Fast Visual Tracking using memory gradient pursuit algorithm

QIANG GUO [1,2] AND CHENGDONG WU [1]
1. School of Information Science and Engineering, Northeastern University
Shenyang, 110004 P.R. China;
Email: royinchina@163.com
2. Library, National Police University of China ,Shenyang 110035 P.R. China

Sparse representation scheme is very influential in visual tracking field. These L1 trackers obtain robustness by finding the target with the minimum reconstruction error via L1 norm minimization problem. However, the high computational burden of L1 minimization and absence of effective model for appearance changes may hamper its application in real world sceneries. In this research, we present a fast and robust tracking method that exploits a fast memory gradient pursuit algorithm (FMGP) with sparse representation scheme in a Bayesian framework to accelerate the L1 minimization process. For tracking, our approach adopts a non-overlapping covariance descriptor and uses a new similarity metric with scaled unscented transform. In order to reduce the problem of drift tracking, we construct a different template dictionary including benchmark template with different scales, adaptive background templates and stable templates. We test the proposed tracking method on the challenging image sequences. Both quantitative and qualitative results demonstrate the excellent performances of the proposed algorithm compared with several state of the art tracking algorithms.

Keywords: non-overlapping covariance descriptor, fast memory gradient pursuit, L1 minimization, visual tracking, scaled unscented transform

1. INTRODUCTION

Object tracking has long been extensively studied in computer vision field as it is widely applied in object identification, motion analysis, automated surveillance, human computer interaction, to name a few. The main challenge for tracking is to develop a robust and efficient tracking system which can handle appearance changes of the object and background. Numerous tracking methods have been proposed, such as Multiple instance learning [1], Distribution fields [2], Adaptive particle sampling and Adaptive appearance [3]. A though recent survey can be found in [3].

A visual tracking method usually consists of an appearance model which is first used to represent the target, a dynamic model, and a search strategy which is utilized to find the likely states in the current frame. In general, from the perspective of representation scheme, tracking algorithms can be classified as either generative algorithms or discriminative models. The former models typically focus on modeling appearance and use it to search for image regions with minimum reconstruction error as tracking results, such as ASLA [4], IVT [5], sparse representation referred to as L1 tracker [6]. The latter algorithms pose object tracking as a detection problem in which a classifier is learned to separate the target object from its surrounding background within a local region, such as CT [7], MIL.

Communicated by QIANG GUO.
* This work was supported by The National Natural Science Foundation of CHINA (No. 61273078)

Recently, much research has been focused on sparse representation methods. The pioneer work was reported in [8], where the algorithm is based on the Particle Filter framework. A sparse representation based appearance model in a visual tracking scenario is proposed, in which the target appearance is expressed as a sparse linear combination with a basis library consisting of target templates, positive and negative trivial templates via l1-minimization [9].The advantage of this method lies in the promised robustness for image corruptions. However, its extensive computational cost of the l1-minimization hampers the practical implementations. Another drawback is that the limitation of expressiveness of the object and model because it can be only represented by the subspace spanned by the training templates ignoring the discriminative information such as statistical and spatial correlation between the pixels [10]. Therefore, significant view or pose changes can not be handled efficiently.

Several tracking algorithms have been developed within the Bayesian framework to handle the time consuming issue and improve the performance. Li et al. [11] adopt dimensionality reduction and customized Orthogonal Matching Pursuit (OMP) to accelerate the speed. However, we note that useful information such as color feature is not accommodated in [11]. A minimum error bounding strategy [12] is introduced to reduce the number of L1 norm minimization. In [13], Bao et al. propose a new L1 tracker based on the accelerated proximal gradient approach (L1APG). The templates used in these trackers are updated in a simple and unreliable way, which leads to drift tracking during appearance changes. Although these improved methods run faster than the original L1 tracker, there is still much room for improvement of robustness and efficiency.

In this paper, inspired by application of OMP in compressive sensing tracking [11], we propose an efficient and robust tracking method. The contributions of our work are summarized as follows.

- We propose a fast memory gradient pursuit (FMGP) algorithm for accelerating L1 tracker, which makes use of the sparse signal recovery power to reduce the computational complexity without losing the tracking performance. Thereby, we formulate a more efficient tracker than state of the art tracker [12], [13].
- The appearance model is efficiently represented by non-overlapping covariance matrix which can accommodate more informative feature and more discriminative than the original image patch. Moreover, a new templates update scheme is involved in our tracker, which improves the robustness of the tracker. We also scale the target model to different scales and incorporate them in our template model. Thereby, to some extent, scale invariance can be achieved.
- We develop a new metric approach of similarity of covariance feature match process by Scaled unscented transformation (SUT), which approximates mean and covariance representation on Euclidean vector space. The metric is simpler to calculate and better than other metrics for covariance matrix.

Our method is different from [20], [34]. We take non-overlapping covariance matrix dividing target area into segments to represent target appearance model instead of [20] which obtain covariance matrix descriptor of the whole region without considering spatial information. The democratic integration method in [34] is for adaptive feature selection in appearance model phase while we use it as our update template strategy to model the variation.

2. CONTEXT AND RELATED WORK

Our tracking algorithm is built upon a sparse representation tracking framework based on estimating the distribution of each target state by a particle filter. To present our model in the latter section clearly, we briefly review the particle filter approach in section2.1, and then L1 tracker or sparse representation in section 2.2. Some new tracking methods based on particle filter can be found in [14], [15], and [16]. In [17], multiple instance learning and local sparse representation are combined for tracking.

2.1 Particle filtering

The particle filter uses the available measurement information to estimate the hidden state variables. It approximates the posterior probability density function $p(s_{0:k}|z_{1:k})$ by a group of random particles or samples $\{s_k^i\}_{i=1}^{N_s}$ with its corresponding weight $\{w_k^i\}_{i=1}^{N}$, where s_k represent system state at the frame time instant k, z_k is observation state which describes the location and the feature of the target in visual tracking task. The sample with the maximum weight is considered as the target.

The weights of the samples are updated as

$$w_k^i = w_{k-1}^i \frac{p(z_k|s_k^i)p(s_k^i|s_{k-1}^i)}{q(s_k^i|s_{0:k-1}^i, z_{0:k})} \tag{1}$$

where $p(z_k|s_k^i)$ denotes the observation likelihood, $q(s_k^i|s_{0:k-1}^i, z_{0:k})$ denotes the importance distribution, $p(s_k^i|s_{k-1}^i)$ is the transition probability distribution.

In reality, some simple forms of the Eq. (1) have been proposed [18]. In the case of the bootstrap filter, the weights become the observation likelihood.

2.2 Sparse representation

For each particle, we can obtain a corresponding region using parameters such as coordination and size in the particle. Then, the particle can be represented in sparse representation framework which has been widely applied in numerous vision applications. In [18] [19], the target candidate is sparsely represented as a linear combination of the atoms of a dictionary composed of dynamic target templates and trivial templates.

A new image target candidate patch $\boldsymbol{y} \in R^d$ can be approximately represented as a linear combination of templates $\mathbf{T}$. If given a region image template set $\mathbf{T} = [t_1, t_2, \ldots, t_{N_t}] \in R^{D \times N_t} (D \gg N_t)$ containing N_t target templates and a trivial template set $\mathbf{E} = [\mathbf{I}, -\mathbf{I}] \in R^{D \times 2D}$, a common sparse representation model is then [6],

$$\mathbf{y} = \mathbf{T}\mathbf{a} + \mathbf{e} = [\mathbf{T}, \mathbf{I}, -\mathbf{I}] \begin{bmatrix} \mathbf{a} \\ \mathbf{e}^+ \\ \mathbf{e}^- \end{bmatrix} \tag{2}$$

where $\mathbf{a} = [a_1, a_2, \ldots, a_{N_t}]^T \in R^{N_t}$ is a template coefficient vector, $\mathbf{e}$ is the error vector. $\mathbf{e}^+$ and $\mathbf{e}^-$ represent a positive trivial coefficient vector and a negative trivial coefficient vector, $\mathbf{I}$ is the identity matrix. Let $\mathbf{A} \doteq [\mathbf{T}, \mathbf{E}] \in R^{D \times (2D + N_t)}$ is the over

complete dictionary and the sparse coefficient vector $\mathbf{x} = [\mathbf{a}, \mathbf{e}^+, \mathbf{e}^-]$ can be achieved via solving the following l1-norm minimization problem with non-negativity constraints.

$$\min\|\mathbf{x}\|_1 \quad s.t. \ \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \varepsilon, \ \ \mathbf{x} \succcurlyeq 0 \tag{3}$$

$$r = \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \tag{4}$$

The residual is obtained by (4). The formula (3) is also well known as the basis pursuit denoising problem with a scalar λ as follows:

$$\min \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda\|\mathbf{a}\|_1 \quad s.t. \ x \succcurlyeq 0 \tag{5}$$

where $\mathbf{a}$ is the control coefficient.

The tracking result is the particle obtaining the smallest residual after projecting on the target template subspace. Therefore, the estimated tracking results as follows.

$$\hat{\mathbf{x}} = \text{argmin} \ \|\mathbf{y} - \mathbf{T}\mathbf{a}\|_2 \tag{6}$$

Finally, the observation likelihood of state can be denoted as:

$$p(\mathbf{y_t}|\mathbf{s_t}) = \vartheta \exp(-\rho\|\mathbf{y_t} - \mathbf{T}\mathbf{a}\|_2^2) \tag{7}$$

Where ϑ is a normalization constant, ρ is a constant controlling the shape of the Gaussian kernel.

3. COVARIANCE DESCRIPTOR AND CORRESPONDING NOVEL SIMILARITY METRIC

Region covariance were introduced by [20] as a novel region descriptor. Advantages of region covariance can be reviewed in [16][17]. The reason why we do not consider other descriptors such as SIFT and SURF which are more suitable for detecting the object in every frame for tracking system is that they are complicated to integrate into L1 tracker framework. Let O be the observed image with the $\mathrm{w} \times \mathrm{h} \times \mathrm{d}$ dimensional feature image F, covariance descriptor extracted from O, $F(\mathrm{x}, \mathrm{y}) = \emptyset(\mathrm{O}, \mathrm{x}, \mathrm{y})$, where $\emptyset$ can be any mapping from O to F such as color, gradients, filter responses. The region R is represented by the $d \times d$ covariance matrix of the feature points

$$C = \frac{1}{N-1}\sum_{i=1}^{N}(f_i - \mu)(f_i - \mu)^T \tag{8}$$

where N is the number of pixels in the region patch, and μ is the mean vector of the feature set $\{f_\mathrm{i}\}_{\mathrm{i}=1}^{\mathrm{N}}$. The element (i, j) of a second order sample matrix $C \in R^{d \times d}$ represents the correlation between feature i and feature j at the specified position x and y in the image. When the extracted d-dimensional feature includes the pixel's coordinate, the covariance descriptor encodes the spatial information of features.

3.1 Related work on Covariance Metric

We choose the covariance descriptor as the target representation. However, some weaknesses need to be minimized before applying it to our tracking framework. We note that the covariance descriptor does not lie in the Euclidean space. As the covariance matrixes are symmetric positive-definite matrixes lied in the Lie algebra or the Riemannian Manifolds, computations involving this metric are expensive, especially for large matrices and even more so, in gradient based algorithms. The current common distance or similarity computations with covariance such as [20] and [21] tend to be very slow.

A closely related one is Log-Euclidean metrics proposed in [22], which is also a Riemannian metric. By turning covariance matrices into symmetric matrices through logarithm map, the matrices can then be handled in Euclidean space. Thus the dissimilarity can be measured in the domain of logarithms by Euclidean metrics.

$$\rho(C_1, C_2) = \|log(C_1) - log(C_2)\| \tag{9}$$

The similarity measurements mentioned above cannot be applied in real-time applications because of expensive computations and manifold mappings. In contrast to these approaches, where the similarity is approximated and computationally expensive processing steps are taken on Riemannian manifolds, we take the idea in [23] of relying on approximating the first and second order moments on Euclidean vector space. The idea is based on choosing a representative set of samples of two given distributions and to compute distance in Euclidean for similarity. We introduce the idea for approximating similarity of covariance descriptor through another way in section 3.2.

3.2 The Scaled Unscented Transform (SUT)

The unscented transformation is a method for calculating the statistics of a random variable which undergoes a nonlinear transformation and builds on the principle that it is easier to approximate a probability distribution than an arbitrary nonlinear function [23]. [23] take the unscented transform (UT) proposed in [24], which approximates a distribution by specified sampling sigma points instead of approximating an arbitrary non-linear function by mapping to manifolds. The mean and covariance of probability distributions can be matched efficiently by UT. However, its aggregation of selected sigma points will deteriorate when the dimension of sample vectors increases, and the radius of the sphere that bounds all the sigma points will increase, which introduce errors. To overcome the weakness of UT, we adopt the scaled unscented transform (SUT) [25], which control the expansion of the sigma points and ensure the positive semi-definiteness of estimated covariance matrix.

In the following, the well studied mean and covariance descriptors, and the idea of approximation will be discussed. If given a set of sig points S constructed with a sigma point selection algorithm such as the non-linear transform [26], the set of $2n_s + 1$ vectors s_j is generated as follows.

$$s_0 = \mu \tag{10}$$

$$s_j = \mu + \left(\sqrt{(n_s + \lambda)P_s}\right)_j \quad j = 1,2, \dots, n_s \tag{11}$$

$$s_{j+d} = \mu - \left(\sqrt{(n_s + \lambda)P_s}\right)_j \quad j = n_s + 1, \ldots, 2n_s \tag{12}$$

with $\lambda = \alpha^2(\mathrm{n_s} + \omega) - \mathrm{n_s}$ representing the scale parameter, where j denotes the index , $\left(\sqrt{(n_s + \lambda)P_s}\right)_j$ defines the j-th column or row of the square root matrix, α determines the spread of the sigma points around mean μ, P_s is the covariance and ω is a secondary scaling factor.

The estimated mean μ' and covariance P_s' of S are computed as follows.

$$P_s' = \sum_{j=0}^{2n_s} w_j^c (s_j - \mu)(s_j - \mu)^T \tag{13}$$

$$\mu' = \sum_{j=0}^{2n_s} w_j^m s_j \tag{14}$$

Each sigma point has its corresponding weight w_i, as follows:

$$w_0^m = \frac{\lambda}{n_s + \lambda} \tag{15}$$

$$w_0^c = \frac{\lambda}{n_s + \lambda} + 1 - \alpha^2 + \beta \tag{16}$$

$$w_i^m = w_i^c = \frac{\lambda}{2(n_s + \lambda)} \qquad j = 1,2, \ldots n_s \tag{17}$$

where β is a parameter which minimizes the effects from high order terms, we set $\beta = 2$ for Gaussian distribution, the superscript m represents the weight for the mean calculation, the superscript c indicates the weight for the covariance calculation.

The adoption of SUT allows us to increase the robustness of the final technique without any extra computational expenses [27]. It is obvious that each of these generated vectors s_i describes a d-dimensional Euclidean space. Consequently, the approximated covariance representation in Euclidean vector space derived by SUT enables a plausible and simple integration into the tracking framework.

3.3 Approximated Non-overlapping Covariance descriptor

As the covariance descriptor is more robust to problems such as pixel-pixel misalignment, changes in pose and illumination, we then choose it as our target descriptor. However, it is not suitable for L1 tracker. To increase the robustness of tracking and handle the tracking procedure during occlusions, we divide the object region patch into $p \times q$ small non-overlapping blocks, and then each block is represented by a reference region covariance matrix feature respectively, denoted as C_{ij}, $\mathrm{i} = 1,2 \ldots, \mathrm{p}$, $\mathrm{j} = 1,2 \ldots, \mathrm{q}$. To obtain the feature vector S_{ij}, the adoption of the SUT, which can compute distances between covariance descriptors efficiently and discriminately on Euclidian space, is different from [28] applying the Log-Euclidean mapping to C_{ij}. Finally, all feature vectors S_{ij} are concatenated to constitute the final representation. After concatenating these vectors into one and deleting other blocks' position information, the object can be described by our approximated non-overlapping covariance feature vector

c. Consequently, the feature can be applied to sparse representation framework.

The use of the above approximated non-overlapping covariance matrix bring multiple advantages. First, it fully captures both spatial and statistical information of the target. Second, it provides an elegant solution to merge different features and modalities with an economic computational time. Third, it provides a covariance representation on Euclidean space and is capable of comparing regions similarity efficiently using classic Euclidean distance metric. Though the covariance feature contains position information of an image patch, the position information of sparse representation is indistinct after transform. Thereby, we discard trivial templates for occlusion reasoning in [9].

4. PROPOSED TRACKING ALGORITHM

In section 3, covariance descriptor and its metric is adopted to represent the target, we use L1 tracker framework based on particle filter in section 2 to analyze the candidate location at next frame. In this section, we give the details of our tracking method. The efficiency of L1 tracking framework is improved by merging our MGP algorithm.

4.1 Fast Memory Gradient Pursuit Algorithm

The Eq.(3) can be solved by greedy approximation approach which is generally categorized into two types. One type is Matching Pursuit such as Orthogonal Matching Pursuit (OMP) [29], Regularized Orthogonal Matching Pursuit (ROMP) [30], which are iterative greedy algorithms that select atoms most correlated with the current residuals at each step. The solutions are obtained by least square method. However, the Matching Pursuit methods are inefficient and cause heavy computation burden, especially when the number of particle is large in tracking. Another type of greedy approximation is direction pursuit approach. The latter has faster convergence rate than matching pursuit type and can reduce computation time with gradient pursuit method.

In [6][9], Xue proposes the L1 tracker using preconditioned conjugate gradients algorithm as reconstruction algorithm. Li et al. [11] adopt customized Orthogonal Matching Pursuit as reconstruction algorithm. We propose a fast memory gradient method to reduce the reconstruction time and solve the l1-minimization problem efficiently. More details of experimental results of MGP are available in [31].

Our FMGP algorithm firstly adopts regularization orthogonal matching strategy to select atom sets fast and efficiently. Then under the framework of direction pursuit, the search step size is determined by non-monotonic linear search strategy and update direction is fixed with the memory gradient algorithm. After that, sparse coefficients are achieved. The proposed algorithm takes advantages of globally fast and stable convergence of memory gradient algorithm with Armijo line search to avoid local optimal solution. Gradient methods usually take the form as follows.

$$\mathbf{x}_n = \mathbf{x}_{n-1} + \alpha_{n-1}\mathbf{d}_{n-1} \tag{18}$$

Where $\mathbf{x}_{\mathrm{n}}$ is the n-th approximation to the solution, $\mathbf{d}_{n-1}$ is a search direction, $\alpha_{\mathrm{n}-1}$ is a step size. We also denote function $g(\cdot)$ as gradient function. Iterative method is widely used for the solution. During the algorithm of conjugated gradient, direction is

updated by conjugated gradient. The memory gradient method also aims to accelerate the steepest descent method and achieve a stable convergence at the same time, which take negative gradient direction of current iteration and search direction of previous iteration as the search direction of current iteration. Directional update process of memory gradient pursuit algorithm can be described as follows.

Step 1) If $\|g(\mathbf{x}_n)\| = 0$, then stop; else go to step 2)

Step 2) Compute search direction $\mathbf{d}_n$ according to formula (19), search step size α_n is determined by non monotonic Armijo linear search strategy as [32]. The gradient value $g(\mathbf{x}_n)$ written as $\mathbf{g}_n$

$$\mathbf{d}_n = \begin{cases} -\mathbf{g}_n & n = 1 \\ -[(1-\beta_n)\mathbf{g}_n + \beta_n \mathbf{d}_{n-1}] & n \geq 2 \end{cases} \tag{19}$$

$$\mathbf{g}_n = \nabla\left[\frac{1}{2}(\mathbf{y}-\mathbf{A}\mathbf{x})^{\mathrm{T}}(\mathbf{y}-\mathbf{A}\mathbf{x})\right] = \mathbf{A}^{\mathrm{T}}(\mathbf{y}-\mathbf{A}\mathbf{x}_{n-1}) = \mathbf{A}^{\mathrm{T}}\mathbf{r}_{n-1} \tag{20}$$

$$\beta_n = \frac{\rho\|\mathbf{g}_n\|^2}{\|\mathbf{g}_n\|^2 + |\mathbf{g}_n{}^{\mathrm{T}}\mathbf{g}_{n-1}|} \quad , \quad \rho \in \left(0, \frac{1}{2}\right) \tag{21}$$

Step 3) Set $n = n + 1$, go to Step 1)

Directly solving (3) can be computationally expensive especially when dimensionality D grows. Li et al. [11] first apply customized orthogonal matching pursuit to real-time tracking to accelerate the tracking process. Here we employ our proposed fast memory gradient algorithm instead of OMP. From the Eq. (21), parameter β_{n} can be derived by [33] as follows. More derivation process details in [31].

$$\beta_n = \frac{\mathbf{d}_{n-1}^{\mathrm{T}}\mathbf{G}\mathbf{g}_n}{\mathbf{d}_{n-1}^{\mathrm{T}}\mathbf{G}\mathbf{g}_n + \left\|\mathbf{d}_{n-1}^{\mathrm{T}}\mathbf{G}\mathbf{d}_{n-1}\right\|_2^2} = \frac{< \left(\mathbf{A}^{\mathbf{\Lambda}^n}, \mathbf{d}_{n-1}^{\mathbf{\Lambda}^n}\right), \left(\mathbf{A}^{\mathbf{\Lambda}^n}, \mathbf{g}_n^{\mathbf{\Lambda}^n}\right) >}{< \left(\mathbf{A}^{\mathbf{\Lambda}^n}, \mathbf{d}_{n-1}^{\mathbf{\Lambda}^n}\right), \left(\mathbf{A}^{\mathbf{\Lambda}^n}, \mathbf{g}_n^{\mathbf{\Lambda}^n}\right) > + \left\|\left(\mathbf{A}^{\mathbf{\Lambda}^n}, \mathbf{d}_{n-1}^{\mathbf{\Lambda}^n}\right)\right\|_2^2} \tag{22}$$

where $\mathbf{\Lambda}^n$ is the set of indices at iteration n, $\mathbf{A}^{\mathbf{\Lambda}^n}$ denotes the matrix composed of the columns from $\mathbf{A}$ restricted to the set $\mathbf{\Lambda}^n$, and let $\mathbf{G} = \left(\mathbf{A}^{\mathbf{\Lambda}^n}\right)^{\mathrm{T}}\mathbf{A}^{\mathbf{\Lambda}^n}$. The new form of parameter β_n is more efficiently for computation because $\left(\mathbf{A}^{\mathbf{\Lambda}^n}, \mathbf{d}_{n-1}^{\mathbf{\Lambda}^n}\right)$ has already been calculated at previous iteration. Sparsity level L represents the number of zero elements of a signal. It should be noted that superscript n of g is the same as subscript n of g in Eq. (22).

Algorithm 1 Fast Memory Gradient Algorithm for tracking

Input A normalized observation $\mathbf{y} \in \mathrm{R}^{\mathrm{d}}$, A template set $\mathbf{A}$, sparsity level $L = \|\mathbf{y}\|_0$

1) Initialize the residual $\mathbf{r_0} = \mathbf{y}$, index set $\mathbf{\Lambda} = \phi$, $\mathbf{J} = \phi$

2) Repeat until stop criteria

① Calculate coefficient $\mathbf{u}$ and choose L largest components of $\mathbf{u}$, then put corresponding indices set in $\mathbf{J}$

$$\mathbf{u} = \{u_i | u_i = |< \mathbf{r}, \mathbf{A}_i >, i = 1,2, \ldots, N|\}$$
$$\left\|u_{\mathbf{J}_0}\right\| = \max\left\{\left\|u_{\mathbf{J}_n}\right\|, n = 1,2, \ldots, L\right\}$$

where $\mathbf{J}_n$ is an index set including n sets of coefficient. Then, regularization process to get the initial candidate set $\mathbf{J}_0$ which holds the index corresponding to maximum $\mathbf{u}$, update support set $\mathbf{A}^{\mathbf{\Lambda}^n}$, indices set $\mathbf{\Lambda}^n = \mathbf{\Lambda}^{n-1} \cup \mathbf{J}_0$

3) Calculate $\mathbf{x}_n$, $\mathbf{d}_n$, $\mathbf{r}_n$ by MGP algorithm

① calculate gradient $\mathbf{g}_n = \mathbf{A}^{\mathrm{T}}\mathbf{r}_{n-1}$

② calculate direction by Eq.(19) and Eq. (22)

③ calculate step size by Armijo non-monotonic linear search strategy

④ Obtain estimate solution by Eq. (18) $\mathbf{x}_{n+1}^{\mathbf{\Lambda}^n} = \mathbf{x}_n + \alpha_n \mathbf{d}_n$

⑤ update residual $\mathbf{r}_n = \mathbf{y} - \mathbf{A}^{\mathbf{\Lambda}^n}\mathbf{x}_n$

Output support set $\mathbf{\Lambda}^n$, recover coefficient $\mathbf{x}_n^{\mathbf{\Lambda}^n}$

4.2 Fast Tracking by MGP Algorithm

The basic idea of our tracking algorithm is that each candidate is sampled in the region of interests at new frame. Then, samples and templates are described by non-overlapping covariance features. Each of candidate targets is sparsely represented in the space spanned by templates. Incorporate these templates into the optimization process in Eq. (5). The sparse coefficient is achieved by solving optimization problem using FMGP. Then, the observation reconstruction error is calculated via Eq. (7). The SUT is used for the covariance feature match process.

Algorithm 2 Fast visual tracking using FMGP algorithm

Input • Current frame F_k • Sample Set $s_{k-1}^i = \{x_{k-1}^i\}_{i=1}^{N_s}$,
• A templates set $T = \{T_i\}_{i=1}^n$

Begin

1: Generates new particles $s_k^i, i = 1, \ldots, N_s$ to get candidate target
2: **for** $i = 1$ to N_s do
3: Drawing the new sample x_k^i from x_{k-1}^i
4: Patch initialization and Templates generation of the patch
5: Scale the target model to different scales
6: Get $\mathbf{x}$ via solving Eq. (5) with Algorithm 1;
7: Calculate residual $\mathbf{r}_i$ via Eq. (4);
8: Calculate observation likelihood via Eq. (7) and get the target $\mathbf{y}_k$
9: **end**
10: Chosen the tracking result via Eq. (6)
11: Recalculating $\mathbf{x}_k$ by solving Eq. (3)
12: Update templates T

End

Output

•Tracked target (x_k, y_k)• Updated target dynamic state • Updated target templates **T**

4.3 Scale Invariance based Template Updating

The main challenge of tracking can be attributed to the difficulty in handling the appearance changes of a candidate object. We design a proper model update scheme to

adapt the changes.

In our method, image patches can be categorized into three types as our template set. The first type called benchmark template is manually initialized from the object region, and is kept until tracking finished. The second type is adaptive background templates containing some background pixels around the initialized tracked target, and is adaptive updated. The third type is stable template initialized by first order image moment. The benchmark and stable template can prevent drifting problem. Backgrounds templates are initially obtained by perturbating around the benchmark at frame one and update during tracking process. In our update scheme, templates are weighted by their contributions with its coefficient when representing the estimated target candidate. The only ground truth is the benchmark template at the first frame, and is never updated.

The earlier results sometimes would be more significant than newly required results because the latter may involved more background noises. While in other perspective, newer frames also bring newly useful information. To capture the essence of changes of environment, we use strategy of democratic integration [34] to adaptive update the role of the recent observations of stable template by Eq. (23) and Eq. (24).

For the stable template, template updating as follows.

$$Y_k = \eta y_k + (1-\eta) Y_{k-1} \tag{23}$$

$$T_s = Y_k \tag{24}$$

where η is a forgotten factor, $\eta = 1 - e^{-1/\tau}$, T_s represents a template. When current state is more reliable, η should be larger. y_k represents observation determined by Eq. (7). A quality function in Eq.(25) measures how successful the template predicts the result or how much it agrees with it.

$$q_l(t) = e_t^{-\theta_l} \tag{25}$$

θ_l denotes distance between image candidate and benchmark template. The update scheme is made adaptive by defining dynamics for the reliabilities.

$$\tau_l \dot{\varepsilon}_l(t) = q_l(t) - \varepsilon_l(t) \tag{26}$$

where τ_l is the time constant for updating template every τ_l interval, $\varepsilon_l(t)$ is the template obtained at the estimated target position. Raising the weight of high value of quality function and decreasing the weight of low value through Eq. (26), adaptive weight update scheme can be achieved. Moreover, the weighting scheme is adopted to ensure that less modeling power is expended to fit older observations.

For the adaptive background templates, the template with the smallest coefficient is replaced with the newly tracking result when similarity of candidate patch and stable template with the largest coefficient exceed a predefined threshold H_1.

Considering the importance of scale variant for tracking, we also scale the target model to different scales. But, matching every scaled template leads to repeated calculation in the match process. For real time requirement, only two scaled templates such as -10% and 10% sizes of the original patch at previous frame are added to template set at every frame.

5. EXPERIMENTS

In this section, qualitative and quantitative experimental results are presented to evaluate the performance of the proposed tracker on the list of video sequences publicly available by [35], which contain challenging variations including background changes, pose and scale changes, occlusions and background clutter (the implementations provided by the authors used for fair comparisons). Our method is implemented in Matlab on Intel Core I7 3.4 GHz PC with 4G RAM. We test the processing speed of our tracker on the standard benchmark which consists of different sequence dataset in [35].

Our method is compared with five state-of-the-art algorithms including (ALSA) [4], the Multiple instance learning tracker (MIL), L1APG [8], Distribution fields tracker (DF) [2] and Compressive tracker (CT) [7]. For fair evaluations, we use the source code of each tracker and take the experiment evaluation methodology as [35] which contains 51 test sequences. For our experiments, we set important parameter as:
$N_s = 500,\ n_s = 2,\ \tau = 33,\ H_1 = 20,\ \alpha = 0.9,\ \tau_l = 0.5.$

The ground truth object locations are obtained by manual labels at each frame in all of the test sequences. The average processing speed of our tracker is about 20 frames per second (FPS), which will be further improved with code optimization and dimensionality reduction. It should be noted that our experiment has taken advantage of acceleration by performing minimum error bounded L1 in [12].

5.1 Comparison with other L1 trackers

Our proposed tracking algorithm can be categorized into L1 trackers. For fair comparison, we use the homologous appearance model and just compare the efficiency for l1-norm minimization. The ratio of average time of running our proposed algorithm to L1 tracker is 1: 40, the ratio of our method to L1APG is 1:2, and the ratio of our algorithm to L1MP [36] is 1:60. The frame per second (FPS) for overall algorithm of L1 trackers can be found in Table II.

5.2 Quantitative Comparison

There are two widely used evaluation criteria in many experiments to assess the performance of the tracker. One is the success rate (SR).The score of success rate is defined as $\text{score} = \frac{A \cap B}{A \cup B}$, where A is the ground truth bounding box and B is the track result rectangle, $\cap$ and $\cup$represents the intersection and union of the two regions respectively. The result of a frame is classified as the success when the score is larger than a given threshold 0.5. However, using one success rate value at a specific threshold (e.g. score=0.5) for tracker evaluation may not be representative. Therefore, we use the area under curve (AUC) of each success plot to rank the tracking algorithms as [34] in the Fig. 1. We just take the six classical trackers of totally 29 trackers in [34] and the value is obtained by testing on 51 sequences. Another criterion is the precision rate (center location error, CLE). It can be seen from Table I of the center location error that our algorithm performs better than other five state-of –the-art algorithms only except in only few frames. The best two results are shown in red and blue fonts. For presentation clarity, we just list the results of 13 sequences in terms of center location error.

Table 1.The average Center location error (in pixels) of the 13 sequences

Sequence	Ours	L1APG	DFT	CT	MIL	ASLA
Car4	**6.1**	36.9	39	38.3	37.9	**5.8**
Coke can	**8.5**	40	29.7	**18.23**	20.85	30.6
David1	**18.2**	38.3	28.6	25.8	29.1	**21.8**
FaceOcc1	**6.9**	8.3	**5**	9.32	24.2	26.7
FaceOcc2	**9.2**	13	**11.25**	17.41	21	12.6
Girl	**11.8**	14.2	32.6	23.8	27.7	**12.5**
Shaking	**7.6**	39.7	10.9	37.8	16.3	**5.8**
Skating1	**36**	66.2	63.6	78.1	71	**23.3**
Soccer	**21**	98.4	67.1	**33**	57.3	117
Sylvester	12.9	23.4	56	**9.1**	12.6	**6.3**
Tiger1	**10**	21	**7.9**	13.3	14.8	16
Trellis	**14.8**	15.5	32.2	67.3	65.8	**11.7**
Woman	**11**	134.9	**8.6**	129.7	144	151.2
Average SR	**13.4**	42.3	**30.2**	38.5	41.7	33.9

Table 2. Average tracking speeds of the six algorithms

Sequence	Ours	L1APG	DF	CT	MIL	ASLA
Average FPS	20	14.5	13	71	38	8.9

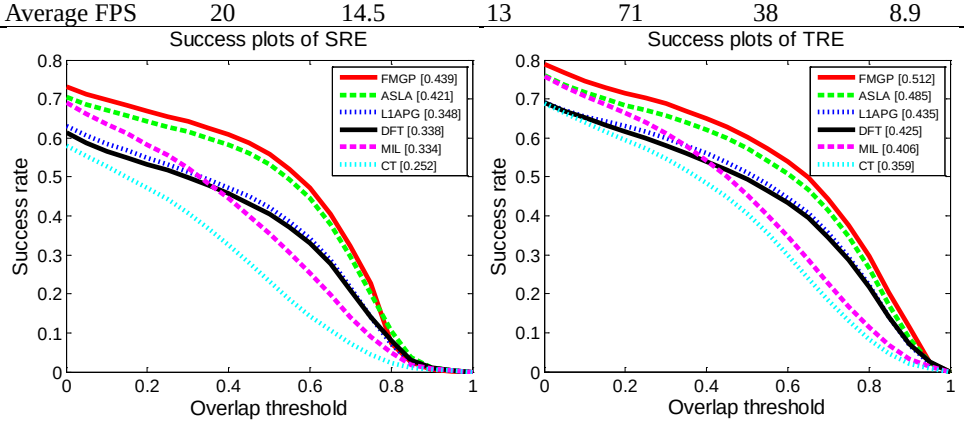


Figure 1.Plots of TRE and SRE for 51 sequences. The performance score for each tracker is shown in the legend.

The robustness evaluations of trackers are also considered in [34] presenting two new and reasonable ways, which are temporal robustness evaluation (TRE) and spatial robustness evaluation (SRE) that analyze a tracker's robustness to initialization by perturbing the initialization spatially(i.e., start by different bounding boxes). For TRE, the entire sequence is partitioned into segments and runs the tracker to the end of the sequence, which is evaluated on each segment.

5.3 Qualitative Comparison

We qualitative evaluate performance of tracking results of the 51 widely used representative video sequences in five different aspects. And we show some representative tracking results on the sequences for CT, DFT, ASLA, L1APG, MIL and FMGP tracking methods presented in Figure 2,3,4,5.

(a) Coke can

(b) Shaking

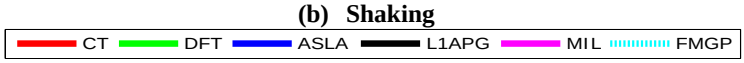
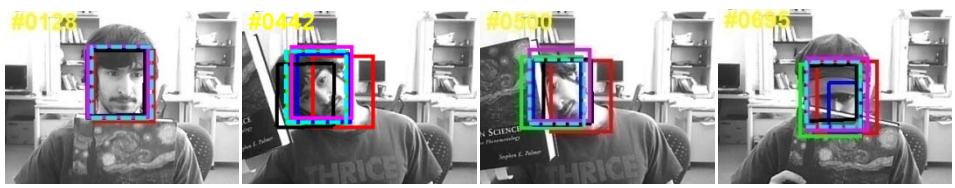
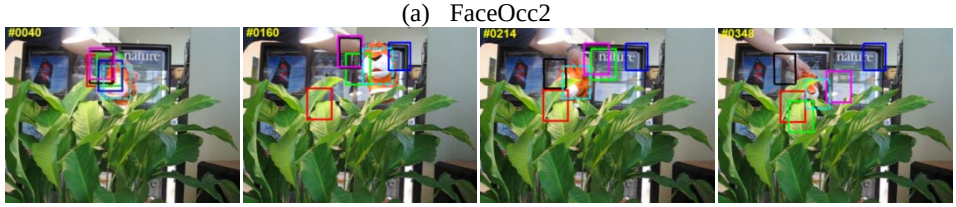


Fig2. Some tracking results of the sequences Coke can and Shaking.



(a) FaceOcc2



(b) Tiger1

(c) Woman

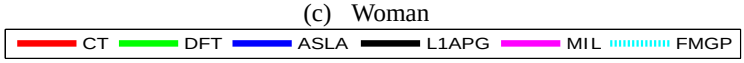


Fig3. Some tracking results of the sequences FaceOcc2, Tiger1 and Woman.

Illumination, pose and scale variation. We elaborately select the evaluated sequences with different kind of large illumination variations. The evaluated sequences in Coke and Shaking show a coke can and a singer in Fig. 2 that undergoes drastic illumination variations due to cast shadows and ambient lights. In coke sequence, DFT, CT and MIL perform well in the first 100 frames, but gradually drift (see #169, #194, #231). Only the proposed tracker is able to track the target more accurately during the whole process. In the Soccer sequence (see frames in Fig4), the player is occluded in a scene with large change and illumination, our tracker performs well during most of the

time, still better than other trackers. This can be attributed to the use of non-overlapping covariance descriptor which takes advantage of discriminative information such as statistical information and different features. When the coke can moves to the camera, it becomes large. Our tracker still captures the target well because the scale template involved and template update schema.

Occlusion and pose variation. The trackers are facing occluded challenge in the FaceOcc2, Tiger1 and Woman. For the FaceOcc2 sequence, CT, MIL trackers using Haar-like feature starts to drift with the book when moving up and down, then drift problem becomes more and more serious (see frames #442, #500, #695). The woman sequence has non-rigid deformation and heavy partial occlusion (see frames #130, #222, #567). Our tracker and DFT are able to achieve favorable tracking results on this sequence both in terms of accuracy and success rate. Our algorithm performs better than other five trackers especially after long frames in Woman sequence. As non-overlapping descriptor exploits a large number of regions, the spatial information between regions is maintained, and benchmark template could also facilitate the final result. Therefore, the visible part is effectively represented by non-overlapping sparse reconstruction.

Background clutters and rotation. In the Skating and Woman sequences of Fig. 4, the target object moves in clutter backgrounds. All the trackers succeed in tracking the target object before out-of-plane pose change occurs in the Skating sequences (see frames #70, #205). There are also multiple objects similar to the target in the Skating sequence. As these objects move around the tracking target in different direction, it also poses a background clutters challenge for visual tracking. Nevertheless, our tracker is able to relocate the target while all the other methods lose track of the target gradually. It can be explained by that the adopted appearance model based on sparse representation and our template scheme contribute much to the good performance.

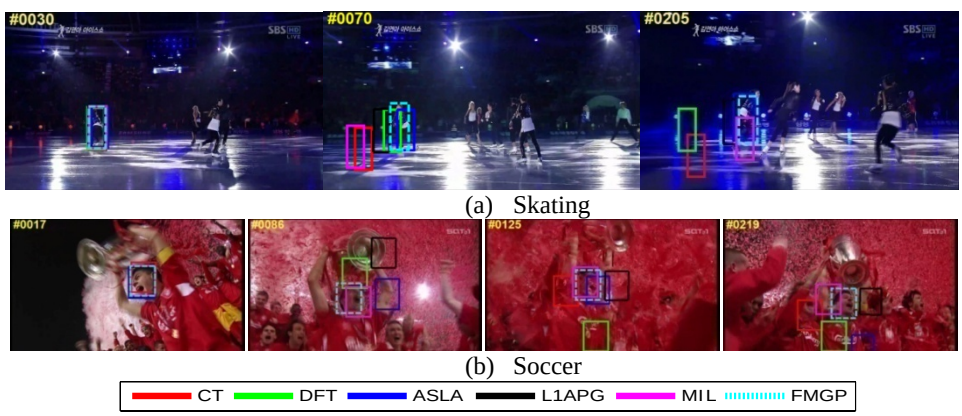


Fig4. Some tracking results of the sequences Skating and Soccer.

From the comparisons above, our FMGP can alleviate the drift problem mainly due to adaptive template update strategy, more robust features for target representation, and accurate similarity metric. For instance, in the skating sequence, our tracker successfully

tracks the target most of the time because of the adoption of a benchmark template which provides the resistance to the drifting problem.

6 CONCLUSION

This paper presents an efficient fast memory gradient pursuit tracking method which replaces the l1 norm convex optimization. In this work, we also exploit the inherent discrimination of non-overlapping descriptor and replace costly similarity measurements on manifolds with simple distance computations in higher dimensional Euclidean space. In order to prevent the drift tracking, template sets and the update scheme are also well designed. Experimental results on challenging video sequences show that our tracker achieves favorably against several state-of-the-art algorithms. In future, we will take speedup in scale update and customized greedy pursuit algorithm into consideration.

REFERENCE

1. B. Boris, M.H. Yang, and S.Belongie, "Visual tracking with online multiple instance learning," *Computer Vision and Pattern Recognition*, 2009, pp. 983-990.
2. S. L. Laura and E. L. Miller, "Distribution fields for tracking," *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1910-1917.
3. A. Smeulders, D. Chu, R. Cucchiara, S. Dehghan, and M. Shah, "Visual Tracking: An Experimental Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.1, 2013.
4. J. Xu, H. Lu, and M-H Yang, "Visual tracking via adaptive structural local sparse appearance model," *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1822-1829.
5. D. A. Ross, J. L, R-S. L, and M-S. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, Vol. 77, 2008, pp. 125-141.
6. M. Xue and H.B. Ling, "Robust visual tracking using l1 minimization," *IEEE 12th International Conference on Computer Vision*, 2009, pp. 1436-1443.
7. K. H. Zhang, L. Zhang, and M-H. Yang, "Real-time compressive tracking," *ECCV 2012 Computer Vision*, 2012, pp. 864-877.
8. H. Cheng, Z. Liu, L. Yang, and X. Chen, "Sparse representation and learning in visual recognition: Theory and applications," *Signal Processing*, Vol. 93, 2013, pp. 1408-1425.
9. M. Xue, H. Ling, Y. Wu, E. Blasch, and L.Bai, "Robust visual tracking and vehicle classification via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.33, 2011, pp.2259-2272.
10. S.P. Zhang, H.X. Yao, H. Y. Zhou, and S.H. Liu, "Robust visual tracking based on online learning sparse representation," *Neurocomputing*, Vol.100, 2013, pp. 31-40.
11. H. X. Li, C.H. Shen, and Q. Shi, "Real-time visual tracking using compressive sensing," *Computer Vision and Pattern Recognition*, 2011, pp. 1305-1312.
12. M. Xue, H. Ling, Y. Wu, E. Blasch, and L. Bai, "Minimum error bounded efficient l 1 tracker with occlusion detection," *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1257-1264.

13. C.Bao, Y. Wu, and H.Ling, "Real time robust l1 tracker using accelerated proximal gradient approach," *Computer Vision and Pattern Recognition* ,2012, pp. 1830-1837.
14. H.Y. Cheng, and J.N. Hwang, "Adaptive particle sampling and adaptive appearance for multiple video object tracking," *Signal Processing*, Vol.89, 2009, pp. 1844-1849.
15. M. Toivanen and J. Lampinen, "Incremental object matching and detection with Bayesian methods and particle filters," *IET Computer Vision*, Vol.5, 2011, pp. 201-210.
16. P. Chavali and A. Nehorai, "Hierarchical particle filtering for multi-modal data fusion with application to multiple-target tracking," *Signal Processing*, Vol.97, 2014, pp. 207-220.
17. C. Xie, J. Tan, P. Chen, J. Zhang, and L. H, "Multiple instance learning tracking method with local sparse representation," *IET Computer Vision*, Vol.7, 2013, pp. 320-334.
18. P. Song, and H. Liu, "Robust Visual Tracking Using Incremental Sparse Representation," *Chinese Intelligent Automation Conference*, 2013, pp.691-698.
19. B. Ma, H. Hu, S. Liu, and J. C, "Robust Visual Tracking Using Local Sparse Covariance Descriptor and Matching Pursuit," *International Conference on Neural Information Processing*, 2013, pp. 485-492.
20. O.Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," *ECCV Computer Vision*, 2006, pp. 589-600.
21. F. Porikli, O. Tuzel, and P. Meer, "Covariance tracking using model update based on lie algebra," *Computer Vision and Pattern Recognition*, Vol. 1, 2006, pp.728-735.
22. V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Geometric means in a novel vector space structure on symmetric positive-definite matrices," *SIAM journal on matrix analysis and applications*, Vol.29, 2007, pp. 328-347.
23. S. Kluckner, T. Mauthner, and H. Bischof, "A Covariance Approximation on Euclidean Space for Visual Tracking," *AAPR/OAGM*, 2009.
24. S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," *International Society for Optics and Photonics*, 1997, pp. 182-193.
25. S. J. Julier, "The scaled unscented transformation," *in Proceedings of Conference on American Control*, Vol. 6, 2002, pp. 4555-4559.
26. S.J Julier, J.K. Uhlmann, "A general method for approximating nonlinear transformations of probability distributions," Technical report, Robotics Research Group, Department of Engineering Science, 1996.
27. L.B. Dorini, and S K .Goldenstein. "Unscented feature tracking," *Computer Vision and Image Understanding*, 2011, Vol.115, pp. 8-15.
28. X. Zhang, W. Li, W.Hu, H. Ling, and S. Maybank, "Block covariance based l1 tracker with a subtle template dictionary," *Pattern Recognition*, Vol.46, 2013, pp.1750-1761.
29. Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," *Signals, Systems and Computers, in Proceedings of Conference Record of The Twenty-Seventh Asilomar*, 1993, pp.40-44.
30. D. Needell and R. Vershynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *Foundations of computational mathematics*, Vol.9, 2009, pp. 317-334.

31. Q. Guo and C.W, “Image reconstruction algorithm of compressed sensing based on memory gradient pursuit,” *Journal of Image and Graphics*, Vol.19, 2014, pp. 670-676.
32. L. Grippo, F. Lampariello, and S.Lucidi. “A nonmonotone line search technique for Newton's method,” *SIAM Journal on Numerical Analysis*, Vol.23, 1986, pp. 707-716.
33. G. H. Golub and C.F. V. Loan, Matrix computations, The Johns Hopkins University Press, 1996, pp.520-532.
34. J. Triesch and C. V. D. Malsburg, “Democratic integration: Self-organized integration of adaptive cues,” *Neural computation*, Vol.13, 2001, pp. 2049-2074.
35. Y. Wu, J. Lim, and M-H Yang, “Online object tracking: A benchmark,” IEEE Conference on Computer Vision and Pattern Recognition, 2013.
36. B. Ma, H. Hu, S. Liu, and J. Chen, “Robust Visual Tracking Using Local Sparse Covariance Descriptor and Matching Pursuit,” *Neural Information Processing*, 2013, pp. 485-492.

Qiang Guo is a Ph.D. student in College of Information Science and Engineering of Northeastern University. He received M.Sc. degree in 2007 from Northeastern University. Now he is currently a lecturer in National Police University of China. His main research focuses on video image processing, pattern recognition.

Chengdong Wu received M.S. degree in automatic control from Tsinghua University in 1988. He received Ph.D. degree in automatic control from Northeastern University in 1994. Now he is currently a professor in College of Information Science and Engineering, Northeastern University. His current research focuses on image processing, wireless sensor networks, home automation and robot.