Parallel Search Strategy in Kernel Feature Space to Track FLIR Target

Zhen Shi, Chang'an Wei*, Junbao Li, Ping Fu, Shouda Jiang

Department of Automatic Testing and Control

Harbin Institute of Technology, Harbin 150080, China

Abstract

Robust tracking in the forward looking infrared (FLIR) sequences is still a challenging problem in the field of computer vision. Because images acquired by the infrared sensors are characterized by low signal-to-clutter ratios(SCR) and targets of interest may exhibit profound appearance variations due to ego-motion of the sensor platform and complex maneuvers. Though many efforts have been delivered, there are still some issues to be addressed. First, intensity features is not enough to deal with complex appearance variations for the challenging sequence. Second, to obtain satisfying estimation of target state, plenty particles have to employed to approximate its probability density function (pdf). To deal with the two problems, a parallel search strategy based on kernel sparse representation (KLIPS tracker) is proposed to perform the tracking task in the FLIR sequences. With the ability of capturing the nonlinear features, kernel method is introduced to deal with complex appearance variations. After the kernel function is constructed based on the histogram features, both the target templates and candidates are mapped into the kernel feature space. Then efficient state particles are selected based on sparse representation which can be used to estimate the target state. The proposed method is tested on the AMCOM database, and the experimental results demonstrate its excellent performance in tracking accuracy compared with some state-of-the-art trackers.

Keywords: sparse representation; kernel method; FLIR target tracking

1. Introduction

Recently, the production cost of infrared sensors is reduced obviously due to technological advancements, thus it can be employed in more application fields. For infrared images, the intensity of an object is determined by its

*Corresponding author.

Email address: weichangan@hit.edu.cn (Shouda Jiang)

temperature and radiated heat which is not influenced by illumination conditions and surface features. Specially, attention has been specifically devoted to infrared sensors which are sensitive to long-wave infrared (LWIR) light in the range 8-12 μm . It is very useful in many practical applications because they can see heat sources at night or through smoke, fog, haze, etc [1].

Object tracking is a fundamental task which has been well studied in the last decades and numerous approaches have been proposed [2]. Generally speaking, it can be categorized into two types. The first category is free tracking, where initialization of the target on the first frame is required, and tracking is carried out based on information in current and past frames [3]. The second category is specific tracking, in which a pre-trained object detector is needed to obtain the desired target, and information in both past and future frames can be used for tracking [4]. For practical application, the target in FLIR sequence is desired to be locked since it has been detected, thus it belongs to the first category obviously. In fact, target tracking has remained a challenging problem in the field of computer vision and special treatments are necessary to design a tracker for FLIR image sequences [5]. In this paper, two issues are discussed about tracking a target in the FLIR sequences which are stated as follows.

One important issue in the tracking system is target representation. For a target of interest in the complex scenes, its appearance may exhibit profound variations over relatively short time scales. Intrinsic factors include pose change and shape deformation while extrinsic factors include illumination change, ego-motion of camera and partial occlusions. In fact, it is still a challenging problem to deal with the unexpected appearance variations. Obviously, it is not reliable to track a target in FLIR sequences only with intensity information, and more sophisticated techniques are required for representing the target appearance to deal with various challenging problems. It is desired to be robust to pose change, shape deformation, illumination change, partial occlusion, etc.

The other issue is target state estimation in the particle filter framework. At present, there are two ways to estimate the target state after the particles with associated weights are available. For the first method, the most similar state particle is taken as the result directly. In fact, the target state is a continuous variable whose pdf is approximated with discrete particles. Obviously, a reliable estimation can be obtained only when the particle number is large enough. For the other method, the target state is estimated by all the weighted particles. However, not each candidate contains the target while background is dominate in the false candidates. In fact, only those candidates containing the target

are corresponding to the efficient state particles. When the target state is estimated with all the weighted particles, it is inevitable to introduce error which will cause drift problem.

To solve the two issues mentioned above, a parallel search strategy is proposed in the kernel feature space. First, kernel method is introduced to deal with complex appearance variations. After the kernel function is constructed based on the histogram features, both the target templates and candidates are mapped into the kernel feature space. Then efficient state particles are selected based on sparse representation which can be used to estimate the target state. The block diagram of proposed method is shown in Fig.1

Compared to exciting literature, the distinctive contributions of this paper are summarized as follows.

1) For a target with profound variations, we propose to model its appearance with kernel method which is efficient to capture the nonlinear similarity of features. Considering the target characteristic in infrared images, the kernel function is constructed with histogram features which are not sensitive to part occlusion, illumination variation and shape deformation.

2) When the target state pdf is approximated with weighted particles, it is not reliable to estimate its state only with the most similar one. Moreover, it is inevitable to introduce error when all the particles are employed. Considering the present methods, we propose to parallel search for the efficient particles based on sparse representation which are fused to obtain the target state. In this way, the estimation accuracy can be improved and satisfying tracking performance can be obtained.

The rest of this paper is organized as follows. The related works is reviewed briefly in Section 2. After the principle of particle filter is introduced in Section 3, a parallel search strategy in the kernel feature space is proposed in Section 4. In Section 5, the solution to kernel sparse representation is obtained based on constructed kernel matrixes and vectors. Experiments results are given in Section 6 then the paper is concludes in Section 7.

2. Related work

There are plenty methods to represent the target appearance in the previous work. The sum of squared difference (SSD) is used as a cost function in the tracking problem [6] and the mixture model is used to model the appearance variation [7]. An appearance-adaptive model is incorporated in the particle filter framework to realize robust visual

Figure 1: Block diagram of KL1PS tracker

tracking and classification algorithms [8]. The target appearance can also be represented with an eigenspace [9], affine warps of learned linear subspaces [10], or an adaptive low dimensional subspace [11]. Besides, the appearance is modeled as a structured combination of the subspaces learned by partial least squares analysis [12]. Due to the powerful feature learning characteristic of deep learning, target appearance can be observed by multiple sub-regions and each region is observed by a deep learning model [13]. Histogram of the pixel intensities has been adopted in the appearance models of several recent mean-shift trackers [14]. Besides, histogram of the local standard deviation has also been used for mean-shift tracking of infrared targets [15]. The histogram features are widely used due to their simplicity and efficiency as well as the scale and rotation invariance properties. For the target representations based on histogram, appearance learning is generally accomplished by iteratively updating a reference histogram. An extension of the simple histogram-based appearance learning strategies maintains explicit appearance models for both the target and the surrounding background which has been used to combat the drifting problem [16] [17].

Generally speaking, there are two categories of methods to search for the target. In the first category, tracking is formulated in the optimization framework with an objective function, and gradient descent methods can be used for locating the target. For example, the first order Taylor expansion is used to linearize the nonlinear cost function, and the motion parameters are estimated iteratively [18]. Further, mean shift method is used for searching the target by using the Bhattacharyya coefficient as the similarity metric with kernel-regularized histograms [14]. A discriminative

approach is adopted to identify spatial attentional regions with a gradient-based formulation to locate objects [19]. A tracking algorithm based on distribution fields is proposed which allow smoothing the objective function without blurring the image, and the target is located by searching for the local minimum by using a coarse-to-fine strategy [20]. However, objective functions for tracking are usually nonlinear with many local minima. To alleviate this problem, dense sampling methods have been adopted at the expense of a high computational load [21] [22] [23]. Thus stochastic search algorithms such as particle filter have been widely used since they are relatively insensitive to the local minimum and are computationally efficient [24] [25]. Recently, methods based on particle filter have been developed using effective observation models which have demonstrated their success [26] [27].

Recently, numerous tracking methods based on sparse representations(SR) have been proposed. In the L1 tracker [26], target tracking is formulated to solve plenty ℓ_1 norm minimization problems. Each candidate target is represented sparsely with a dictionary composed of target templates and trivial templates, then target candidate with the smallest reconstruction error is treated as the optimal one. To handle occlusion better, local sparse representation has been introduced into tracking [28]. For run-time efficiency, a minimal error bounding strategy is introduced to reduce the number of ℓ_1 norm minimization problems [29]. Besides, accelerated proximal gradient approach is introduced to solve ℓ_1 norm minimization problems [30]. To enhance tracking robustness, a local sparse appearance model is proposed with the mean shift algorithm to locate target [31]. Given that the representation of particles is jointly sparse, tracking can be formulated as a multi-task sparse learning problem [32]. A collaborative tracking algorithm is proposed which combines a sparsity-based discriminative classifier and a sparsity-based generative model [33]. Besides, sparse coding of local image patches with spatial layout in target is used to model the appearance for tracking [27]. To deal with outliers for target tracking, a least soft threshold squares algorithm is proposed by modeling image noise with the Gaussian-Laplacian distribution other than the trivial templates [34].

At present, most methods only focus on capturing features in the original space which is not enough to deal with complex appearance variances. Thus kernel method has been introduced as an elegant way to solve nonlinear problems in the kernel feature space [35]. Some common kernel functions include linear kernel, polynomial kernel, Gaussian radial basis function (RBF) kernel and wavelet kernel [36] [37]. After the samples are mapped into a high dimensional feature space by a nonlinear function, it means that they are redistributed. If the kernel function is utilized

appropriately, a test sample can be represented more accurately as the combination of training samples from the same class in the kernel feature space. There have been plenty work which combine the sparsity-induced techniques and kernel methods, such as 1-norm SVM [38], sparse kernel regressor [39], kernel sparse representation (KSR) [40] and so on. For kernel sparse representation-based classifier (KSRC), the nonzero entries of the test sample coefficient will be more associated with training samples from the same class. It means that the nonlinear similarity of features may be captured more easier in the kernel feature space with high dimension [41].

3. Particle Filter Principle

When the target state is available in current frame, it is desired to locate it in the subsequent frames. From the viewpoint of statistics, tracking can be conducted to estimate the posterior probability of target state. For most practical applications, a kinetic model of the maneuvering target is nonlinear and the target state pdf is non-Gaussian. Particle filter [24] is a recursive Bayesian filter using Monte Carlo (MC) simulations, which generalizes the traditional Kalman filter. In the particle filter framework, there are two steps of prediction and update. When the number of particles is sufficiently large, the posterior pdf of the target state can be approximated well irrespective of its underlying distribution.

Let x_t denote the target state at frame t , thus, the predicting distribution given all available observations $y_{1:t-1} = \{y_1, y_2, \dots, y_{t-1}\}$ up to frame $t - 1$ can be computed recursively as follows

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1}. \quad (1)$$

After observation y_t is available at frame t , the target state pdf can be updated with the Bayesian principle as follows

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})}, \quad (2)$$

where $p(y_t|x_t)$ denotes the observation probability.

Then the posterior pdf of the target state can be approximated by using N particles $X_t = \{x_t^1, x_t^2, \dots, x_t^N\}$ with the associated weights $W_t = \{w_t^1, w_t^2, \dots, w_t^N\}$. The state particles are generated by using a sequential importance

118 distribution $q(x_t|y_{1:t}, x_{1:t-1})$ and their weights can be updated as follows

$$w_t^i = w_{t-1}^i \frac{p(y_t|x_t^i)p(x_t^i|x_{t-1}^i)}{q(x_t|x_{1:t-1}, y_{1:t})}. \quad (3)$$

119 For the bootstrap filter, $q(x_t|x_{1:t-1}, y_{1:t}) = p(x_t|x_{t-1})$ and the weight has a simple form of $w_t^i \propto w_{t-1}^i p(y_t|x_t^i)$.

120 However, the variance of the weights increases over time, therefore most particles only have negligible weights
 121 after several iterations. In other words, most computational cost is spent on updating weights of particles that are
 122 useless for estimating the target state. To avoid such a degeneracy phenomenon, it is necessary to resample according
 123 to weights, thus a set of new state particles with equal weights are obtained in each frame.

124 4. Parallel Search Strategy in Kernel Feature Space

125 The kernel method is a well-known technique in machine learning which can capture the nonlinear similarity to
 126 reduce the feature quantization error. With regard to the complex variances of target appearance, the observation is
 127 performed after both target templates and candidates are mapped into the kernel feature space. Then only several
 128 efficient state particles are selected to estimate the target state.

129 4.1. State Transition Model Based on Affine Transformation

130 When an affine transformation is employed to describe target area, the state vector x_t can be modeled by six
 131 parameters $x_t = (a_1, a_2, a_3, a_4, a_5, a_6)$, where (a_1, a_2, a_3, a_4) denote the deformation and (a_5, a_6) represent the position.
 132 Given that these six parameters are independent of each other, the state transition model $p(x_t|x_{t-1})$ can be obtained as
 133 follows

$$p(x_t|x_{t-1}) = N(x_t; x_{t-1}, \sigma^2), \quad (4)$$

134 where $N(\cdot)$ denotes a Gaussian distribution with the mean value x_{t-1} and covariance σ^2 . Thus, state particles at frame
 135 t can be obtained as follows

$$(a_1, a_2, a_3, a_4, a_5, a_6)_t^i = (a_1, a_2, a_3, a_4, a_5, a_6)_{t-1}^i + \varepsilon, \quad (5)$$

136 where ε denotes the Gaussian white noise with covariance σ^2 .

4.2. Observation Model in Kernel Feature Space

After state particles are available, the corresponding areas denoting target candidates $Y_t(i) (i = 1, \dots, N)$ can be normalized with the same size of target templates $T_t(i) (i = 1, \dots, n)$. In fact, not each candidate area contains the target while background is dominate in the false candidates. Moreover, only those areas containing the target are determined by the efficient state particles which should be used to estimate the target state. Motivated by the fact that kernel method can capture the nonlinear similarity of features to make samples more separable, both the target templates and candidates are mapped from original space χ ($\chi \subset \mathbb{R}^d, d = w \times h$) to high-dimensional kernel feature space F ($F \subset \mathbb{R}^D$) through a nonlinear function φ .

Given that similar samples lie in the same subspace, each target template can be linearly represented by target candidates in the kernel feature space F as

$$\varphi(T_t(i)) = \varphi(Y_t)\alpha_i. \quad (6)$$

Then sparse constraints can be enforced on the coefficient α_i , thus only several true ones are selected from all the target candidates simultaneously. The above method can be realized as follows

$$\hat{\alpha}_i = \arg \min_{\alpha_i} \|\alpha_i\|_0 \quad s.t. \quad \varphi(T_t(i)) = \varphi(Y_t)\alpha_i, \quad (7)$$

where $\|\cdot\|_0$ denotes the ℓ_0 norm counting the number of nonzero entries in a vector. Recent development in the emerging theory of sparse representation and compressed sensing reveals that if the solution sought is sparse enough, it is equal to that of the ℓ_1 norm minimization problem [42]. Thus the solution can be obtained by solving an ℓ_1 -regularized least-squares problem as

$$\hat{\alpha}_i = \arg \min_{\alpha_i} \frac{1}{2} \|\varphi(T_t(i)) - \varphi(Y_t)\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1. \quad (8)$$

However, nonzero entries of the coefficient can be any real numbers without restrictions, thus candidates with reversed intensity pattern may be selected. It is necessary to impose nonnegative constraints on the coefficient to eliminate such false candidates and the solution can be obtained by the following form as

$$\hat{\alpha}_i = \arg \min_{\alpha_i} \frac{1}{2} \|\varphi(T_t(i)) - \varphi(Y_t)\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1 \quad s.t. \quad \alpha_i \succcurlyeq 0. \quad (9)$$

4.3. Target State Estimation

For each target template, after the coefficient $\hat{\alpha}_i$ is available, its indexes of nonzero entries determine the efficient state particles and their values reflect corresponding weights. The template number n should be set properly to achieve a balance between computational efficiency and tracking effectiveness. For a reliable template $T_i(i)$, it can be represented with only a few target candidates. This implies that the coefficient $\hat{\alpha}_i$ is sparse with large nonzero entries. Thus, the weight of the i th target template $w_T(i)$ can be updated as $\hat{\alpha}_i(j^*)$, where $j^* = \arg \max_j \hat{\alpha}_i(j)$. With respect to the weight of each target template, all the coefficients of the target templates can be fused as follows

$$\hat{\alpha} = \frac{1}{n} \sum_{i=1}^n w_T(i) \times \hat{\alpha}_i. \quad (10)$$

Then the coefficient $\hat{\alpha}$ is normalized as follows

$$w_X(i) = \frac{\hat{\alpha}(i)}{\sum_{i=1}^N \hat{\alpha}(i)}, \quad (11)$$

where $\sum_{i=1}^N w_X(i) = 1$ and the nonzero entries of w_X are taken as the corresponding weights of efficient state particles.

Thus the target state can be estimated as follows

$$\hat{x}_t = \sum_{i=1}^N w_X(i) \times x_t^i. \quad (12)$$

The comparison of three ways to estimate the target state is shown in Fig.2.

Figure 2: Comparison of three ways to estimate the target state. (a) the most similar state particle. (b) all state particles. (c) efficient state particles.

4.4. Template Update

Intuitively, the target appearance may remain unchanged only for a certain period, therefore it is necessary to update the target templates. However, it is inevitable to introduce errors when the target templates are updated with

170 the tracking result. If the target templates are updated too frequently, the tracker can easily drift because of the
 171 accumulated errors.

172 To solve this problem, the target templates can be updated dynamically. The similarity between the i th target
 173 template and the tracking result can be measured by the using function $\cos(\theta_i)$ where θ_i denotes the angle between
 174 them. A suitable threshold θ_{thr} is set to determine whether the target template should be updated. Once the maximal
 175 angle $\theta_{max} = \max_i \theta_i (i = 1, \dots, n)$ exceeds the threshold θ_{thr} , the target template should be replaced by the tracking
 176 result.

177 The parallel search strategy in kernel feature space is summarized in Alg.1.

Algorithm 1 The parallel search strategy in kernel feature space

Input:

current frame F_t , state particle set X_{t-1} , target template set T_{t-1} ;

Output:

target state $\hat{x}_t$, state particle set X_t , target template set T_t ;

```

1: for  $t = 1$  to the number of frames do
2:   New state particles  $X_t$  are drawn from  $x_{t-1}^i$  based on (5);
3:   for  $i = 1$  to the number of target templates do
4:     Efficient state particles can be selected based on (9) whose solution denotes the associated weights;
5:     The  $i$ th target template weight  $w_T(i)$  can be updated as  $\hat{\alpha}_i(j^*)$ , where  $j^* = \arg \max_j \hat{\alpha}_i(j)$ ;
6:   end for
7:   All the coefficients corresponding target templates are fused based on (10) then normalized as (11);
8:   The target state  $\hat{x}_t$  can be estimated by the efficient state particles with associated weights as (12);
9:   The angle between target template and tracking result are computed and the maximal one is  $\theta_{max}$ ;
10:  if  $\theta_{max} > \theta_{thr}$  then
11:    Target template with the maximal angle  $\theta_{max}$  is replaced by the tracking result;
12:  end if
13: end for

```

5. Solution to Kernel Sparse Representation

After the kernel functions are available, the parallel search strategy can be performed in kernel feature space F . In this part, the kernel functions are constructed based on the histogram features which are not less sensitive to part occlusion, illumination variation and shape deformation.

5.1. Dimensionality Reduction

In fact, the dimensionality of kernel feature space F is very high or even infinite, thus it is necessary to perform reduction. It means that samples from F can be projected into low-dimensional subspace by a projection matrix P . In this way, eq(9) can be modified as follows

$$\hat{\alpha}_i = \arg \min_{\alpha_i} \frac{1}{2} \|P^\top \varphi(T_t(i)) - P^\top \varphi(Y_t) \alpha_i\|_2^2 + \lambda \|\alpha_i\|_1 \quad s.t. \quad \alpha_i \succcurlyeq 0. \quad (13)$$

Then KPCA can be used to perform dimensionality reduction and each projection vector is a linear combination of target candidates in F as

$$P_j = \sum_{i=1}^N c_{j,i} \varphi(Y_t(i)), \quad (14)$$

where $c_j = [c_{j,1}, \dots, c_{j,n}]$ is the coefficient corresponding to the j th projection vector P_j . Let $C = [c_1, \dots, c_{\tilde{d}}]$ denote the projection coefficient matrix and $\tilde{d}$ present the number of projection vectors, then the projection matrix can be expressed as $P = \varphi(Y_t)C$. Substituting it into (14), then we get

$$\hat{\alpha}_i = \arg \min_{\alpha_i} \frac{1}{2} \|C^\top k(\cdot, T_t(i)) - C^\top K \alpha_i\|_2^2 + \lambda \|\alpha_i\|_1 \quad s.t. \quad \alpha_i \succcurlyeq 0, \quad (15)$$

where $k(\cdot, T_t(i)) = \varphi(Y_t)^\top \varphi(T_t(i))$ denotes the kernel vector and $K = \varphi(Y_t)^\top \varphi(Y_t)$ presents the kernel matrix. Besides, $C^\top k(\cdot, T_t(i))$ and $C^\top K$ denote target templates and candidates respectively, then zero-mean-unit-norm can be performed after dimension reduction.

For KPCA, the projection coefficient matrix are normalized eigenvectors corresponding to nonzero eigenvalues which are the solutions of the following eigenvalue problem

$$n\lambda c = Kc, \quad (16)$$

where c_j are normalized such that $\lambda_j c_j^\top c_j = 1$. We take $\tilde{d}$ eigenvectors corresponding to the first $\tilde{d}$ largest eigenvalues

$\lambda_j, j = 1, \dots, \tilde{d}$, where $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{\tilde{d}}$ and $C = [c_1, \dots, c_{\tilde{d}}] \in \mathbb{R}^{n \times \tilde{d}}$.

For the kernel matrix K , each of its element denotes the similarity between two target candidates which is determined by the kernel function $\kappa(\cdot, \cdot)$. Thus the kernel vector can be defined as

$$K = \begin{bmatrix} k(Y_t(1), Y_t(1)) & k(Y_t(1), Y_t(2)) & \cdots & k(Y_t(1), Y_t(n)) \\ k(Y_t(2), Y_t(1)) & k(Y_t(2), Y_t(2)) & \cdots & k(Y_t(2), Y_t(n)) \\ \vdots & \vdots & \ddots & \vdots \\ k(Y_t(n), Y_t(1)) & k(Y_t(n), Y_t(2)) & \cdots & k(Y_t(n), Y_t(n)) \end{bmatrix}. \quad (17)$$

Similarly, each element of the kernel vector $k(\cdot, T_t(i))$ denotes the similarity between the target candidate and target template. Thus the kernel vector can be defined as

$$k(\cdot, T_t(i)) = [k(Y_t(1), T_t(i)), \cdots, k(Y_t(n), T_t(i))]^\top. \quad (18)$$

After both the kernel matrix K and kernel vector $k(\cdot, T_t(i))$ are available, the eq(15) can be solved as ℓ_1 norm minimization problem.

5.2. Construction of Kernel Matrix and Kernel Vector

In this part, we will discuss how to construct the kernel matrix K and kernel vector $k(\cdot, T_t(i))$. In fact, the key is kernel function $\kappa(\cdot, \cdot)$ which denotes the similarity between two samples in the kernel feature space. After a special feature is selected, the similarity can be measured by inner production. For maneuvering targets in the FLIR sequences, it is not suitable to track a target with holistic features based on raw intensity. Thus efficient features are desired to describe the target area which can perform robust tracking task.

In this paper, the target appearance is described with histogram features which are not less sensitive to part occlusion, illumination variation and shape deformation. With regard to the low SCR of FLIR sequences, the local standard deviation feature is fused with intensity feature. Spatial information can be added to the histogram features with isotropic kernel (Parzen window) and some popular kernels include Gaussian kernel, Triangular kernel, Bi-weight kernel, Epanechnikov kernel, etc. Among these kernels, the Epanechnikov kernel yields the best minimum mean integrated square error between two kernel densities in the continuous domain which is selected in this paper.

The above mind can be formulated as follows. For an area S with centroid c , the intensity value of pixel at position r is mapped to the u th bin of quantized feature space by function $b : \mathbb{R}^2 \rightarrow \{1, \cdots, m\}$. Then the intensity histogram

feature f_{int} with Parzen window K_H can be obtained by

$$f_{int}(u) = \lambda_1 \sum_{r \in S} K_H(r - c) \delta[b(r) - u], \quad (19)$$

where δ is the Kronecker delta function. Besides, λ_1 is a normalization constant such that $\sum_{u=1}^m f_{int}(u) = 1$.

In a similar manner, local standard deviation histogram feature f_{lsd} with Parzen window K_H can be obtained by

$$f_{lsd}(u) = \lambda_2 \sum_{r \in S} K_H(r - c) \delta[b(r) - u], \quad (20)$$

where λ_2 is a normalization constant such that $\sum_{u=1}^m f_{lsd}(u) = 1$.

After the histogram features are available, the kernel matrix and kernel vector can be constructed based on them.

Let K_{int} and K_{lsd} be kernel matrices obtained based on the intensity histogram and local standard deviation histogram respectively, then Bhattacharyya coefficient is used to measure the similarity. Take intensity kernel matrix K_{int} for example, each element can be obtained by

$$K_{int}(i, j) = BhaCoeff(f_{int}(Y_t(i)), f_{int}(Y_t(j))), \quad (21)$$

where $f_{int}(\cdot)$ is the function to get the intensity histogram of an area, and $BhaCoeff(\cdot, \cdot)$ is the Bhattacharyya coefficient function. In a similar manner, K_{lsd} can be obtained by

$$K_{lsd}(i, j) = BhaCoeff(f_{lsd}(Y_t(i)), f_{lsd}(Y_t(j))). \quad (22)$$

where $f_{lsd}(\cdot)$ is the function to get the local standard deviation histogram of an area.

Similarity, let $k_{int}(\cdot, T_t(i))$ and $k_{lsd}(\cdot, T_t(i))$ be kernel vectors obtained based on the intensity histogram and local standard deviation histogram respectively. Take intensity kernel vector $k_{int}(\cdot, T_t(i))$ for example, each of its element can be obtained by

$$k_{int}(j, T_t(i)) = BhaCoeff(f_{int}(Y_t(j)), f_{int}(T_t(i))). \quad (23)$$

In a similar manner, local standard deviation kernel vector $k_{lsd}(\cdot, T_t(i))$ can be obtained by

$$k_{lsd}(j, T_t(i)) = BhaCoeff(f_{lsd}(Y_t(j)), f_{lsd}(T_t(i))). \quad (24)$$

Two histogram features are selected to construct the kernel functions, and fusion is necessary to complement each other. One simple yet efficient way is to set equal weights for them, then the fused kernel matrix can be obtained as

$$K = \frac{1}{2}(K_{int} + K_{lsd}). \quad (25)$$

235 Similarly, the fused kernel vector can be obtained as

$$k(\cdot, T_t(i)) = \frac{1}{2}(k_{int}(\cdot, T_t(i)) + k_{lsd}(\cdot, T_t(i))). \quad (26)$$

236 The proposed method to solve kernel sparse representation is summarized in Alg.2.

Algorithm 2 Solution to kernel sparse representation

Input:

current frame F_t , target candidates Y_t , target templates T_t ;

Output:

the coefficient $\hat{\alpha}_i$;

- 1: For both target templates and candidates, two histogram features can be extracted based on (19) and (20);
 - 2: Two kernel matrices can be obtained based on the histogram features as (21) and (22);
 - 3: Two kernel vectors can be obtained based on the histogram features as (23) and (24);
 - 4: The projection matrix P can be defined as (14) and the coefficient matrix C can be obtained based on (16);
 - 5: The kernel matrices and vectors are fused as (25) and (26) respectively;
 - 6: The columns of $C^\top K$ and $C^\top k(\cdot, T_t(i))$ are normalized with zero-mean-unit-norm;
 - 7: The ℓ_1 norm minimization problem (15) is solved to obtain the coefficient $\hat{\alpha}_i$;
-

237 **6. Numerical Experiment**

238 Our method is evaluated on the AMCOM FLIR sequences in grayscale format (128×128 pixels). The sequences
 239 selected exemplify most of the typical challenges including low SCR, significant pose change, size variations, ego-
 240 motion of the camera platform, strong background clutter, etc.

241 The number of target templates is 10 which is a trade-off between computational efficiency and effectiveness.
 242 In the particle filter framework, 100 particles with associated weights are used to approximate the target state pdf.
 243 Both target templates and candidates are normalized to have the same size 8×8 , and the variances of the affine
 244 parameters are (0.05, 0.0005, 0.0005, 0.05, 1, 1). Moreover, 16 bins is set to construct distributions of the intensity and
 245 the standard deviation measures.

The method proposed is evaluated on the personal computer with 2.67GHz Inter(R) Core(TM) I3 CPU and 2GB memory. Further, Windows 7 and Matlab R2010a are installed. The ℓ_1 norm minimization problem is solved with SPArse Modeling (SPAM) Software package [43]. Extensive experiments demonstrate the performance of the proposed method, then we compare it with several state-of-the-art trackers including CSK [44], CT [45], DFT [20], Frag [46], IVT [11], L1 [26], MTT [47] and ORIA [48].

6.1. Qualitative Results

The first experiment is performed on the sequence of LW-15-NS and the results are shown in Fig.3. A stationary target stays at the center position and the infrared sensor comes close to it. Besides, low SCR of the sequence is also a challenging problem for the tracking task. At first, the IVT tracker drifts from the target then the CT, DFT and Frag trackers also fails when the target size becomes larger with time.

The second experiment is performed on the sequence of LW-14-15 and the results are shown in Fig.4. In a sequence with low SCR, the infrared sensor with strong ego-motion comes close to a stationary target. It can be seen that all the trackers can perform the tracking task throughout the sequence. However, when the target becomes larger with time, most of them can only lock part of the target due to appearance variations. The KL1PS tracker performances best because kernel method is introduced by the histogram features which are not sensitive to size change.

The third experiment is performed on the sequence of LW-16-08 and the results are shown in Fig.5. The size of the interesting target becomes larger with time which stays at the center position of images. Another target with similar appearance moves to the former one and partial occlusion happens when they are close enough. The IVT, MTT and DFT trackers lock on the false one after the two targets encounter, and the ORIA tracker drifts from the true target. Moreover, the L1 tracker also locks on the similar one for a moment then it recovers to track the interesting target again.

The forth experiment is performed on the sequence of LW-17-20 and the results are shown in Fig.6. A target moves along the road to turn right and both its size and pose have changed obviously. Besides, the illumination condition is varying with time which is also a challenging problem. It is obvious that all the trackers can track the target throughout the sequence. In fact, after the target turns right, its left part is similar to the target appearance in the former frames. It is obvious that the KL1PS tracker can lock on the main part of the target while only a small part

can be locked by the other trackers. For particle filter framework is employed in the KL1PS tracker, it is relatively insensitive to the local minimum. Besides, the KL1PS tracker can perform well also because histogram features are suitable to deal with pose change.

The fifth experiment is performed on the sequence of LW-18-16 and the results are shown in Fig.7. A dark target is moving toward the infrared sensor, and there is a similar target on the left side of the interesting one. At the beginning, the DFT tracker drifts from the target of interest to lock on the similar one. After the target size becomes larger, and the Frag, IVT, L1 and ORIA trackers fail due to obvious appearance variances.

The sixth experiment is performed on the sequence of LW-18-17 and the results are shown in Fig.8. There are two maneuvering targets in the scene and the left one is occluded by smog. The CT tracker drifts from the target of interest to lock on the similar one. Moreover, the CSK, Frag, IVT, DFT and L1PS trackers can not deal with the target appearance variance well caused by view angle which also fail to lock on the target throughout the sequence.

The seven experiment is performed on the sequence of LW-18-18 and the results are shown in Fig.9. A maneuvering target moves forwards in the complex background. Moreover, it is also challenging for the tracking task in a sequence with low SCR. After the target turn right, the Frag, IVT, DFT trackers fail to lock on the target due to obvious appearance variances.

The eighth experiment is performed on the sequence of LW-22-08 and the results are shown in Fig.10. A stationary target stays in the scene with low SCR, and all the trackers except for the Frag perform the tracking task well throughout the sequence. However, the KL1PS tracker can only lock part of the target when the infrared sensor comes close to it. It is caused by the histogram features which are employed to model the target appearance.

6.2. Quantitative Results

In this paper, the criterion of position error(PosErr) is used to evaluate all the trackers objectively. It measures the distance by pixels between the ground truth target center and the tracking result center which is defined as

$$PosErr = \sqrt{(c_{trk}^x - c_{gt}^x)^2 + (c_{trk}^y - c_{gt}^y)^2}, \quad (27)$$

where (c_{trk}^x, c_{trk}^y) and (c_{gt}^x, c_{gt}^y) denote the central position of tracking result and ground truth respectively. Ideally, the tracking result should fluctuate slightly around the ground truth which means that the tracker locks the target well.

Figure 3: Tracking results on the sequence LW-15-NS

Figure 4: Tracking results on the sequence LW-14-15

Figure 5: Tracking results on the sequence LW-16-08

Figure 6: Tracking results on the sequence LW-17-20

Figure 7: Tracking results on the sequence LW-18-16

Figure 8: Tracking results on the sequence LW-18-17

Figure 9: Tracking results on the sequence LW-18-18

Figure 10: Tracking results on the sequence LW-22-08

6.2.1. Evaluation of the proposed method

First, we will evaluate the KL1PS tracker with respect to the particle number n which can be set as ($n = 50, 100, 200$), and the results denoting the mean value of all frames are presented in Tables 1. In this table, the tracking error becomes smaller when the particle number is increasing. However, the improvement is not obvious when the particle number exceeds 100 which means that a satisfying result can be obtained without a large number of particles.

Table 1: Comparisons of average PosErr on different particle numbers

Sequence \ Number	50	100	200
LW-15-NS	5.39	2.75	2.11
LW-14-15	5.98	2.68	3.22
LW-16-08	4.42	2.66	3.04
LW-17-20	5.09	4.41	4.23
LW-18-16	5.71	4.77	4.54
LW-18-17	5.09	4.45	3.65
LW-18-18	4.97	4.09	3.48
LW-22-08	3.05	2.09	2.02

Then three search methods are compared and the results are shown in Tables 2. With the same particle number $N = 100$, the parallel search strategy has been proved to show the best performance. For the most similar particle, it is not reliable especially when the particle number is not large enough. When all the particles are employed to estimate the target state, it is inevitable to introduce error which will cause drift problem. The proposed method only use several efficient particles to estimate the target state which can improve the tracking accuracy.

6.2.2. Comparison with state-of-the-art trackers

Comparisons on the above eight FLIR sequences based on PosErr are shown in Fig.11. It shows the change of position error throughout the sequences for all the trackers. Besides, the average PosErr is listed in Table 3 which

Table 2: Comparisons of average PosErr on three search methods

Sequence \ Number	the most similar state particle	all the state particles	the efficient state particles
LW-15-NS	18.79	11.28	2.75
LW-14-15	8.25	4.66	2.68
LW-16-08	4.42	11.27	2.66
LW-17-20	6.02	4.23	4.41
LW-18-16	28.80	19.56	4.77
LW-18-17	15.42	20.39	4.45
LW-18-18	20.03	25.88	4.09
LW-22-08	2.25	3.05	2.09

310 is convenient for quantitative comparison. It can be seen that all the PosErr value of the KL1PS tracker is small
 311 which means that it locks the target well throughout each sequence. Totally speaking, it shows the best performance
 312 compared with other state-of-the-art trackers in tracking accuracy.

Table 3: Average PosErr comparisons of the above eight FLIR sequences

Sequence \ Methods	CSK	CT	DFT	Frag	IVT	L1	MTT	ORIA	KL1PS
LW-15-NS	25.24	2.36	48.38	28.12	39.74	1.38	1.08	1.03	2.75
LW-14-15	6.08	5.18	2.96	4.13	3.65	3.36	5.84	7.25	2.68
LW-16-08	3.13	3.40	3.16	2.45	7.89	3.45	8.46	8.66	2.66
LW-17-20	5.75	5.50	5.82	6.48	6.55	6.34	5.75	6.94	4.41
LW-18-16	1.36	6.17	49.55	26.60	23.25	7.56	9.66	9.36	4.77
LW-18-17	36.75	28.92	9.95	28.78	16.65	2.35	2.59	1.03	4.45
LW-18-18	4.41	5.21	4.32	36.03	25.61	4.94	4.63	12.79	4.09
LW-22-08	19.67	5.45	2.93	48.87	1.54	2.13	4.11	2.40	2.09

Figure 11: PosErr comparisons of the above eight FLIR sequences

The speed of all the trackers is measured in frames/s, and the comparative result is shown in Table 4. Although our method is slower than most of the others, it is much faster than MTT tracker. Based on the above experimental setup, the KL1PS tracker can perform the tracking task over 5 frames/s which is not very satisfying for real-time applications. The main reason is that it is time-consuming to compute two histogram features for plenty target candidates.

Table 4: Comparisons of speed by frames/s

Methods	CSK	CT	DFT	Frag	IVT	L1	MTT	ORIA	KL1PS
Time									
Speed	1282.12	31.17	13.42	12.36	42.25	7.87	0.30	27.90	5.31

7. Conclusion

In this paper, a parallel search strategy in the kernel feature space is proposed to track a target in the FLIR sequences. The kernel method is introduced to deal with target appearance variations and sparse representation is employed to selected efficient state particles. Due to only efficient state particles are employed to estimate the target state, a satisfying tracking accuracy can be obtained without a very large number of particles. Extensive experiment results have shown that the KL1PS tracker performs well in the challenging situations including pose change, illumination variations, occlusion, low SCR, etc. Though the speed is not very satisfying at present, it is desired to be improved based on the optimized implementation in the future work.

Acknowledgment

This work is supported by the National Science Foundation of China under Grant No. 61301207 and No.61371178. It is also supported by Program for New Century Excellent Talents in University under Grant No. NCET-13-0168.

References

- [1] A. Sanna, F. Lamberti, Advances in target detection and tracking in forward-looking infrared (flir) imagery, Sensors 14 (11) (2014) 20297–20303.

- [2] H. Yang, L. Shao, F. Zheng, L. Wang, Z. Song, Recent advances and trends in visual tracking: A review, *Neurocomputing* 74 (18) (2011) 3823–3831.
- [3] Y. Gao, R. Ji, L. Zhang, A. Hauptmann, Symbiotic tracker ensemble toward a unified tracking framework, *Circuits and Systems for Video Technology*, IEEE Transactions on 24 (7) (2014) 1122–1131.
- [4] X. Chen, Z. Qin, L. An, B. Bhanu, An online learned elementary grouping model for multi-target tracking, in: *Computer Vision and Pattern Recognition (CVPR)*, 2014 IEEE Conference on, IEEE, 2014, pp. 1242–1249.
- [5] V. Venkataraman, G. Fan, J. P. Havlicek, X. Fan, Y. Zhai, M. B. Yeary, Adaptive kalman filtering for histogram-based appearance learning in infrared imagery, *Image Processing*, IEEE Transactions on 21 (11) (2012) 4622–4635.
- [6] S. Baker, I. Matthews, Lucas-kanade 20 years on: A unifying framework, *International journal of computer vision* 56 (3) (2004) 221–255.
- [7] A. D. Jepson, D. J. Fleet, T. F. El-Maraghi, Robust online appearance models for visual tracking, *Pattern Analysis and Machine Intelligence*, IEEE Transactions on 25 (10) (2003) 1296–1311.
- [8] S. K. Zhou, R. Chellappa, B. Moghaddam, Visual tracking and recognition using appearance-adaptive models in particle filters, *Image Processing*, IEEE Transactions on 13 (11) (2004) 1491–1506.
- [9] M. J. Black, A. D. Jepson, Eigentracking: Robust matching and tracking of articulated objects using a view-based representation, *International Journal of Computer Vision* 26 (1) (1998) 63–84.
- [10] J. Ho, K.-C. Lee, M.-H. Yang, D. Kriegman, Visual tracking using learned linear subspaces, in: *Computer Vision and Pattern Recognition (CVPR)*, 2004 IEEE Conference on, Vol. 1, IEEE, 2004, pp. I–782.
- [11] D. A. Ross, J. Lim, R.-S. Lin, M.-H. Yang, Incremental learning for robust visual tracking, *International Journal of Computer Vision* 77 (1-3) (2008) 125–141.
- [12] J. Yan, X. Chen, D. Deng, Q. Zhu, Structured partial least squares based appearance model for visual tracking, *Neurocomputing* 144 (2014) 581–595.
- [13] G. Wu, W. Lu, G. Gao, C. Zhao, J. Liu, Regional deep learning model for visual tracking, *Neurocomputing* 175 (2016) 310–323.
- [14] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, *Pattern Analysis and Machine Intelligence*, IEEE Transactions on 25 (5) (2003) 564–577.
- [15] A. Yilmaz, K. Shafique, M. Shah, Target tracking in airborne forward looking infrared imagery, *Image and Vision Computing* 21 (7) (2003) 623–635.
- [16] R. T. Collins, Y. Liu, M. Leordeanu, Online selection of discriminative tracking features, *Pattern Analysis and Machine Intelligence*, IEEE Transactions on 27 (10) (2005) 1631–1643.
- [17] H. T. Nguyen, A. W. Smeulders, Robust tracking using foreground-background texture discrimination, *International Journal of Computer Vision* 69 (3) (2006) 277–293.
- [18] B. D. Lucas, T. Kanade, et al., An iterative image registration technique with an application to stereo vision., in: *IJCAI*, Vol. 81, 1981, pp. 674–679.
- [19] J. Fan, Y. Wu, S. Dai, Discriminative spatial attention for robust tracking, in: *Computer Vision–ECCV 2010*, Springer, 2010, pp. 480–493.

- [20] L. Sevilla-Lara, E. Learned-Miller, Distribution fields for tracking, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 1910–1917.
- [21] H. Grabner, M. Grabner, H. Bischof, Real-time tracking via on-line boosting., in: BMVC, Vol. 1, 2006, p. 6.
- [22] B. Babenko, M.-H. Yang, S. Belongie, Robust object tracking with online multiple instance learning, Pattern Analysis and Machine Intelligence, IEEE Transactions on 33 (8) (2011) 1619–1632.
- [23] S. Hare, A. Saffari, P. H. Torr, Struct: Structured output tracking with kernels, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 263–270.
- [24] M. S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking, Signal Processing, IEEE Transactions on 50 (2) (2002) 174–188.
- [25] Y. Wu, J. Lim, M.-H. Yang, Object tracking benchmark, Pattern Analysis and Machine Intelligence, IEEE Transactions on 37 (9) (2015) 1834–1848.
- [26] X. Mei, H. Ling, Robust visual tracking using ℓ_1 minimization, in: Computer Vision, 2009 IEEE 12th International Conference on, IEEE, 2009, pp. 1436–1443.
- [27] X. Jia, H. Lu, M.-H. Yang, Visual tracking via adaptive structural local sparse appearance model, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 1822–1829.
- [28] Q. Wang, F. Chen, W. Xu, M.-H. Yang, Online discriminative object tracking with local sparse representation, in: Applications of Computer Vision (WACV), 2012 IEEE Workshop on, IEEE, 2012, pp. 425–432.
- [29] X. Mei, H. Ling, Y. Wu, E. Blasch, L. Bai, Minimum error bounded efficient ℓ_1 tracker with occlusion detection, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 1257–1264.
- [30] C. Bao, Y. Wu, H. Ling, H. Ji, Real time robust ℓ_1 tracker using accelerated proximal gradient approach, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2012, pp. 1830–1837.
- [31] B. Liu, J. Huang, L. Yang, C. Kulikowsk, Robust tracking using local sparse appearance model and k-selection, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 1313–1320.
- [32] T. Zhang, B. Ghanem, S. Liu, N. Ahuja, Robust visual tracking via multi-task sparse learning, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 2042–2049.
- [33] W. Zhong, H. Lu, M.-H. Yang, Robust object tracking via sparse collaborative appearance model, Image Processing, IEEE Transactions on 23 (5) (2014) 2356–2368.
- [34] D. Wang, H. Lu, M.-H. Yang, Least soft-threshold squares tracking, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, 2013, pp. 2371–2378.
- [35] B. Schölkopf, S. Mika, C. J. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, A. J. Smola, Input space versus feature space in kernel-based methods, Neural Networks, IEEE Transactions on 10 (5) (1999) 1000–1017.
- [36] C. J. Burges, A tutorial on support vector machines for pattern recognition, Data mining and knowledge discovery 2 (2) (1998) 121–167.
- [37] L. Zhang, W. Zhou, L. Jiao, Support vector machines based on the orthogonal projection kernel of father wavelet, International Journal of

Computational Intelligence and Applications 5 (03) (2005) 283–303.

[38] L. Zhang, W. Zhou, On the sparseness of 1-norm support vector machines, *Neural Networks* 23 (3) (2010) 373–385.

[39] V. Roth, Sparse kernel regressors, in: *Artificial Neural Networks ICANN 2001*, Springer, 2001, pp. 339–346.

[40] S. Gao, I. W.-H. Tsang, L.-T. Chia, Kernel sparse representation for image classification and face recognition, in: *Computer Vision–ECCV 2010*, Springer, 2010, pp. 1–14.

[41] L. Zhang, W.-D. Zhou, P.-C. Chang, J. Liu, Z. Yan, T. Wang, F.-Z. Li, Kernel sparse representation-based classifier, *Signal Processing, IEEE Transactions on* 60 (4) (2012) 1684–1695.

[42] D. L. Donoho, For most large underdetermined systems of linear equations the minimal ℓ_1 -norm solution is also the sparsest solution, *Communications on Pure and Applied Mathematics* 59 (6) (2004) 797–829.

[43] J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online learning for matrix factorization and sparse coding, *The Journal of Machine Learning Research* 11 (2010) 19–60.

[44] J. F. Henriques, R. Caseiro, P. Martins, J. Batista, Exploiting the circulant structure of tracking-by-detection with kernels, in: *Computer Vision–ECCV 2012*, Springer, 2012, pp. 702–715.

[45] K. Zhang, L. Zhang, M.-H. Yang, Real-time compressive tracking, in: *Computer Vision–ECCV 2012*, Springer, 2012, pp. 864–877.

[46] A. Adam, E. Rivlin, I. Shimshoni, Robust fragments-based tracking using the integral histogram, in: *Computer Vision and Pattern Recognition (CVPR), 2006 IEEE Conference on*, Vol. 1, IEEE, 2006, pp. 798–805.

[47] N. Ahuja, Robust visual tracking via multi-task sparse learning, in: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, 2012, pp. 2042–2049.

[48] Y. Wu, B. Shen, H. Ling, Online robust image alignment via iterative convex optimization, in: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, 2012, pp. 1808–1814.