L_0-Regularized Object Representation for Visual Tracking

Jinshan Pan[1]
sdluran@gmail.com
Jongwoo Lim[2]
jlim@hanyang.ac.kr
Zhixun Su[1]
zxsu@dlut.edu.cn
Ming-Hsuan Yang[3]
mhyang@ucmerced.edu

[1] School of Mathematical Sciences
Dalian University of Technology
Dalian, China

[2] Division of Computer Science & Engineering
Hanyang University
Seoul, Korea

[3] Electrical Engineering and Computer Science
University of California at Merced
California, USA

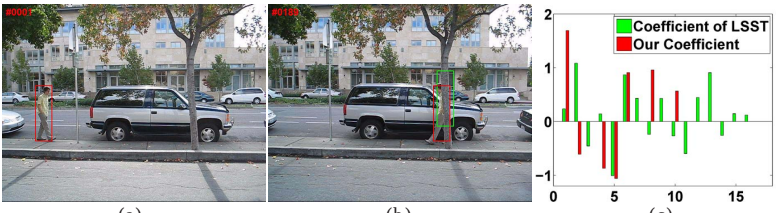
Abstract

In this paper, we propose a robust visual tracking method by L_0-regularized prior in a particle filter framework. In contrast to existing methods, the proposed method employs L_0 norm to regularize the linear coefficients of incrementally updated linear basis. The sparsity constraint enables the tracker to effectively handle difficult challenges, such as occlusion or image corruption. To achieve realtime processing, we propose a fast and efficient numerical algorithm for solving the proposed L_0-regularized model. Although it is an NP-hard problem, the proposed accelerated proximal gradient (APG) approach is guaranteed to converge to a solution quickly. Extensive experimental results on challenging video sequences demonstrate that the proposed method achieves state-of-the-art results both in accuracy and speed.

1 Introduction

Visual tracking is a highly researched topic in the computer vision community since it has been widely applied in visual surveillance, driver assistant system, and many others. Although much progress has been made in the past decades, designing a practical visual tracking system is still a challenging problem due to numerous challenges in real world. For example, pose variation, shape deformation, varying illumination, camera motion, and occlusions may increase the difficulty for visual tracking algorithms.

Recently, sparse representation and compressed sensing techniques [7] have been successfully applied to visual tracking [17, 19, 20]. In this case, the tracker represents each target candidate as a sparse linear combination of dictionary templates that can be dynamically updated to maintain an up-to-date target appearance model. This representation has been shown to be robust against partial occlusions, which leads to improved tracking performance. However, heavy computational overhead in L_1 minimization hampers the tracking speed. Very recent efforts have been made to improve this method in terms of both speed and accuracy by using APG algorithm [4] or modeling the similarity between different candidates [32]. The works in [24, 25] point out that the aforementioned methods do not exploit



(a) (b) (c)

Figure 1: The influence of sparse coefficients in visual tracking. (a) In the first frame, the target is specified with the red rectangle. (b) The tracking results in frame #189 show that the proposed algorithm (the red rectangle) handles occlusion better than LSST [24] (the green rectangle). (c) Using the cropped windows of the frame #189, the estimated coefficients by the proposed algorithm are sparser than those by LSST.

rich and redundant image properties which can be captured compactly with subspace representations. Thus, they propose combining the strength of subspace learning [22] and sparse representation for modeling object appearance. In their work the object templates used in in [4, 19] are replaced with the orthogonal basis vectors (*e.g.*, PCA basis), and the coefficients for an image are obtained by least square (LS) method. However, we empirically find that such linear combination of the orthogonal basis vectors sometimes include redundant parts (*e.g.*, background portions), which will interfere with the accuracy of object representation. Figure 1 demonstrates this observation. As shown in Figure 1 (c), one can see that the coefficient of [24] is actually not sparse and the target object is not tracked well. In comparison, the results with sparsity coefficient perform better.

Based on the above observation, we in this paper address this problem by proposing a tracking method based on an L_0 regularized object representation. The estimation of the L_0 regularized parameters can be efficiently conducted by the proposed APG algorithm.

Contributions: The contributions of this work are threefold. (1) We propose an L_0 regularized representation of the target appearance for visual tracking. Compared to the state-of-the-art algorithms, the proposed method achieves more reliable tracking results. (2) We theoretically show that the use of L_0 regularizer to represent an object has advantages over L_1 or L_2 regularizer when the dictionary is orthogonal. (3) Although the L_0 norm related minimization is an NP-hard problem, we show that the proposed model can be efficiently estimated by the proposed APG method. This makes our tracking method computationally attractive in general and comparable in speed with the methods in [24, 25] and the accelerated L_1 tracker [4].

2 Related Work

In past decades, there have been extensive literatures on object tracking. Comprehensive reviews can be found in [27, 29]. In this paper, we only briefly review the most relevant algorithms. Visual tracking algorithms can be roughly categorized into two kinds: discriminative and generative tracking. Discriminative tracking methods (*e.g.*, [2, 3, 8, 9, 10, 14, 18, 30]) use a binary classifier in solving the tracking problem. The classifier distinguishes the target from background and the region with highest classification score is considered as the target. Generative methods (*e.g.*, [1, 5, 6, 11, 13, 21, 22, 28]) employ a generative appearance model to represent the target's appearance. The tracking is achieved by searching the location which is most similar to the learned appearance model.

Recently, sparse representation has been successfully applied to visual tracking. Mei and Ling [19] assume that a target candidate can be represented as a sparse linear combination of object templates and trivial templates. Liu *et al*. [17] integrate group sparsity and high dimensional features to improve the robustness of tracking algorithm. Li *et al*. [15] use dimensionality reduction and a customized orthogonal matching pursuit algorithm to accelerate [19]. Mei *et al*. [20] propose a robust L_1 tracker with minimum error bound and occlusion detection. In [4], a faster numerical solver of [19] is proposed, and its extended version for handling multi-task is proposed in [32]. Considering the superiority of [22] and [4], the works in [24, 25] combine PCA basis [22] and sparse representation [19] schemes for object tracking. Zhang *et al*. [31] propose low-rank sparse learning method for robust tracking.

Our method is inspired by the sparse representation method [19] and subspace based tracking [22, 24]. We use an orthogonal dictionary to replace the object templates used in [19]. To improve accuracy of object representation and computational efficiency, we propose an L_0 regularized method together with an efficient solver for the object tracking.

3 Visual Tracking via L_0 Representation

In this section we propose a new model for target appearance using L_0 regularization on the coefficients and a fast numerical algorithm for solving the proposed model using APG approach [16, 23]. The visual tracker using the proposed appearance model and particle filter can run in realtime. We also show that the L_0 regularization is very effective in handling outlier pixels such as in occluded regions.

3.1 L_0 Regularized Object Representation

We assume that the target region $\mathbf{y} \in \mathbb{R}^{d\times 1}$ can be represented by an image subspace with corruption,

$$\mathbf{y} = D\boldsymbol{\alpha} + \mathbf{e}, \tag{1}$$

where the columns of $D \in \mathbb{R}^{d\times n}$ are orthogonal basis vectors of the subspace, $\boldsymbol{\alpha}$ is the sparse coefficient vector, and $\mathbf{e}$ represents additive errors modeled by a Laplacian noise. Our goal is to remove redundant features while preserving the useful parts in the subspace. Thus, we propose an L_0 regularized prior to select useful features, which is defined as

$$\min_{\boldsymbol{\alpha},\mathbf{e}} \frac{1}{2}\|\mathbf{y} - D\boldsymbol{\alpha} - \mathbf{e}\|_2^2 + \lambda\|\mathbf{e}\|_1 + \gamma\|\boldsymbol{\alpha}\|_0, \tag{2}$$

where $D^\top D = I$, $\|\cdot\|_0$ denotes the L_0 norm which counts the number of non-zero elements, $\|\cdot\|_2$ and $\|\cdot\|_1$ denote L_2 and L_1 norms, respectively, γ and λ are regularization parameters, and I is an identity matrix. The term $\|\mathbf{e}\|_1$ is used to reject outliers (*e.g*., occlusions), while $\|\alpha\|_0$ is used to select the useful features. We note that if we set $\gamma = 0$, (2) is reduced to [24]. The difference from [24] will be detailed in Sec. 3.4.

3.2 Fast Numerical Algorithm for Solving (2)

Solving (2) is an NP-hard problem because it involves a discrete counting metric. We adopt a special optimization strategy based on the APG approach [16], which ensures each step can be easily solved. The numerical algorithm for solving (2) is summarized in Algorithm 1.

Algorithm 1 Fast numerical algorithm for solving (2)

(i) Set initial guesses $\boldsymbol{\alpha}_0 = \boldsymbol{\alpha}_{-1} = \mathbf{0}$, $\mathbf{e}_0 = \mathbf{e}_{-1} = \mathbf{0}$, and $t_0 = t_{-1} = 1$.

For $k = 0, 1, ...,$ iterate until convergence

$$\begin{cases} \mathbf{z}^{\alpha}_{k+1} := \boldsymbol{\alpha}_k + \frac{t_{k-1}-1}{t_k}(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_{k-1}), \\ \mathbf{z}^{e}_{k+1} := \mathbf{e}_k + \frac{t_{k-1}-1}{t_k}(\mathbf{e}_k - \mathbf{e}_{k-1}), \\ \boldsymbol{\alpha}_{k+1} := \arg\min_{\boldsymbol{\alpha}} \gamma\|\boldsymbol{\alpha}\|_0 + \frac{L}{2}\left\|\boldsymbol{\alpha} - \mathbf{z}^{\alpha}_{k+1} + \frac{\nabla_{\boldsymbol{\alpha}} F(\mathbf{z}^{\alpha}_{k+1}, \mathbf{z}^{e}_{k+1})}{L}\right\|_2^2, \\ \mathbf{e}_{k+1} := \arg\min_{\mathbf{e}} \lambda\|\mathbf{e}\|_1 + \frac{L}{2}\left\|\mathbf{e} - \mathbf{z}^{e}_{k+1} + \frac{\nabla_{\mathbf{e}} F(\mathbf{z}^{\alpha}_{k+1}, \mathbf{z}^{e}_{k+1})}{L}\right\|_2^2, \\ t_{k+1} := \frac{1+\sqrt{1+4t_k^2}}{2}, \end{cases} \tag{3}$$

where $\nabla_{\boldsymbol{\alpha}} F(\boldsymbol{\alpha}, \mathbf{e}) = D^{\top}(D\boldsymbol{\alpha} + \mathbf{e} - \mathbf{y})$, $\nabla_{\mathbf{e}} F(\boldsymbol{\alpha}, \mathbf{e}) = \mathbf{e} - (\mathbf{y} - D\boldsymbol{\alpha})$, and L is a Lipschitz constant.

In Algorithm 1, we need to solve

$$\boldsymbol{\alpha}^*_{k+1} = \arg\min_{\boldsymbol{\alpha}} \gamma\|\boldsymbol{\alpha}\|_0 + \frac{L}{2}\left\|\boldsymbol{\alpha} - \mathbf{z}^{\alpha}_{k+1} + \frac{\nabla_{\boldsymbol{\alpha}} F(\mathbf{z}^{\alpha}_{k+1}, \mathbf{z}^{e}_{k+1})}{L}\right\|_2^2 \tag{4}$$

and

$$\mathbf{e}^*_{k+1} = \arg\min_{e} \lambda\|\mathbf{e}\|_1 + \frac{L}{2}\left\|\mathbf{e} - \mathbf{z}^{e}_{k+1} + \frac{\nabla_{\mathbf{e}} F(\mathbf{z}^{\alpha}_{k+1}, \mathbf{z}^{e}_{k+1})}{L}\right\|_2^2. \tag{5}$$

According to proof of Theorem 3.1 (detailed in Appendix), it is easy to show that the solutions of (4) and (5) can be obtained by

$$\boldsymbol{\alpha}^*_{k+1} = H_{2\gamma/L}\left(\mathbf{z}^{\alpha}_{k+1} - \frac{\nabla_{\boldsymbol{\alpha}} F(\mathbf{z}^{\alpha}_{k+1}, \mathbf{z}^{e}_{k+1})}{L}\right) \tag{6}$$

and

$$\mathbf{e}^*_{k+1} = \mathcal{S}_{\lambda/L}\left(\mathbf{z}^{e}_{k+1} - \frac{\nabla_{\mathbf{e}} F(\mathbf{z}^{\alpha}_{k+1}, \mathbf{z}^{e}_{k+1})}{L}\right), \tag{7}$$

where $\mathcal{S}_{\theta}(x) = \text{sign}(x)\max(|x| - \theta, 0)$, and $H_{\theta}(x)$ is a hard thresholding operator, which is defined as $H_{\theta}(x) = x$; if $x^2 > \theta$ and 0 otherwise.

Due to the orthogonality of D, Algorithm 1 converges fast, and its computation cost does not increase compared to the solver of L_1 regularized model.

3.3 Visual Tracking based on the Particle Filter

Most visual tracking methods are based on the particle filter framework. In this paper, we also employ a particle filter to track the target object. The particle filter provides an estimate of posterior distribution of random variables related to Markov chain. Given a set of observed images $Y_l = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_l\}$ at the l-th frame, the target state variable $\mathbf{x}_l$ which consists of the six parameters of the affine transformation can be estimated by the maximal approximate posterior (MAP) probability

$$\mathbf{x}^*_l = \arg\max_{\mathbf{x}_l} p(\mathbf{x}_l | \mathbf{y}_{1:l}). \tag{8}$$

Based on the Bayes theorem, the posterior distribution can be obtained by

$$p(\mathbf{x}_l|\mathbf{y}_{1:l}) = \frac{p(\mathbf{y}_l|\mathbf{x}_l)p(\mathbf{x}_l|\mathbf{y}_{1:l-1})}{p(\mathbf{y}_l|\mathbf{y}_{1:l-1})}, \tag{9}$$

where $p(\mathbf{y}_l|\mathbf{x}_l)$ is the observation likelihood which reflects the similarity between an observed image patch and the object class. $p(\mathbf{x}_l|\mathbf{y}_{1:l-1})$ is defined as

$$p(\mathbf{x}_l|\mathbf{y}_{1:l-1}) = \int p(\mathbf{x}_l|\mathbf{x}_{l-1})p(\mathbf{x}_{l-1}|\mathbf{y}_{1:l-1})d\mathbf{x}_{l-1}, \tag{10}$$

where $p(\mathbf{x}_l|\mathbf{x}_{l-1})$ is the state transition distribution. In this paper, we set $p(\mathbf{x}_l|\mathbf{x}_{l-1}) = N(\mathbf{x}_l;\mathbf{x}_{l-1},\Sigma)$, where Σ is a diagonal covariance matrix whose elements are the variances of the affine parameters. The observation likelihood $p(\mathbf{y}_l|\mathbf{x}_l)$ is set to be as

$$p(\mathbf{y}_l|\mathbf{x}_l) = \exp(-\tau E(\boldsymbol{\alpha}^*, \mathbf{e}^*)), \tag{11}$$

where $E(\boldsymbol{\alpha}, \mathbf{e}) = \frac{1}{2}\|\mathbf{y} - D\boldsymbol{\alpha} - \mathbf{e}\|_2^2 + \lambda\|\mathbf{e}\|_1$, $\boldsymbol{\alpha}^*$ and $\mathbf{e}^*$ are the optimal solution of (2), and τ is a constant.

The model update process is very important in visual tracking. Since the error term $\mathbf{e}$ can be used to identify some outliers (*e.g.*, Laplacian noise, illumination), we adopt the strategy proposed by [24] to update the appearance model using the incremental PCA with mean update [22] as follows,

$$y_i = \begin{cases} y_i, & e_i = 0, \\ \mu_i, & \text{otherwise}, \end{cases} \tag{12}$$

where y_i, e_i, and μ_i are the i-th elements of $\mathbf{y}$, $\mathbf{e}$, and μ, respectively, μ is the mean vector computed by [22].

3.4 Analysis on the Effectiveness of L_0 Representation

It is known that the L_0 norm is the optimal metric that is able to describe the intrinsic essence of sparse coding [7]. The benefit of the L_0 norm regularized prior is that it is able to reduce the redundant features while keeping the most important part, thereby facilitating the tracking result. In this section, we provide detailed analysis why we use the L_0 norm in the object representation.

When there are no errors (*e.g.*, occlusion) in the observation $\mathbf{y}$, i.e., $\mathbf{e} \approx 0$, then (2) reduces to

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\mathbf{y} - D\boldsymbol{\alpha}\|_2^2 + \gamma\|\boldsymbol{\alpha}\|_0, \quad \text{where } D^\top D = I. \tag{13}$$

In general we can think of L_p regularized error metric,

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\mathbf{y} - D\boldsymbol{\alpha}\|_2^2 + \gamma\|\boldsymbol{\alpha}\|_p^p, \quad \text{where } D^\top D = I, \tag{14}$$

and the solutions for different p are given in the following theorem.

Theorem 3.1 *Assume that $D \in \mathbb{R}^{d\times d}$ and $D^\top D = I$. The solution of (14) when p is 0 is given by*

$$\boldsymbol{\alpha} = H_{2\gamma}(D^\top \mathbf{y}), \tag{15}$$

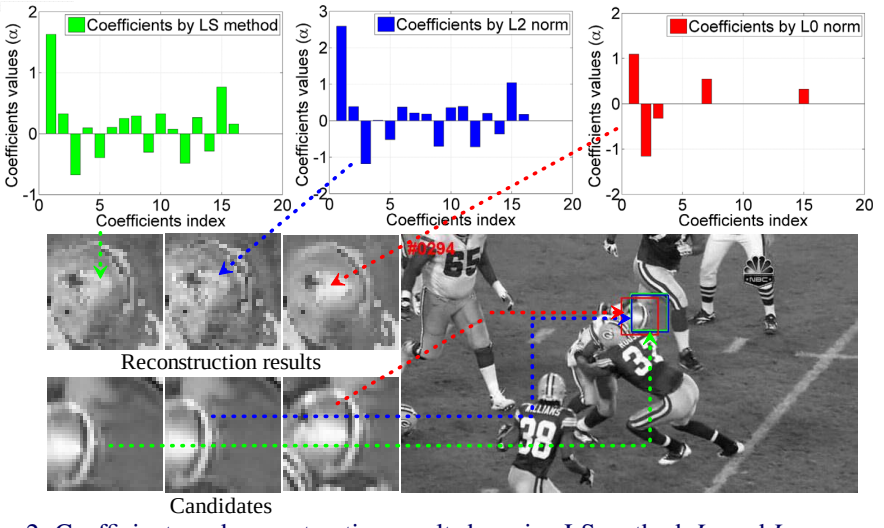


Figure 2: Coefficients and reconstruction results by using LS method, L_2 and L_0 norm under the same dictionary D, respectively. The coefficients by using L_0 norm are more sparse than those by L_2 norm and LS method, and the reconstruction result and the best candidate are also better. The rectangles in the last image represent MAP states for particle filers.

when p is 1, the solution is

$$\boldsymbol{\alpha} = \mathcal{S}_{\gamma}(D^{\top}\mathbf{y}), \tag{16}$$

and when p is 2, the solution becomes

$$\boldsymbol{\alpha} = \frac{D^{\top}y}{1+2\gamma}. \tag{17}$$

The proof can be found in Appendix. Based on Theorem 3.1, we have the following corollary.

Corollary 3.1 *We assume D contains all possible basis vectors. Let* $\mathbf{u}^*$ *denote the non-zero elements of* $D^{\top}y$. *If we set* $\gamma = \frac{1}{2}\min_i\{|\mathbf{u}_i^*|^2\}$, *the solution of* (13) *can exactly recover the data* $\mathbf{y}$.

Corollary 3.1 illustrates that the reconstruction error $\|\mathbf{y} - D\boldsymbol{\alpha}\|_2^2$ by L_0 norm is zero, whereas the reconstruction error $\|\mathbf{y} - D\boldsymbol{\alpha}\|_2^2$ by L_1 or L_2 norm may not be zero according to the solution of (16) or (17).

Theorem 3.1 and Corollary 3.1 demonstrate the properties of L_2, L_1, and L_0 regularized methods in theory. We also use the proposed Algorithm 1 to verify their properties in practice. Figure 2 shows the tracking results by using LS method [24] (i.e., $\gamma = 0$ in (2)), L_0 and L_2 norm under the same dictionary D, respectively. We note that using L_0 regularized method is able to find the good candidate when there exists occlusion, then facilitating the tracking results.

4 Experiments

The proposed method is implemented with MATLAB. We empirically set $\lambda = 0.2$, $\gamma = 0.024$, $\tau = 20$, and the Lipschitz constant $L = 6$. Before solving (2), all the candidates $\mathbf{y}$

are centralized. Considering the efficiency, each target image observation patch is resized to 32×32 pixels, the dictionary D is taken 16 eigenvectors of PCA, 600 particles are adopted, and the model is incrementally updated every 5 frames. The MATLAB code, datasets, and supplementary materials are available at http://faculty.ucmerced.edu/mhyang.

To demonstrate the effectiveness of the proposed method, we use nineteen challenging image sequences which contain different challenging situations (*e.g.*, severe occlusion, illumination, fast motion, etc.) and compare our method with eight state-of-the-art methods: IVT [22], VTD [13], APG-L1 [4], MTT [32], SCM [33], ASLA [12], SP [25], and LSST [24]. For fair comparison, we use the source codes provided by the authors and run them with adjusted parameters for the best performance.

4.1 Quantitative Evaluation

We use two metrics to evaluate the proposed algorithm with other state-of-the-art methods. The first metric is the center location error measured with manually labeled ground truth data. The second one is the overlap rate, i.e., $score = \frac{area(R_T \bigcap R_G)}{area(R_T \bigcup R_G)}$, where R_T is the tracking bounding box and R_G is the ground truth bounding box. The larger average scores mean more accurate results. Table 1 shows the average center location errors in pixels where a smaller average error means a more accurate result. Table 2 shows the average overlap rates. In these two tables, the standard deviation is also employed to further describe the accuracy of each tracker. As can be seen from these two tables, the most sequences generated by our method have lower average error and higher overlap rate values.

Data	IVT	VTD	APG-L1	MTT	SCM	ASLA	SP	LSST	Ours
Animal	16.4±46.1	10.7±17.2	11.2±74.1	17.3±136.3	10.1±11.9	**7.1± 8.7**	15.7±18.2	10.1±15.1	7.0±12.4
Car4	3.2±1.4	55.3±*	160.7±*	27.7±305.5	22.2±89.1	3.7±2.0	3.0±1.7	2.9±6.1	**2.8±1.5**
Car11	2.2±1.3	27.1±*	21.9±*	1.7±0.9	1.7±0.6	1.7±0.8	1.6±0.9	1.6±0.9	**1.5±0.8**
Caviar1	18.6±468.7	2.7±4.0	20.0±208.5	54.1±*	**0.9±0.2**	1.5±5.4	4.3±2.3	1.4±1.4	1.2±1.3
Caviar2	8.6±35.4	4.7±4.7	39.4±965.3	3.02±4.1	2.3±1.4	2.0±1.0	2.2±1.7	2.3±4.0	**1.6±0.7**
Caviar3	65.0±*	55.5±739.2	17.2±205.6	66.3±*	**1.8±1.5**	5.0±58.4	66.8±*	3.1±6.0	2.3± 2.5
DavidOutdoor	5.6±20.7	61.2±*	222.9±*	112.2±*	78.9±*	103.6±*	5.8±12.3	6.4±28.7	**5.3±11.2**
Face	12.3±7.7	190.14±*	91.7±*	99.7±*	13.8±60.8	179.1±*	24.1±*	12.3±9.0	**11.3±6.9**
Occlusion1	10.3±61.8	11.1±54.7	9.9±56.8	14.1±44.7	**3.4± 6.8**	143.7±*	4.7±13.0	5.3±16.5	5.2±10.1
Occlusion2	4.4±32.9	8.8±78.5	8.5±63.0	7.6±39.0	7.8±36.0	6.9±47.9	4.0±5.7	3.1±6.0	**2.9±3.9**
Jumping	5.9±6.5	63.0±616.7	4.5±3.8	6.3±10.7	3.9±7.4	5.3±28.0	5.0±14.8	4.8±4.9	**3.9±3.5**
Singer1	8.5±8.1	4.1±11.9	4.6±4.7	21.5±108.8	3.2±3.2	6.3±6.8	4.8±5.2	3.5±4.4	**2.9±3.4**
Owl	150.9±*	143.2±*	**4.9±8.5**	16.8±107.8	16.9±447.2	134.0±*	47.4±*	6.2±10.7	5.9±17.6
Boy	47.8±*	7.3±22.5	10.6±455.4	5.9±44.9	**2.5±2.8**	2.8±7.7	136.5±*	122.9±*	7.6±259.0
Football	7.0±18.5	5.3±24.4	38.2±*	7.5±29.9	14.1±436.0	**6.1±12.3**	6.6±60.1	7.6±55.7	7.3±35.1
Lemming	16.9±193.1	86.9±*	185.8±*	90.8±*	78.5±*	152.9±*	**9.1±73.9**	81.6±*	13.6±171.5
Dog1	3.3±14.3	15.7±736.3	3.0±7.1	**3.5±6.1**	7.8±41.0	4.1±12.30	4.6±16.0	6.5±62.8	4.4±14.7
Fish	6.3±4.6	7.2±17.0	6.5±11.6	4.0±2.9	8.1±110.2	3.5±1.6	**2.9±1.4**	3.3±2.5	3.1±3.3
Mhyang	2.9±1.7	5.6±6.9	2.6±4.2	2.1±3.0	2.5±9.9	2.1±4.9	**1.4±0.8**	2.0±1.3	2.3±2.0

Table 1: Average center location error (in pixels). The best results are shown in **bold** font. The "*" denotes the value of standard deviation is larger than 1,000.

4.2 Qualitative Evaluation

We choose some examples from the test sequences to illustrate the effectiveness of the proposed method. Figure 3 shows the visualization results.

Fast Motion: Fast motion of the target object usually leads to motion blur which increases the difficulty for tracking. Figure 3 (a-c) show the sequences *Jumping*, *Face*, and *Owl* with fast motion. In Figure 3(a), the captured images are blurred seriously. Our method tracks the target faithfully throughout the three images while the IVT [22], VTD [13], MTT [32], SCM [33], ASLA [12], SP [25], LSST [24] trackers fail to track the target due to severe blur. We note that the LSST tracker [24] performs better in sequences *Face* and *Owl*. However, the linear coefficients of incrementally updated linear basis of this method is obtained by the

Data	IVT	VTD	APG-L1	MTT	SCM	ASLA	SP	LSST	Ours
Animal	0.49±0.01	0.65±0.01	0.61±0.02	0.50±0.02	0.60±0.01	0.65± 0.01	0.50±0.01	0.58±0.01	**0.66±0.01**
Car4	**0.92±0.00**	0.45±0.12	0.24±0.11	0.51±0.04	0.52±0.03	0.90 ±0.00	**0.92±0.00**	**0.92±0.00**	**0.92±0.00**
Car11	0.81±0.00	0.43±0.13	0.52±0.09	0.83±0.01	0.80±0.01	0.84±0.01	0.83±0.01	0.84±0.01	**0.84±0.00**
Caviar1	0.25±0.14	0.85±0.01	0.32±0.18	0.29±0.18	**0.90±0.00**	0.89±0.01	0.73±0.00	0.89± 0.00	**0.90±0.00**
Caviar2	0.45±0.07	0.67±0.03	0.32±0.16	0.70±0.01	0.81±0.00	**0.84±0.00**	0.71±0.00	0.80±0.00	0.77±0.00
Caviar3	0.14±0.10	0.17±0.09	0.40±0.16	0.14±0.10	**0.88±0.00**	0.75±0.08	0.17±0.07	0.85±0.01	0.86±0.01
DavidOutdoor	0.73±0.02	0.44±0.13	0.12±0.07	0.24±0.13	0.32±0.13	0.29±0.15	0.77±0.01	0.76±0.01	**0.78±0.01**
Face	0.75±0.00	0.05±0.02	0.20±0.09	0.26±0.14	0.74±0.01	0.14±0.09	0.68±0.04	0.76±0.00	**0.77±0.00**
Occlusion1	0.84±0.01	0.77±0.02	0.76±0.03	0.82±0.01	**0.93±0.00**	0.24±0.17	0.91±0.00	0.91±0.01	0.90±0.00
Occlusion2	0.61±0.01	0.72±0.03	0.58±0.01	0.74±0.02	0.75±0.02	0.72±0.02	0.84±0.00	0.86±0.01	**0.86±0.00**
Jumping	0.62±0.02	0.08±0.06	0.67±0.01	0.64±0.02	**0.72±0.01**	0.68±0.02	0.69±0.02	0.65±0.01	0.65±0.01
Singer1	0.66±0.02	0.79±0.01	0.70±0.01	0.39±0.06	**0.87±0.00**	0.77±0.01	0.82±0.01	0.80±0.00	0.80±0.01
Owl	0.22±0.13	0.07±0.02	0.82±0.01	0.60±0.04	0.66±0.06	0.24±0.12	0.48±0.17	0.81±0.01	**0.82±0.01**
Boy	0.33±0.14	0.64±0.03	0.66±0.07	0.61±0.03	0.80±0.01	**0.79±0.01**	0.30±0.15	0.31±0.17	0.73±0.07
Football	0.71±0.02	0.74±0.02	0.55±0.14	0.71±0.03	0.67±0.11	**0.75±0.02**	**0.75±0.02**	0.69±0.03	0.73±0.03
Lemming	0.38±0.08	0.35±0.10	0.13±0.08	0.28±0.12	0.39±0.12	0.22±0.10	**0.76±0.02**	0.22±0.13	0.65±0.03
Dog1	0.73±0.03	0.61±0.08	0.70±0.04	0.70±0.04	0.69±0.02	**0.75±0.02**	0.72±0.05	0.71±0.04	**0.75±0.02**
Fish	0.79±0.00	0.75±0.01	0.80±0.01	0.83±0.00	0.79±0.03	**0.86±0.00**	**0.86±0.00**	0.85±0.00	0.85±0.00
Mhyang	0.87±0.00	0.78±0.01	0.85±0.01	0.81±0.00	**0.89±0.01**	0.88±0.01	0.84±0.00	0.82±0.00	0.83±0.00

Table 2: Average overlap rate. The best results are shown in **bold** fonts.

LS method, which may contain redundant features and interfere the reconstruction results (see Figure 2). Thus, it is not able to capture the accurate candidate.

Occlusion: Occlusion is one of the crucial problem in visual tracking. Figure 3(d), (e), (f), (k), and (l) show several examples with severe occlusion. The IVT method does not consider the occlusion in object representation and this method is less effective for the sequence with large occlusion. Both LSST and the proposed method consider the occlusion in object representation, where the occluded part is handled by $\|\mathbf{e}\|_1$. Thus, these two methods perform well when a sequence contains occlusion. However, the proposed method further use L_0-regularized term to reduce the redundant features. The results are better than those of LSST.

Illumination Change: The sequences shown in Figure 3(g) and (h) contain large illumination changes. Because our method adopts (12) to update the appearance model using the incremental PCA, it is able to deal with the illumination changes.

Background Clutter: Figure 3(i) and (j) show the tracking results in the *Car11* and *Animal* sequences with complex background. Moreover, the *Car11* sequence contains illumination changes and the *Animal* sequence also contains abrupt motion. As the proposed model is able to reduce the redundant features and takes occlusion into account, the tracking results are comparable with the state-of-the-art methods presented in Figure 3(i) and (j).

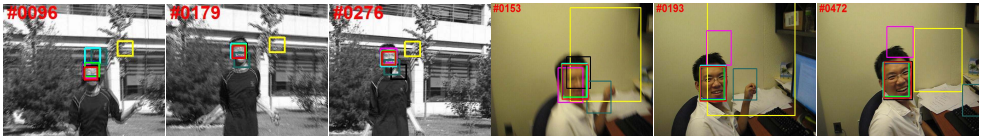
4.3 Running Time Comparison

Running time is also an important issue in tracking algorithms. Because our tracking method is based on the particle filter framework and sparse representation, we compare the proposed method with the state-of-the-art algorithms using similar approaches [4, 24]. For fair comparison, we use the same template size. Table 3 shows the comparison result. The running time[1] comparison results show that our method is much efficient and comparable to [24].

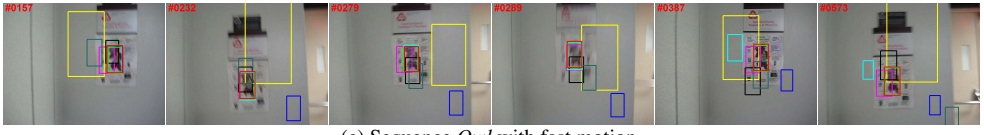
Template size	APG-L1	LSST	Ours
32×32 (pixels)	~ 0.98	~ 3.16	~ 2.30

Table 3: Comparison of running speed (frames per second)

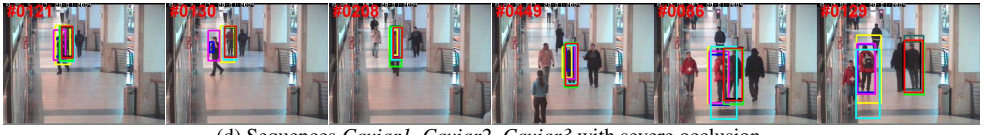
[1] All algorithms are tested on an Intel Xeon 2.53GHz machine with 12GB memory using a MATLAB implementation.



(a) Sequence *Jumping* with fast motion. (b) Sequence *Face* with fast motion.

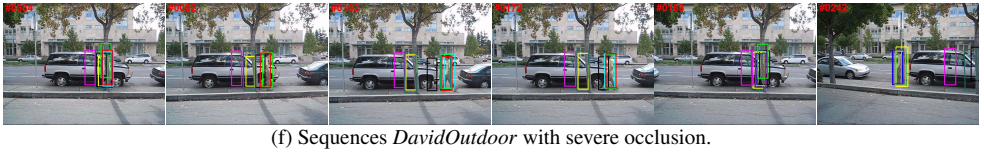


(c) Sequence *Owl* with fast motion.

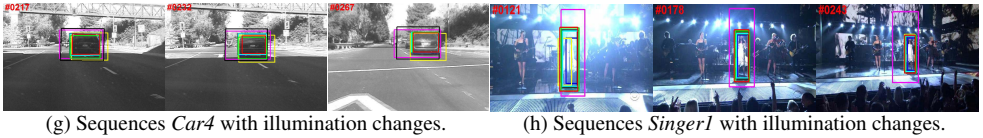


(d) Sequences *Caviar1*, *Caviar2*, *Caviar3* with severe occlusion.

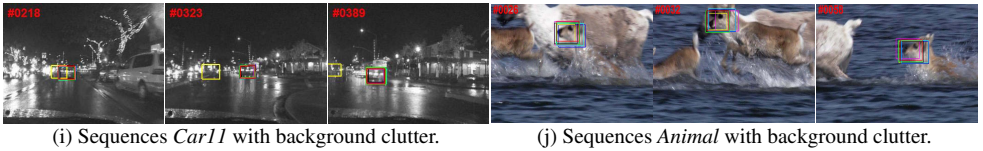
(e) Sequences *Occlusion2* with severe occlusion.



(f) Sequences *DavidOutdoor* with severe occlusion.

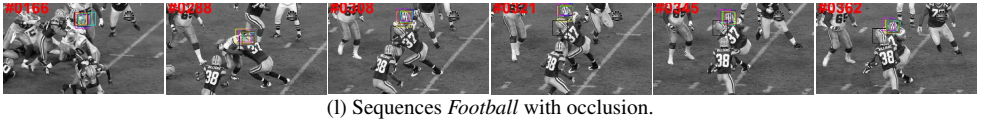


(g) Sequences *Car4* with illumination changes. (h) Sequences *Singer1* with illumination changes.



(i) Sequences *Car11* with background clutter. (j) Sequences *Animal* with background clutter.

(k) Sequences *Lemming* with occlusion.



(l) Sequences *Football* with occlusion.

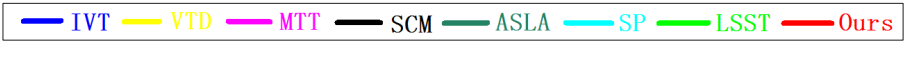


Figure 3: Sample tracking results of evaluated algorithms on several challenging image sequences. (best viewed on high-resolution display)

5 Conclusion

In this paper, we propose an L_0 sparse representation method for robust visual tracking. We provide some analysis about L_0 regularized representation when the dictionary is orthogonal and show that it has better ability to represent an object than L_1 or L_2 regularized representation. Moreover, we also develop a fast and efficient algorithm to solve the proposed model. Extensive experiments verify the superiority of our method over state-of-the-art methods, both qualitatively and quantitatively. Given the elegant properties of the L_0 norm, we plan to apply it to other vision problems, such as dictionary learning and sparsity based face recognition [26].

Appendix: Proof of Theorem 3.1

Because $D^\top D = I$, we have $\|\mathbf{y} - D\boldsymbol{\alpha}\|_2^2 = \|D^\top(\mathbf{y} - D\boldsymbol{\alpha})\|_2^2$. Thus, (13) is equivalent to the following minimization problem:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\|D^\top \mathbf{y} - \boldsymbol{\alpha}\|_2^2 + \gamma\|\boldsymbol{\alpha}\|_0. \tag{18}$$

Note that each element of $\boldsymbol{\alpha}$ is independent each other. Then, (18) can be solved with respect to each element α_i. Note that the solution of one dimensional L_0 regularized problem (18) is

$$\alpha_i = H_{2\gamma}((D^\top \mathbf{y})_i). \tag{19}$$

Similarly, the solution of one dimensional L_1 regularized problem can be obtained by shrinkage formula.

Acknowledgements We would like to thank the anonymous reviewers for their helpful comments and suggestions. The work is supported partly by the ICT R&D programs of MSIP/KEIT (No. 10047078), MSIP/IITP (No. 14-824-09-006), NSF CAREER Grant (No. 1149783), NSF IIS Grant (No. 1152576), NSFC (Nos. 61173103, 61300086, and 91230103), and National Science and Technology Major Project (2013ZX04005021).

References

[1] Amit Adam, Ehud Rivlin, Ilan Shimshoni, and David Reinitz. Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE TPAMI*, 30(3):555–560, 2008.

[2] Shai Avidan. Ensemble tracking. *IEEE TPAMI*, 29(2):261–271, 2007.

[3] Boris Babenko, Ming-Hsuan Yang, and Serge J. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, pages 983–990, 2009.

[4] Chenglong Bao, Yi Wu, Haibin Ling, and Hui Ji. Real time robust ℓ_1 tracker using accelerated proximal gradient approach. In *CVPR*, pages 1830–1837, 2012.

[5] Michael J. Black and Allan D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *IJCV*, 26(1):63–84, 1998.

[6] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *IEEE TPAMI*, 25(5):564–575, 2003.

[7] David Leigh Donoho. Compressed sensing. *IEEE TIT*, 52(4):1289–1306, 2006.

[8] Helmut Grabner and Horst Bischof. On-line boosting and vision. In *CVPR*, pages 260–267, 2006.

[9] Helmut Grabner, Michael Grabner, and Horst Bischof. Real-time tracking via on-line boosting. In *BMVC*, pages 47–56, 2006.

[10] Sam Hare, Amir Saffari, and Philip H. S. Torr. Struck: Structured output tracking with kernels. In *ICCV*, pages 263–270, 2011.

[11] Allan D. Jepson, David J. Fleet, and Thomas F. El-Maraghi. Robust online appearance models for visual tracking. *IEEE TPAMI*, 25(10):1296–1311, 2003.

[12] Xu Jia, Huchuan Lu, and Ming-Hsuan Yang. Visual tracking via adaptive structural local sparse appearance model. In *CVPR*, pages 1822–1829, 2012.

[13] Junseok Kwon and Kyoung Mu Lee. Visual tracking decomposition. In *CVPR*, pages 1269–1276, 2010.

[14] Christian Leistner, Helmut Grabner, and Horst Bischof. Semi-supervised boosting using visual similarity learning. In *CVPR*, 2008.

[15] Hanxi Li, Chunhua Shen, and Qinfeng Shi. Real-time visual tracking using compressive sensing. In *CVPR*, pages 1305–1312, 2011.

[16] Zhouchen Lin, Arvind Ganesh, John Wright, Leqin Wu, Minming Chen, and Yi Ma. Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. Technical report, UIUC, 2009.

[17] Baiyang Liu, Lin Yang, Junzhou Huang, Peter Meer, Leiguang Gong, and Casimir Kulikowski. Robust and fast collaborative tracking with two stage sparse optimization. In *ECCV*, pages 624–637. 2010.

[18] Rong Liu, Jian Cheng, and Hanqing Lu. A robust boosting tracker with minimum error bound in a co-training framework. In *ICCV*, pages 1459–1466, 2009.

[19] Xue Mei and Haibin Ling. Robust visual tracking using ℓ_1 minimization. In *ICCV*, pages 1436–1443, 2009.

[20] Xue Mei, Haibin Ling, Yi Wu, Erik Blasch, and Li Bai. Minimum error bounded efficient ℓ_1 tracker with occlusion detection. In *CVPR*, pages 1257–1264, 2011.

[21] Fatih Porikli, Oncel Tuzel, and Peter Meer. Covariance tracking using model update based on lie algebra. In *CVPR*, pages 728–735, 2006.

[22] David A. Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *IJCV*, 77(1-3):125–141, 2008.

[23] Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. submitted to SIAM J. Optimiz., 2008.

[24] Dong Wang, Huchuan Lu, and Ming-Hsuan Yang. Least soft-thresold squares tracking. In *CVPR*, pages 2371–2378, 2013.

[25] Dong Wang, Huchuan Lu, and Ming-Hsuan Yang. Online object tracking with sparse prototypes. *IEEE TIP*, 22(1):314–325, 2013.

[26] John Wright, Allen Y. Yang, Arvind Ganesh, Shankar S. Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE TPAMI*, 31(2):210–227, 2009.

[27] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *CVPR*, pages 2411–2418, 2013.

[28] Ming Yang, Ying Wu, and Gang Hua. Context-aware visual tracking. *IEEE TPAMI*, 31 (7):1195–1209, 2009.

[29] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4), 2006.

[30] Kaihua Zhang, Lei Zhang, and Ming-Hsuan Yang. Real-time compressive tracking. In *ECCV*, pages 864–877, 2012.

[31] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja. Low-rank sparse learning for robust visual tracking. In *ECCV*, pages 470–484, 2012.

[32] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja. Robust visual tracking via structured multi-task sparse learning. *IJCV*, 101(2):367–383, 2013.

[33] Wei Zhong, Huchuan Lu, and Ming-Hsuan Yang. Robust object tracking via sparsity-based collaborative model. In *CVPR*, pages 1838–1845, 2012.