Robust Visual Tracking via Parallel Kernel Sparse Representation and NormalHedge

Jinjun Kuang
Intelligent Multimedia Technology Research Center
Room Hanguo Center Building NO.85, Jinyu Avenue
Chongqing, China
Email: kuangjinjun@cigit.ac.cn

Cheng Cheng
Intelligent Multimedia Technology Research Center
Room Hanguo Center Building NO.85, Jinyu Avenue
Chongqing, China
Email: chengcheng@cigit.ac.cn

***Abstract*—In this paper, a novel visual object tracking method based on NormalHedge (NH) and parallel kernel sparse representation (PKSR) is proposed to achieve robust tracking accuracy under challenging conditions such as the target and its background sharing similar patterns, occlusion and deformation. Kernel functions are capable of improving classification performance by casting features to high dimensional kernel space. However, standard coordinate descent-based sparse representation method is not efficient enough for tracking problems. Thus we propose a kernel parallel coordinate descent method (KPCD) to efficiently solve $_1$ minimization in the kernel space. The classification framework is proposed to calculate the loss value of each particle as well. Then, an adaptive dictionary updating method is proposed to create the over-completed dictionary. In addition, the states of the target are calculated by a recently developed online learning method similar to the particle filter called NormalHedge (NH), which has an effective re-sampling method to avoid the degeneracy problem. Extensive test results show the proposed method outperforms several state-of-art tracking methods in complex conditions.**

Keywords-Parallel Kernel Sparse Representation; NormalHedge; Visual Tracking (key words)

I. INTRODUCTION

Robust visual object tracking is a core area in computer vision research. It is relatively easy when the object is isolated and easily distinguished from the background. However, if the environment is crowed and complex, then the background and the object may share similar patterns, other objects may occlude the target, or even worse, deformations of the target may occur during the tracking process.

In order to solve those problems, a number of tracking methods were proposed in recent years. Among the recent advances made in this field, the sparse representation classification (SRC)-based visual tracking framework is perhaps the most significant one because its unique ability to handle illumination variations, occlusions, and random noises. Combining kalman-filtering technique, Han et al. [1] proposed a sample-based adaptive sparse representation visual tracking framework. Mei et al. [2] proposed a visual tracking and vehicle classification method as well. In their work, the tracking is considered as the binary classification problem, and their methods are similar to the SRC method proposed by Wright et al.[3].

In aforementioned frameworks, it is believed that the sparsity is the only key for classification. However, Rigamonti et al. [4] argued that solely enforcing the sparsity is not helpful in terms of classification accuracy. If there are high interclass similarities among dictionaries, or the target and its background sharing similar patterns, the classification rate could be greatly degraded. Collins et al. [10] did interesting researches on online selection of discriminative tracking features. Their work has profound impact for tracking problems, however, this idea cannot be directly incorporated into the SRC framework. Luckily, the kernel trick is capable of improving the classification performance as well, it maps the linear inseparable features into a higher dimensional feature space. Inspired by it, combining local binary pattern and coordinate descent (CD) method, an image classification method is proposed by Kang et al. [5]. Gao et al. [6] also proposed a similar image classification framework based on feature-sign search (FSS). Both methods have obtained promising results, however, the efficiency of the CD and the FSS-based KSR framework are not good enough for the real-time tracking. Thus, in this paper, we propose the KPCD to efficiently solve the $_1$ minimization problem, the sparse representation framework combining KPCD in this paper is called the PKSR.

Particle filtering technique has been widely used to estimate the hidden states of a target. The Particle filtering usually suffers degeneracy problem, which means most weights can concentrate on a small portion of particles, and this problem will greatly degrade the tracking performance. In this paper, instead of using a standard particle filter, a recently developed online learning algorithm NormalHedge (NH) [8] is used to estimate hidden states. The NH is similar to Particle filtering uses a bunch of particles to estimate the states of the target; besides, the NH has a very efficient re-sampling mechanism which will prevent the degeneracy problem from happening.

Furthermore, since the appearance of the target may vary greatly over the course of the tracking process, the high intra-class variation problem may degrade the tracking performance. In order to solve this problem, we consider the tracking as an incremental classification problem. The online adaptive dictionary updating scheme is proposed to online create an over-completed dictionary which contains both pure and sufficient target appearances.

II. Parallel Kernel sparse representation classification

Considering the tracking as a binary classification problem, the resolution of each patch is $w \times h$. The system dictionary $\mathbf{D}$ includes 2 classes, $\mathbf{D} = [\mathbf{D_t}, \mathbf{D_b}] \in R^{d \times z}$, the atom vector $\mathbf{v}$ is the basic element of the system dictionary, $\mathbf{D} = [\mathbf{v_{t,1}}, ..., \mathbf{v_{t,n}}, \mathbf{v_{b,1}}, ..., \mathbf{v_{b,q}}]$ and $\mathbf{v} \in \mathbf{R^d}$, $\mathbf{d} = \mathbf{w} \times \mathbf{h}$. The objective function of the kernel sparse representation framework can be formulated as follows.

$$\min_{\mathbf{C}} J(\mathbf{C}) = \frac{1}{2} \left\| \varphi(\mathbf{y}) - \sum_{i=1:z} \varphi(\mathbf{v_i}) \cdot c_i \right\|_2^2 + \lambda \|\mathbf{C}\|_1 \qquad (1)$$

where $\varphi(\cdot)$ maps a feature vector from low dimensional feature space to a high dimensional kernel space. The parameter $\lambda > 0$ is a scalar regularization parameter that balances the tradeoff between the sparsity and the reconstruction error. $\mathbf{D} \rightarrow \tilde{\mathbf{D}} = \left(\varphi(\mathbf{v}_1), \varphi(\mathbf{v}_2), ..., \varphi(\mathbf{v}_z)\right)$ and $K(\mathbf{x}, \mathbf{x}) = \varphi(\mathbf{x})^T \varphi(\mathbf{x}) = 1$.

In order to solve Eq. (1), in the following we develop a KPCD method which extends the Parallel Coordinate Descent [7] approach in the kernel space. The KPCD method is more efficient than many convex optimization methods. Given the number of parallel updates $P > 0$, the KPCD updates P coefficients simultaneously in each iteration.

To apply the KPCD to Eq. (2), first taking partial derivative of $J(C)$ with respect to c_j, when $c_j \neq 0$.

$$k(\mathbf{v}_j, \mathbf{v}_i) = \varphi(\mathbf{v}_j)^T \cdot \varphi(\mathbf{v}_i) \qquad (2)$$

$$k(\mathbf{v}_j, \mathbf{y}) = \varphi(\mathbf{v}_j)^T \cdot \varphi(\mathbf{y}) \qquad (3)$$

$$\nabla_c J(\mathbf{C}) = \sum_{i=1:z, i \neq j} c_i \cdot K(\mathbf{v}_j, \mathbf{v}_i) - K(\mathbf{v}_j, \mathbf{y}) + \lambda \text{sign}(c_j) \qquad (4)$$

Let Eq. (4) equals to zero, then choose the number of parallel updates $P > 0$. Similar to [7], if

$$\text{sign}(c_j) \cdot \left(c_j^k - \nabla_c J(\mathbf{C})\right) > 0 \qquad (5)$$

then, we get the update of c_j

$$c_j^{k+1} = c_j^k - \nabla_c J(\mathbf{C}) \qquad (6)$$

if

$$\text{sign}(c_j) \cdot \left(c_j^k - \nabla_c J(\mathbf{C})\right) \leq 0 \qquad (7)$$

then

$$c_j^{k+1} = 0 \qquad (8)$$

where $j \in [0, z]$ is a random integer and P coefficients $\mathbf{C}_p = \{c_1, c_2, ..., c_p\}$ have been randomly selected in each iteration. Given a proper kernel function, Eq. (5) ~ Eq. (8) will iteratively update $\mathbf{C}_p$ until convergence is achieved.

Note that the kernel function has to satisfy $K(\mathbf{x}, \mathbf{x}) = \varphi(\mathbf{x})^T \varphi(\mathbf{x}) = 1$, that means some kernels (e.g. polynomial or sigmoid kernel) are incompatible with KPCD framework. Furthermore, there are many kernel functions could fit into the proposed framework (i.e. Gaussian kernel, Laplace kernel and Rational quadratic kernel). After extensive experiments, we found the Laplace kernel is most suitable for the proposed method. The mathematical expression for the Laplace kernel is

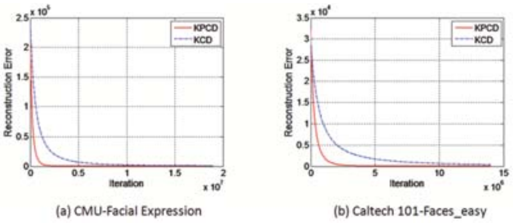

Figure 1. Comparisons of efficiency using two datasets. For the KPCD, the number of the parallel updates. The KCD updates one coefficient at a time. The Laplace kernel function is used for both methods. Experiment results suggest the KPCD provides near-linear speedups.

$$K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \exp\left(-\frac{\left|\mathbf{x}_i - \mathbf{x}_j\right|}{\sigma}\right) \qquad (9)$$

In order to calculate the loss l_i for particle i, we also develop the KPCD-SRC criterion as follows, define r_t as the reconstruction error of the target dictionary, then

$$\begin{aligned} r_t &= \left\| \sum_{i=1}^{n} \mathbf{D} \cdot \mathbf{C}_i - \phi(\mathbf{y}) \right\|_2^2 \\ &= \left(\mathbf{C}_t^T K(\mathbf{v}_i, \mathbf{v}_j)_{n \times n} \mathbf{C}_t - 2K(\mathbf{v}_i, \mathbf{y})_{n \times 1}^T \mathbf{C}_t \right) \end{aligned} \qquad (10)$$

where $\mathbf{C}_t$ is the coefficient corresponding to the target dictionary t. Besides, the loss $l_i = r_t$.

III. NormalHedge for Tracking

Similar to the well-known particle filtering technique, the NH uses a number of particles to sample potential areas and then estimate the hidden states of the target. Unlike the particle filtering, the NH has a very efficient re-sampling mechanism to avoid the particle degeneracy problem.

At first, N particles were initialized, $N \geq 2$. For particle i, a weight $\omega_{i,t}$ is assigned to it, and $\mathbf{W}_t$ is the weight vector for all particles. For particle i, there is a loss $l_{i,t}$, $l_{A,t}$ is the total loss of the particle set at iteration t.

$$l_{A,t} = \sum_{i=1}^{N} \omega_{i,t} l_{i,t} \qquad (11)$$

where $l_{i,t} = r_t$, r_t is a reconstruction error calculated by KPCD-SRC. Given the regret value γ of particle i, $\gamma_{i,t} = l_{A,t} - l_{i,t}$, then the accumulated regret value of particle i is,

$$R_{i,t} = r_{i,\tau} + \left(1 - \lambda\right) \cdot R_{i,t-1} \qquad (12)$$

To avoid the particle degeneracy problem, the particle has to be killed if $R_{i,t} < 0$. The half-normal distribution function for NH is defined as

$$\Phi(x,c) = \exp\left(\frac{\left(\max\left(0,x\right)\right)^2}{2c}\right) \tag{13}$$

where $x \in R,\ c > 0$. Given a measurement parameter $c_t > 0$, $R_{i,t}$ and c_t should satisfy the following equation

$$e = \frac{1}{N}\sum_{i=1}^{N}\exp\left(\frac{\left(\max(0,R_{i,t})\right)^2}{2c_t}\right) \tag{14}$$

The weight for particle i at time t is

$$\omega_{i,t} \propto \frac{[R_{i,t-1}]_+}{c_{t-1}}\exp\left(\frac{\left([R_{i,t-1}]_+\right)^2}{2c_{t-1}}\right) \tag{15}$$

Finally, the state could be calculated as

$$\mathbf{X}_{A,t} := \sum\nolimits_{i=1}^{N}\omega_{i,t}x_{i,t} \tag{16}$$

IV. Adaptive Dictionary Updating

In most cases, the target appearance is always changing during the tracking process. In order to adapt the dictionary to those changing target appearances and their backgrounds, it has to be online updated.

As is shown in Figure 2, the proposed online updating strategy could be explained by 3 feature space domains. The green circle is the known target area. The red circle is the known background area. The yellow circle is the potential target area. Note, although the kernel trick is capable of increasing the inter-class separability, in practice there are a small portion of features cannot be separated easily. Therefore, the target-related reconstruction residual r_t and the background-related reconstruction residual r_b might be greater than zero at the same time, and there is an overlapped area between the green area and the red area. The specific algorithm is introduced as follows.

(a) Given a minimal threshold $\xi_{\min}$, if $r_t < \xi_{\min}$, that means the test sample i is in the green area, and this sample can be represented by the known target dictionary.

(b) Given a maximal threshold $\xi_{\max}$, if $\xi_{\min} < r_t < \xi_{\max}$ and $r_t < r_b$, that means the test sample i is in the yellow area, and this sample should be added into the target dictionary $\mathbf{D}_t$ because it is similar to the existing target samples and could not be represented by background samples.

(c) If $\xi_{\min} < r_t < \xi_{\max}$ and $r_t > r_b$, that means the test sample i is in the red area, and this sample is a known background, it should not be added into the dictionary.

(d) If $r_t > \xi_{\max}$ and $r_b > \xi_{\max}$ that means it is neither the known target nor the known background, it should not be added into the dictionary.

In addition, the background dictionary should also be updated. At time step t, the tracking system will sample around the target and get some patches, then transform these patches into the background dictionary $\mathbf{D}_b$.

In this paper, $\xi_{\min} = 0.2$, $\xi_{\max} = 0.5$. And the time threshold is set to $\psi = 200s$.

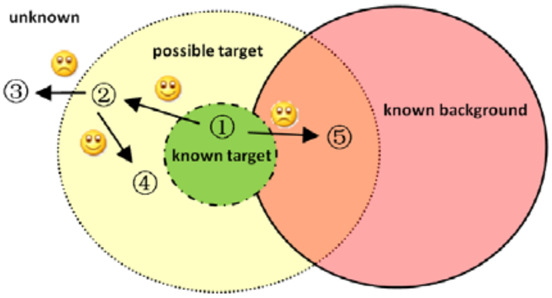

Figure 2. The dictionary online updating method.

V. Main Algorithm

The main algorithm can be summarized as follow:

Step 1: Acquire the target patch and background patches by manually in the first frame. Then use those patches to create $\mathbf{D}_t$ and $\mathbf{D}_b$.

Step 2: Calculate all losses by the PKSR method.

Step 3: Calculate the states of the target by the NH.

Step 4: Update the target dictionary and the background dictionary.

Step 5: Go to step 2.

VI. Experiments

A. Experimental Setup

The experiments are performed on a PC with a 2.0 GHz quad-core CPU and 4GM RAM. The test videos were recorded in different formats. The targets underwent challenging situations such as random noise, deformation and partial occlusions.

The proposed method is compared to the AdaSR [1] and the partial filter [9]. The resolutions of the videos were adjusted to 320 by 480; The size of target patch were adjusted to 16 by 16; The frame rate is 15 FPS.

B. Comparison Results

Figure 3 demonstrates some critical tracking results of three methods on four different test datasets. Figure 3 (a) shows results on the "PETS2006-S7-T6-B". This dataset is a regular indoor captured video; the target is a pedestrian walking from the right to the left. In this video, the target can be easily distinguished from the background and there are not many other disturbances during the tracking process. Thus all four trackers have successfully finished their tasks. As is shown in Table 1, the average distance of the proposed method from the tracked location to the ground truth is 5.3, which is slightly better than other two methods. Those images shown in Figure 3 (b) are test results from heavy noise conditions. The target in the test video “Karl-Wilhelm-StraBe: snow” is a blue vehicle which is blurred by the heavy fog. At initial frames, all three methods have tracked the target, however, the AdaSR and the particle filter failed to complete their tasks. According to Table 1, the AdaSR has better performance than the particle filter due to the SRC framework insensitive to noise. Figure 3 (c) shows results on the “visor: Indoor People Tracking with occlusions”. The target is a man who was occluded many times by a blue board and a dustbin. The particle filter lost its target when similar pattern appears; the AdaSR successfully followed the target until the target was almost fully occluded by the blue board. The proposed method successfully tracked the target throughout the process, however, due to the severe occlusion, sometime its bounding box seems a little loose. As is shown in Table 1, the tracking results of the proposed method are less accurate than experiments on other videos. Figure 3 (d) shows results on the “headtracker” dataset. The target is a lady's head. The target has been drastically deformed during the tracking process. The AdaSR and the particle filter failed to follow the target when deformation appears. The proposed method has successfully tracked the target due to its adaptive dictionary updating mechanism. The average tracking performances, the proposed method outperforms the particle filter and the AdaSR by 7.2 and 11.5 have been shown in Table 1, respectively.

TABLE 1. PERFORMANCE COMPARISON BETWEEN DIFFERENT METHODS WITH DIFFERENT COITIONS.

	Normal	Noise	Occlusion	Deformation
Proposed	5.3	7.8	16.1	12.3
PF	9.7	21.8	39.6	19.5
AdaSR	6.6	17.2	28.0	23.8

C. *Discussing*

The proposed method outperformed other state-of-art methods in challenging situations such as random noise, occlusion and deformation. For most cases, the proposed method is capable of tracking the target with high accuracy except that the target is fully occluded.

Another limitation of the proposed method is that if the initial template of the target is somehow polluted, the tracking accuracy will be degraded significantly.

VII. CONCLUSION

This paper presents a novel method based on NormalHedge (NH) and parallel kernel sparse representation (PKSR) for visual object tracking. The PKSR is proposed to efficiently solve the ℓ_1 minimization in kernel space and handling the high inter-class similarity problem; the adaptive dictionary updating strategy is proposed to handle the deformation and template drifting problem. Furthermore, the NH is used to estimate hidden states of the target. The widely known particle degeneracy problem is avoided by the unique re-sampling mechanism of the NH.

Comparison experiments were carried out with other state-of-art tracking methods to demonstrate the superiority of the proposed method. Experimental results show the proposed method is robust to random noise, partial occlusion and deformation.

The parallel processing mechanism is capable of linearly increasing the efficiency of ℓ_1 minimization process. The efficiency provided by it has opened a number of opportunities for other applications.

In future, the authors would like to continual the research on tracking systems, such as re-acquiring target after tracking failure occurred and tracking multiple targets simultaneously.

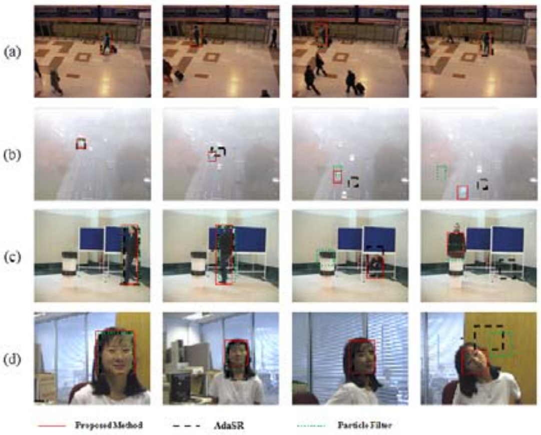

Figure 3. Tracking results in four different test videos.

ACKNOWLEDGMENT

Research was sponsored by committee on science and technology of Chongqing under grant No. cstc2011ggC40009 and cstc2012gg-sfgc0079, respectively. The authors would also like to thank the reviewers for their valuable comments and suggestions.

REFERENCES

[1] Z. HAN, J. JIAO, B. ZHANG, Q. YE, et al. “Visual object tracking via sample-based adaptive sparse representation (AdaSR),” Pattern Recognition, no. 44, pp. 2170-2183, 2011.

[2] X. Mei and H. Ling, “Robust visual tracking and vehicle classification via sparse representation,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, no. 99, pp. 2259-2272, 2011.

[3] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 31, no. 2, pp. 210-227, 2009.

[4] R. Rigamonti, M.A. Brown and V. Lepetit, “Are sparse representations really relevant for image classification? ” Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, no. 1545-1552, 2011.

[5] C. Kang, S. Liao, S. Xiang, “Kernel sparse representation with local patterns for face recognition,” Image Processing (ICIP), 2011 18th IEEE International Conference on, 2011: 3009-3012.

[6] S. Gao, I. Tsang, and L. Chia, “Kernel sparse representation for image classification and face recognition,” Computer Vision-ECCV 2010, pp. 1-14, 2010.

[7] J.K. Bradley, A. Kyrola, D. Bickson and C. Guestrin, “Parallel coordinate descent for ℓ1-regularized loss minimization,” Arxiv preprint arXiv:1105.5379, 2011.

[8] K. Chaudhuri, Y. Freund, and D. Hsu, “A parameter-free hedging algorithm,” Advances in Neural Information Processing Systems, pp. 297-305, 2009.

.

[9] F. Jiang, W. Guijin, L. Chang, L. Xinggang and W. Weiguo, “Robust Object Tracking via Combining Observation Models,” IEICE TRANSACTIONS on Information and Systems, pp. 662-665, 2010.

[10] R.T. Collins, Y. Liu, and M. Leordeanu, “Online selection of discriminative tracking features,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, pp. 1631-1643, 2005