that the minimization (11) is equivalent to the following minimization problem:

$$\arg\min_{\mathbf{a}} \frac{1}{2}\|\mathbf{y}-A'\mathbf{a}\|_2^2+\lambda\mathbf{1}_T^\top\mathbf{a}_T+\|\mathbf{a}_I\|_1+\frac{\mu_t}{2}\|\mathbf{a}_I\|_2^2+\mathbf{1}_{\mathbb{R}_+^n}(\mathbf{a}_T). \quad (15)$$

Then, the APG method can be applied to (15) with

$$\begin{aligned} F(\mathbf{a}) &= \frac{1}{2}\|\mathbf{y} - A'\mathbf{a}\|_2^2 + \lambda\mathbf{1}_T^\top\mathbf{a}_T + \frac{\mu_t}{2}\|\mathbf{a}_I\|_2^2, \\ G(\mathbf{a}) &= \|\mathbf{a}_I\|_1 + \mathbf{1}_{\mathbb{R}_+^n}(\mathbf{a}_T). \end{aligned} \quad (16)$$

All steps in Algorithm 1 are trivial except Step 2, in which we need to solve an optimization problem:

$$\alpha_{k+1} = \arg\min_{\mathbf{a}} \frac{L}{2}\|\mathbf{a} - \beta_{k+1} + \frac{\nabla F(\beta_{k+1})}{L}\|_2^2 + G(\mathbf{a}). \quad (17)$$

[1] the gradient of F is Lipschitz continuous if $\|\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^N$, for some constant L.

For general function G, it cannot be directly solved. However, in our setting, we have the analytic solution for (17); see Proposition 4.2. The algorithm for solving ℓ_1-norm related minimization (11) is given in Algorithm 2.

Proposition 4.2 *If $F(\boldsymbol{a})$ and $G(\boldsymbol{a})$ are defined in* (16)*, then the minimization problem* (17) *has the following solution:*

$$\begin{aligned} \alpha_{k+1}|_T &= \max(0, g_{k+1}|_T) \\ \alpha_{k+1}|_I &= \mathfrak{T}_{\lambda/L}(g_{k+1}|_I). \end{aligned} \tag{18}$$

where $g_{k+1} = \beta_{k+1} - \frac{\nabla F(\beta_{k+1})}{L}$ and $\mathfrak{T}$ is the soft-thresholding operator: $\mathfrak{T}_\lambda(x) = \operatorname{sign}(x)\max(|x| - \lambda, 0)$.

Proof See the appendix A.

Algorithm 2 Real Time Numerical algorithm for solving the minimization (11)

```
(i) Set $\alpha_0 = \alpha_{-1} = \mathbf{0} \in \mathbb{R}^N$ and set $t_0 = t_{-1} = 1$.
(ii) For $k = 0, 1, \ldots$, iterate until convergence
```

$$\begin{cases} \beta_{k+1} := \alpha_k + \frac{t_{k-1}-1}{t_k}(\alpha_k - \alpha_{k-1}); \\ g_{k+1}|_T := \beta_{k+1}|_T - (A'^\top(A'\beta_{k+1} - \mathbf{y}))|_T/L - \lambda\mathbf{1}_T; \\ g_{k+1}|_I := \beta_{k+1}|_I - (A'^\top(A\beta_{k+1} - \mathbf{y}))|_I/L \\ \qquad -\mu\beta_{k+1}|_I/L; \\ \alpha_{k+1}|_T := \max(0, g_{k+1}|_T); \\ \alpha_{k+1}|_I := \mathfrak{T}_{\lambda/L}(g_{k+1}|_I); \\ t_{k+1} := (1 + \sqrt{1 + 4t_k^2})/2. \end{cases}$$

Tight Lipschitz constant L estimation. There is only one parameter, the Lipschitz constant L of ∇F, is involved in Algorithm 2. This Lipschitz constant L plays a crucial role in the above algorithm. Algorithm 2 with an wrong L will either diverges or converges very slowly. Next, we give a tight upper bound of L for F defined in (16) such that L is automatically set with optimal performance; see Proposition 4.3. The detailed description of the proposed real time L1 tracker, called *APG-L1* tracker, is given in algorithm 3.

Proposition 4.3 *Let F denote the function defined in* (16) *with $A' = [T, I]$, where T is template set and I is the identity matrix. The upper bound of the Lipschitz constant L for ∇F is given as follows.*

$$L \le \lambda_{max}^2 + \mu_t + 1, \tag{19}$$

where λ_{max} is the largest singular value of T.

Proof See Appendix B.

5. Experiments

Through the experiments, APG algorithm is implemented with Matlab, $\mu_t = 5$ in (11) when the occlusion is not detected and 0 otherwise, and $\lambda = 10^{-2}, T = 8$ in Algorithm 2.

Algorithm 3 APG-L1 Tracker

```
Input:
Current frame $F_t$;
Sample Set $\mathbf{S}_{t-1} = \{\mathbf{x}_{t-1}^i\}_{i=1}^N$;
Template set $\mathbf{T} = \{\mathbf{t}_i\}_{i=1}^n$.
for $i = 1$ to $N$ do
    Drawing the new sample $\mathbf{x}_t^i$ from $\mathbf{x}_{t-1}^i$;
    Preparing the candidate patch $\mathbf{y}_t^i$ in template space;
    Solving the least square problem (9);
    Computing $q_i$ according to (10);
end for
Sorting the samples in descent order according to $q$;
Setting $i = 1$ and $\tau = 0$.
while $i < N$ and $q_i \ge \tau$ do
    Solving the minimization (11) via Algorithm 2;
    Computing the observation likelihood $p_i$ in (6);
    $\tau = \tau + \frac{1}{2N}p_i$;
    $i = i + 1$;
end while
Set $p_j = 0, \forall j \ge i$.
Output:
Finding the $\mathbf{x}_t^*$ according to (7);
Detecting the occlusion [16] and update $\mu$ in (11);
Updating the template set $\mathbf{T}_{t-1}$ [16];
Updating the sample set $\mathbf{S}_{t-1}$ with $p$.
```

5.1. Comparison with the L1 Tracker [16]

The computation efficiency and tracking accuracy of the proposed APG-L1 tracker is first compared to that of the BPR-L1 tracker [16] on ten sequences. The average running time of the proposed APG-based solver v.s. the interior point method used [16] is about $1:150$. As a result, the average running time of the APG-L1 tracker v.s. the BPR-L1 tracker is around 1:20, with 600 particles. The APG-L1 tracker achieves about average 26 frames per second with 600 particles on a PC with Intel i7-2600 CPU (3.4GHz). The output bounding boxes of the target from the two tackers are similar in many sequences, while the results from APG-L1 are more accurate on some challenging sequences.

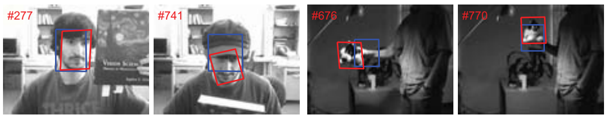

Figure 2. Demonstration of the improvement of APG-L1 tracker (red) over BPR-L1 (blue) on tracking accuracy.

5.2. Qualitative Comparison with Other Methods

The performance of the proposed APG-L1 tracker is also evaluated on eight publicly available video sequences and is compared with five latest state-of-the-art trackers named Incremental Visual Tracking (IVT) [18], Multiple Instance Learning (MIL) [2], Visual Tracking Decomposition (VTD) [12], Incremental Covariance Tensor Learning (ICTL) [23], and Online AdaBoost (OAB) [9]. The tracking results of the compared methods were obtained using the codes provided by the authors with the default parameters and using the same initial positions in the first frame.

The sequence *jump* was captured outdoors. The target was jumping and the motion blurs are very severe. Results on several frames are presented in Fig. 3 (a). The APG-L1 tracker, IVT, OAB, and MIL tracks the target faithfully throughout the sequences. The other trackers fails track the target when there are abrupt motion and severe motion blur.

The sequence *car* shows a vehicle undergoes drastic illumination changes as it passes beneath a bridge and under trees. Tracking results on several frames are shown in Fig. 3 (b). The APG-L1 tracker and IVT can track the target well despite the drastic illumination changes, while the other trackers lose the target after it goes through the bridge.

Results of the sequence *singer* are shown in Fig. 3 (c). In this sequence, we show the robustness of our algorithm in severe illumination changes and large scale variations. Only our APG-L1 tracker and the VTD tracker can track the target throughout the sequence.

In the sequence *woman* (Fig. 3 (d)), only the APG-L1 tracker is able to track the target during the entire sequence. The other trackers drift to the man when he occludes the target due to his similar appearance as the target.

In the sequence *pole*, a person is walking away from the camera and is occluded by the pole for a short time (Fig. 3 (e)). The IVT loses the target from the start and the VTD starts to drift off the target at frame 274 and finally loses the target. All the rest successfully track the target but our APG-L1 tracker recovers the target scale better.

Results on the sequence *sylv* are shown in Fig. 3 (f), where a moving animal doll is undergoing challenging pose variations, lighting changes and scale variations. The IVT, and VTD eventually fails at frame 605 as a result of drastic pose and illumination changes. The rest trackers are able to track the target for this long sequence while our APG-L1 tracker performs with higher accuracy.

Results of the sequence *deer* are shown in Fig. 3 (g). In this sequence, we show the robustness of our algorithm in background clutters and the fast motion. Only our APG-L1 tracker and VTD can track the target through the sequence.

Fig. 3 (h) shows the results on the sequence *face*. Many trackers start drifting from the target when the man's face is severely occluded by the book. The APG-L1 tracker and IVT handle this very well and continue tracking the target when the occlusion disappears.

	MIL	OAB	ICTL	VTD	IVT	ours
jump	0.030	0.030	0.198	0.221	**0.020**	0.025
car	0.749	0.786	0.326	0.313	0.049	**0.048**
singer	0.299	0.466	0.503	**0.056**	0.155	0.069
woman	0.361	0.179	0.323	0.339	0.148	**0.032**
pole	0.007	0.010	0.008	0.049	0.572	**0.003**
sylv	0.069	0.058	0.096	0.203	0.197	**0.032**
deer	0.022	0.060	0.306	0.027	0.110	**0.017**
face	0.120	0.144	0.137	0.209	**0.053**	0.062
Ave.	0.207	0.217	0.237	0.177	0.163	**0.036**

Table 1. The average tracking errors. The error is measured using the Euclidian distance of two center points, which has been normalized by the size of the target from the ground truth. The last row is the average error for each tracker over all the test sequences.

5.3. Quantitative Comparison with other methods

To quantitatively evaluate the robustness of the APG-L1 tracker under challenging conditions, we manually annotated the target's bounding box in each frame for all test sequences. The tracking error evaluation is based on the relative position errors (in pixels) between the center of the tracking result and that of the annotation. As shown in Fig.4 and Table 1, the APG-L1 tracker achieves comparable to the best performer on the sequence *jump*, *singer* and *face* to the best-performed trackers, and on all the other sequences it performs best.

6. Conclusion

In summary, based on the framework of the L1 tracker [15, 16], we developed a real time L1 visual tracker with improved tracking accuracy. The accuracy improvement is achieved via a new minimization model for finding the sparse representation of the target and the real time performance is achieved by a new APG based numerical solver for the resulting ℓ_1 norm minimization problems. The experiments also validated the high computational efficiency and better tracking accuracy of the proposed APG-L1 tracker.

References

[1] S. Avidan. Ensemble tracking. In *CVPR*, 2005.

[2] B. Babenko, M. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, 2009.

[3] M. Black and A. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *IJCV*, 26(1):63–84, 1998.

[4] J. Cai, S. Osher, and Z. Shen. Linearized bregman iterations for compressed sensing. *Math. Comp*, 78:55–59, 2009.

[5] E. Candes, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Commu. on pure and applied mathematics*, 59(8):1207–1223, 2006.

[6] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *PAMI*, 25(5):564–577, 2003.

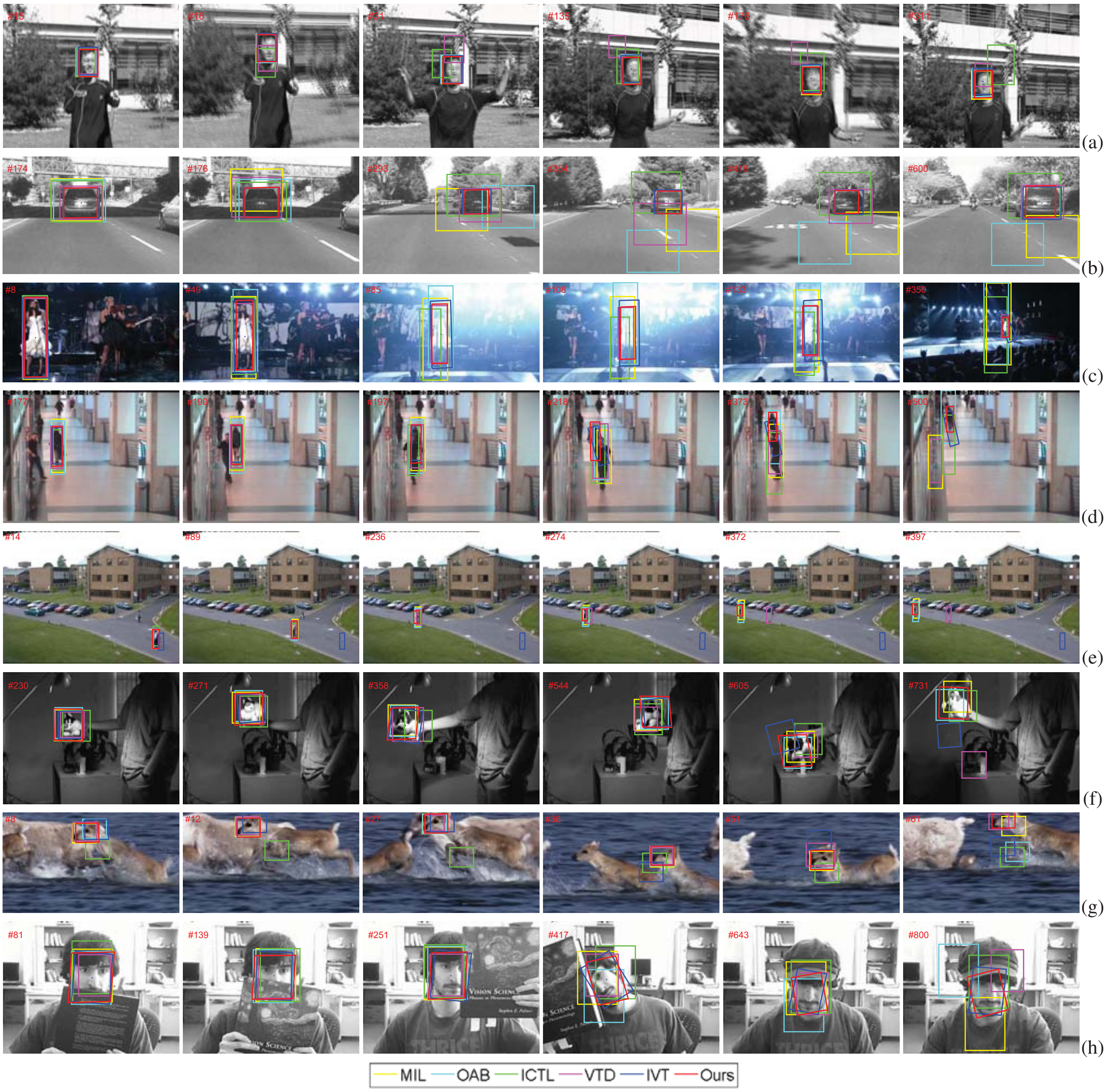

Figure 3. Tracking results of different algorithms for sequences *jump*(a), *car*(b), *singer*(c), *woman*(d), *pole*(e), *sylv*(f), *deer*(g) and *face*(h).

[7] D. Donoho. Compressed sensing. *IEEE Trans. on Information Theory*, 52(4):1289–1306, 2006.

[8] T. Goldstein and S. Osher. The split bregman method for l1 regularized problems. *SIAM J Imag. Sci.*, 2:323–343, 2009.

[9] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via online boosting. In *BMVC*, 2006.

[10] S. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale l1-regularized least squares. *IEEE J Sel Topics in Sig. Proc.*, 1:606–617, 2007.

[11] S. Kwak, W. Nam, B. Han, J.H. Han. Learning Occlusion with Likelihoods for Visual Tracking. In *CVPR*, 2011.

[12] J. Kwon and K. Lee. Visual tracking decomposition. In *CVPR*, 2010.

[13] H. Li, C. Shen, and Q. Shi. Real-time Visual Tracking Using Compressive Sensing. In *CVPR*, 2011.

[14] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. Kulikowski. Robust and fast collaborative tracking with two stage sparse optimization. In *ECCV*, 2010.

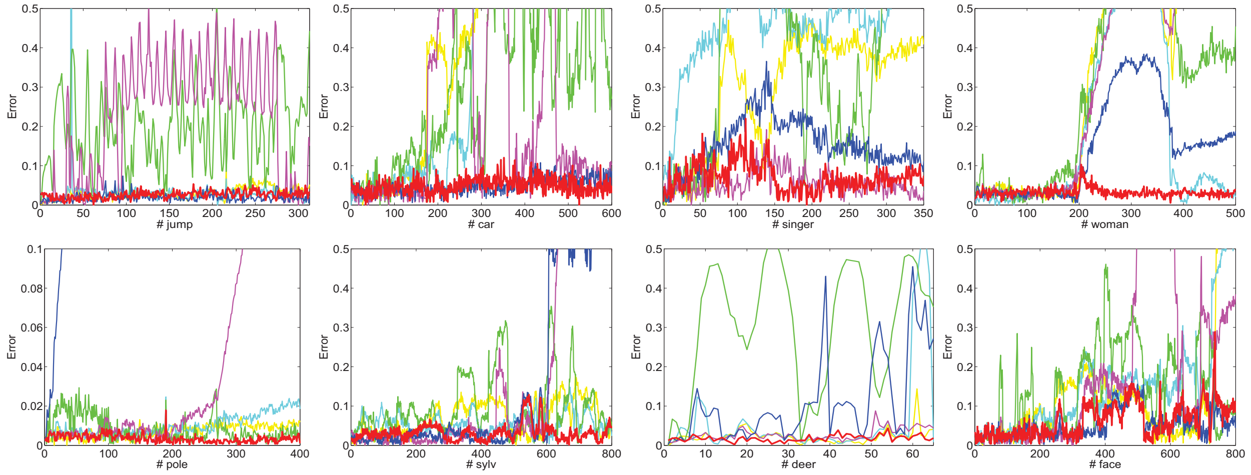

Figure 4. The tracking error for each test sequence. The error is measured the same as in Table 1 and the legend as in Fig.3.

[15] X. Mei and H. Ling. Robust visual tracking using l1 minimization. In *ICCV*, 2009.

[16] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai. Minimum error bounded efficient L1 tracker with occlusion detection. In *CVPR*, 2011.

[17] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on lie algebra. In *CVPR*, 2006.

[18] D. Ross, J. Lim, R. Lin, and M. Yang. Incremental learning for robust visual tracking. *IJCV*, 77(1):125–141, 2008.

[19] Z. Shen, K. Toh, and S. Yun. An accelerated proximal gradient algorithm for frame based image restorations via the balanced approach. *SIAM J on Imag. Sci.*, 4:573, 2011.

[20] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. *SIAM J on Opti.*, 2008.

[21] O. Williams, A. Blake, and R. Cipolla. Sparse bayesian learning for efficient visual tracking. *PAMI*, 27(8):1292–1304, 2005.

[22] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *PAMI*, 31(1):210-227, 2009.

[23] Y. Wu, J. Cheng, J. Wang, H. Lu, J. Wang, H. Ling, E. Blasch, and L. Bai. Real-time Probabilistic Covariance Tracking with Efficient Model Update. *IEEE T-IP*, 2012.

[24] Y. Wu, H. Ling, J. Yu, F. Li, X. Mei, and E. Cheng. Blurred Target Tracking by Blur-driven Tracker. In *ICCV*, 2011.

[25] A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Fast l1-minimization algorithms and an application in robust face recognition: a review. In *ICIP*, 2010.

[26] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *Acm Computing Surveys*, 38(4):13, 2006.

[27] Z. Yin and R. Collins. Object tracking and detection after occlusion via numerical hybrid local and global mode-seeking. In *CVPR*, 2008.

Appendix A: Proof of Proposition 4.2.

The optimization problem (17) is expressed as follows,

$$\min_{\mathbf{a}} \frac{L}{2}\|\mathbf{a} - g_{k+1}\|_2^2 + \mathbf{1}_{\mathbb{R}_n^+}(\mathbf{a}_T) + \|\mathbf{a}_I\|_1. \tag{20}$$

Since the variables of **a** are independent, (20) is the same as

$$\begin{aligned} &\min_{\mathbf{a}_T} \frac{L}{2}\|\mathbf{a}_T - g_{k+1}|_T\|_2^2 + \mathbf{1}_{\mathbb{R}_n^+}(\mathbf{a}_T), \\ &\min_{\mathbf{a}_I} \frac{L}{2}\|\mathbf{a}_I - g_{k+1}|_I\|_2^2 + \lambda\|\mathbf{a}_I\|_1. \end{aligned} \tag{21}$$

It is easy to see the solution of first minimization in (21) is the projection of $g_{k+1}|_T$ to the $\mathbb{R}_n^+$ space, i.e. $\max(0, g_{k+1}|_T)$. For the second minimization in (21), all the variables are independent. So, we only need to solve the following minimization :

$$\min_{x} \frac{L}{2}\|y - x\|_2^2 + \lambda\|x\|_1 \triangleq f(x), \tag{22}$$

where $x, y \in \mathbb{R}$. The minimizer of (22) can be expressed as a soft thresholding operation:

$$x = \mathfrak{T}_{\lambda/L}(y) = \operatorname{sgn}(y) * \max(|y| - L, 0). \tag{23}$$

Thus, we have $\mathbf{a}_I = \mathfrak{T}_{\lambda/L}(g_{k+1}|_I)$ as the minimizer of (21).

Appendix B: Proof of Proposition 4.3.

From (16), we have

$$\nabla^2 F(x) = \begin{pmatrix} T^\top T & T^\top \\ T & (1+\mu)I \end{pmatrix} \tag{24}$$

Assume $T = U\Sigma V^\top$ by singular value decomposition, where U and V are orthonormal matrices, $\Sigma \in \mathbb{R}^{m\times N}(m < N)$ with $\Sigma_{ii} = \lambda_i$ and $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m \geq 0$. It is easy to know $\nabla^2 F(x)$ is similar to $M \triangleq \begin{pmatrix} \Sigma^\top\Sigma & \Sigma^\top \\ \Sigma & (1+\mu)I \end{pmatrix}$. So $\lambda_{Fmax} = \lambda_{Mmax} \leq \lambda_{max}^2 + 1 + \mu$, where λ_{Fmax}, λ_{Mmax} and λ_{max} are the largest singular values of $\nabla^2 F(x)$, M and T respectively.