

Visual tracking using IPCA and sparse representation

Dongjing Shan · Chao Zhang

Received: 9 August 2012 / Revised: 6 July 2013 / Accepted: 6 July 2013 / Published online: 31 July 2013
© Springer-Verlag London 2013

Abstract The main challenging issues in visual tracking can be listed as follows: significant variation of object's appearance or background image, illumination changes, serious or even complete occlusion of object, etc. In order to deal with them, two modules are needed: one is an accurate appearance model updating online and the other one is a robust matching method to find the target according to the learned model. In this paper, we propose a novel tracking method in a particle filter framework based on IPCA and sparse representation, in which IPCA is used to model the object appearance adaptively and sparse representation is used in two aspects: first, it helps to formulate a robust updating scheme of the IPCA; second, it strengthens the matching process significantly when the tracker copes with very challenging cases as mentioned in the beginning. In experiments, we select three state-of-the-art tracking methods for comparison and demonstrate the superiority of our method over them on various data.

Keywords Compressive sensing · ℓ^1 minimization · Sparse representation · Motion · Visual tracking · Particle filter

List of symbols

TI_i The i_{th} training images.
 \bar{TI} Mean value of $\{TI_1, TI_2, \dots, TI_n\}$.

A $d \times n$ data matrix ($[TI_1, TI_2, \dots, TI_n]$).
 B $d \times m$ data matrix ($[TI_{n+1}, \dots, TI_{n+m}]$).
 $U \Sigma V^T$ Singular value decomposition.
 t_f Fixed target template.
 t_d^0 Original dynamical target template.
 t_d Dynamical target template.
 f Forgetting factor.
 n Number of frames processed.
 m Number of frames newly come.
 \bar{V}_m Mean value of the targets in m frames.
 i_i One trivial template.
 $I = [i_1, i_2, \dots, i_l]$ Identity matrix.
 $T = [t_d, t_f]$ Target template set.
 y Tracking result.
 $a = [a_1, a_2]^T$ Target coefficient vector.
 $e = [e_1, \dots, e_l]^T$ Trivial coefficient vector.
 ε Error vector.
 e^+ Positive trivial coefficient vector.
 e^- Negative trivial coefficient vector.
 c Nonnegative coefficient vector.
 y_0 Tracking result in the previous frame.
 t Time in particle filter framework.
 X_t Parameter of affine transformation.
 μ Center of the subspace spanned by U .
 d_t Distance to the subspace.
 d_w Distance to the subspace center.

S. Dongjing (✉) · Z. Chao
Key Laboratory of Machine Perception (MOE), Peking University,
Beijing 100871, China
e-mail: shandongjing@pku.edu.cn

Z. Chao
e-mail: chzhang@cis.pku.edu.cn

1 Introduction

Visual tracking is one of the cardinal problems in computer vision. It has been recently used in many practical applications ranging from robot navigation, intelligent monitor-

ing, medical imaging, augmented reality to human–computer interaction. Nevertheless, the state-of-the-art methods are still far from achieving performance comparable to human ability. Trackers have to deal with several difficulties such as background clutter, serious or even complete occlusions, varying viewpoints and large pose changes [23]. In the past years, various kinds of methods were proposed to deal with them, and a rich literature in this field could be found in the internet.

Firstly, a lot of methods are proposed to formulate tracking in the probabilistic terms. Early works use Kalman filter [13,26] to provide solutions. As the limitation of linear Gaussian model, the particle filter [20,28] comes forth. Nowadays, it has been widely used as a tracking framework in many methods, for it can approximate an arbitrary observation model with a stochastically generated set of weighted samples. Another probabilistic model using mean-shift [6,25,27] prospers in the past years which simulates the distribution of probability density with random samples and searches the peak using gradient ascent algorithm. Secondly, in the offline tracking applications, the global trajectory optimization [4,12,18] is permitted to be applied for the reason that all the frames are available before the tracker starts. The global optimization is expected to result in smoother and more stable trajectories than some online methods. Thirdly, supervised discriminative methods using classification [3,7,21] have been exploited to solve tracking problems, which separate the object from the background by training discriminative classifiers and can achieve superior results. In [2], the proposed tracker uses multiple instance learning (MIL) instead of traditional supervised learning to avoid drifting problem and then leads to a more robust tracker. Furthermore, tracking by detection has been paid more attention in recent years. In the method of tracking-learning-detection (TLD) [15,16], the detection runs simultaneously with an optical tracker and can reinitialize it if it fails.

In this paper, we design a robust visual tracking method based on incremental PCA (IPCA) [9,10] and sparse representation [5,14,17], which uses probabilistic model (group one). Specifically, the method incrementally learns a low-dimensional subspace representation of the target appearance. And then, to find the target in one frame, the sparse representation of the target appearance in the previous frame is calculated on the target template (the mean image) and the trivial templates. Lastly, the resulted coefficients of the sparse representation can be used to lay a mask on the computation of the candidates' distances to the subspace in the particle filter framework, which helps to eliminate the effect of occlusion, noise, and so forth and then to select out the right candidate. Additionally, the coefficients can also be used to modify the updating scheme of the IPCA and make it more accurate to reflect the changes of the target appearance.

The rest of the paper is organized as follows. In the next section, the related work is reviewed. In Sect. 3, we detail

our tracking algorithm, and in Sect. 4, we show experimental results illustrating the superiority of our method compared with another three state-of-the-art trackers.

2 Related work

The most related works with this paper are [23] and [22]. In [23], the authors proposed a tracking method that incrementally learns a low-dimensional subspace representation, adapting online to changes in the appearance of the target. The update scheme of the sample mean and the eigenbasis makes the method robust when tracking the target objects that undergo large changes in pose, scale and illumination.

The work of [22] can be viewed as an extension of [24], in which sparse representation is employed for visual tracking with the intuition that the appearance of a tracked object can be sparsely represented by its appearances in previous frames. Specifically, to find the target in a new frame, each target candidate in the particle filter framework is sparsely represented in the space spanned by target templates and trivial templates. Then, the candidate which has the smallest projection error with its sparse representation is chosen as the tracked target. Mei and Ling [22] achieves good performance in cope with occlusion, noise and other challenging issues as declaimed in the paper.

The difference between [23] and our paper lies in that we design a new templates updating scheme based on sparse representation and modify the distance computing formula (observation model). This way makes the IPCA tracker more robust. Also when using sparse representation, we design a new set of target templates and take use of the coefficients obtained on the whole templates (target templates and trivial templates) to cope with challenging cases like occlusions and other turbulence.

3 Tracking algorithm

3.1 Incremental PCA

In [23], the authors extended the Sequential Karhunen–Loeve (SKL) algorithm [19] presenting a new incremental PCA algorithm that correctly updates the eigenbasis as well as the mean. Given a set of training images $\{TI_1, TI_2, \dots, TI_n\}$, one can obtain eigenvectors U by computing the singular value decomposition $U\Sigma V^T$ of the centered data matrix $[(TI_1 - \bar{TI}), \dots, (TI_n - \bar{TI})]$ where \bar{TI} symbolizes the mean of n images. And then, when m new images $\{TI_{n+1}, \dots, TI_{n+m}\}$ arrive, SKL computes U' and Σ' from the SVD of $[AB]$ ($A: d \times n$ data matrix which is composed of n observation vectors, B : new coming $d \times m$ data matrix). Based on SKL, [23] makes it slightly faster than that in the

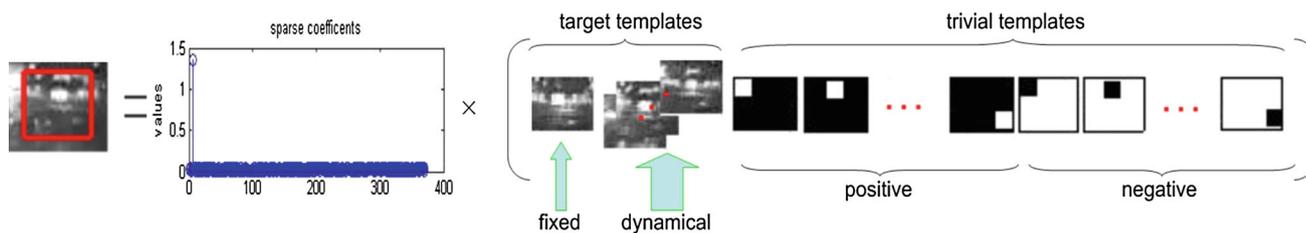


Fig. 1 A target image is sparsely represented by the target templates and trivial templates. The target templates are composed of a fixed target template and a dynamical target template, which is updated conditionally during the tracking process. Trivial templates are composed of the

positive and negative trivial templates, which are used to compensate the effect of occlusion and noise. (from the sequence “car” in Sect. 4.2, Fig. 5.)

computation process, and it avoids performing QR decomposition on the entire matrix $[U \Sigma B]$, instead only orthogonalizes $(B - UU'B)$. And also, Ross et al. [23] take into account the sample mean of the training data, which changes over time as new data come. The update scheme of the mean adopts a forgetting factor to ensure less modeling power with older observations.

In our paper, we use the algorithm briefly described above to construct the target appearance model, i.e., the subspace spanned by U and centered at the mean. And obviously, as detailed in Sect. 3.2.3, we plan to design a new scheme to select the data appropriate for updating the model as challenging cases like occlusion and noise occur.

3.2 Sparse representation of a tracking target

In [22], each candidate in the particle filter framework is sparsely represented in the space spanned by target templates and trivial templates. The target templates are composed of several previous target images (10 images in the paper [22]) and updated according to the weight scheme. We can see that the algorithm is time-consuming. So in our paper, we redesign the target templates and sparsely decompose only one target image, i.e., the target image in the previous frame or the last integral image. Then, we exploit the feature of the sparse coefficients in cope with occlusion, noise, etc, to control the update of the templates and finally utilize the coefficients to modify the observation model in the particle framework.

3.2.1 Target templates and trivial templates

As depicted in Fig. 1 (adapted from Fig. 1 of [24]), two target templates are used for sparse representation. One is called fixed target template (t_f), which is manually selected from the first frame and applied zero-mean-unit-norm normalization. The fixed target template could make the tracking more stable. The other one is called dynamical target template, which is just the sample mean of the incremental

IPCA and updated dynamically to capture the target changes as the tracker proceeds. When the tracker seizes the right and integrated target in a frame, it should be updated into the dynamical target template as follows:

$$t_d = \frac{fn}{fn+m} t_d^0 + \frac{m}{fn+m} \bar{V}_m, \tag{1}$$

where t_d^0 and t_d are the dynamical target templates before and after update, respectively. n denotes the number of frames having been processed, m denotes the number of frames newly come, which is set to 1 in our tracker and \bar{V}_m denotes the mean value of the target images in the m frames. f is a forgetting factor that reduces the effect of previously tracked target images and then keeps the template more consistent with the changing target. Specifically, using $n \leftarrow fn + m$, the effective number of observations will reach equilibrium at $n = fn + m$. So, when $f = 0.95$ and $m = 1$, new observation is included at each update, and the effective size of the observation history will approach $n = 20$.

One trivial template i_i in Fig. 1 is a vector with only one nonzero entry, i.e., $I = [i_1, i_2, \dots, i_l]$ is an identity matrix. Both the target templates and the trivial templates have the same fixed size (when updated into the dynamical template, the object images should be normalized to 12×15 first in our paper) and the tracked image is sparsely represented after being normalized to this size.

3.2.2 Formulation of sparse representation

Obviously, a tracking target lies approximately in the low-dimensional subspace spanned by the training images of the target or the historical images of the target extracted from the preceding frames. Given a target template set $T = [t_d, t_f] \in R^{l \times 2} (l \gg 2)$, containing 2 target templates which are reformed to one-dimensional vectors, a tracking result $y \in R^l$ can be represented approximately as a linear combination of the two target templates as follows:

$$y \approx Ta = a_1 t_d + a_2 t_f. \tag{2}$$

where $a = [a_1, a_2]^T \in R^2$ is called a target coefficient vector.

In many visual tracking scenarios, target objects are often corrupted by noise, partially occluded, or even completely disappear and reappear. Turbulence is brought into the linear representation of the tracking target y , to compensate the error caused by these incidents, Eq. (2) is rewritten as

$$y = Ta + \varepsilon, \tag{3}$$

where ε is an error vector used to compensate the difference of pixel value between the tracking target and the target template. Following the scheme in [24], we use the trivial templates $I = [i_1, i_2, \dots, i_l] \in R^{l \times l}$ to formulate the equation (3) as

$$y = [T, I] \begin{bmatrix} a \\ e \end{bmatrix}, \tag{4}$$

where a trivial template $i_i \in R^l$ is a vector with only one nonzero entry, l is arithmetic product of the size of template and $e = [e_1, e_2, \dots, e_l]^T \in R^l$ is called a trivial coefficient vector.

Since the templates that are most similar to the tracking target are positively related to the target, in the work of [22], the nonnegativity constraints are imposed on the coefficients to help filter out clutter that is similar to tracked targets in reversed intensity patterns. Consequently, model (4) is innovated as

$$y = [T, I, -I] \begin{bmatrix} a \\ e^+ \\ e^- \end{bmatrix} = Bc, \quad \text{s.t. } c \geq 0, \tag{5}$$

where $e^+ \in R^l$, $e^- \in R^l$ are called a positive trivial coefficient vector and a negative trivial coefficient vector, respectively, B is the matrix of template set and c is a nonnegative coefficient vector.

We solve the formulation (5) as an ℓ^1 -regularized least squares problem, which is known to typically yield sparse solutions [24].

$$\min \|Bc - y_0\|_2^2 + \lambda \|c\|_1, \quad \text{s.t. } c \geq 0, \tag{6}$$

where $\|\cdot\|_1$ and $\|\cdot\|_2$ denote the ℓ^1 and ℓ^2 norms, respectively, and y_0 symbolizes the target image in the previous frame or the last integral one (if the target disappears and reappears later). Our implementation solves the formula (6) via an interior-point method based on [17], which uses the preconditioned conjugate gradients (PCG) algorithm to compute the search direction. We record the resultant coefficient vector as c_0 .

3.2.3 Sparse coefficients

If we represent the tracking result in the previous frame as a linear combination of the template set composed of both tar-

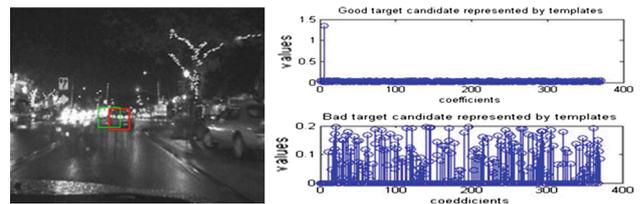


Fig. 2 Left: good (in red) and bad (in green) target candidates. Middle: good candidate represented by templates. Right: bad candidate represented by templates

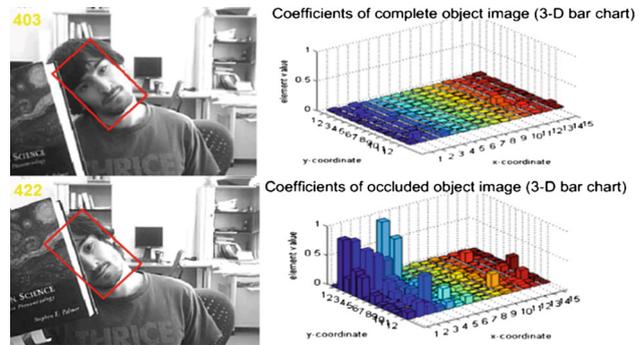


Fig. 3 Coefficients of two object images depicted in 3D space: the first row comprises an intact face image and its sparse coefficients, and the second row comprises an occluded face and its coefficients with some of them significant in the occluded area

get templates and trivial templates, obviously, a good result leads to a sparse coefficient vector by reason that it is similar to at least one of the target templates, and then, coefficients corresponding to trivial templates (named trivial coefficients) tend to be zeros. On the contrary, a bad result or a result seriously occluded by other object often leads to a dense representation since the trivial coefficients are almost all nonzero (see Fig. 2).

If a target image is noised or occluded, the coefficients of the trivial templates can compensate the effect of this turbulence. So, we can exploit the space information of them. In the middle column of Fig. 3, the nonzero coefficients indicate the partial face occluded or noised. The coefficient map just likes a mask that can be used to eliminate or alleviate the turbulence in tracking process. In Sect. 3.3, we illustrate how to use the mask.

Now, we will explain how to control the update of the IPCA model and the dynamical target templates (just the sample mean in IPCA). As mentioned in Sect. 3.1, it is easy to see that we just need to select the fitting images from new coming ones to update the model. Apparently, if the tracker drifts to a wrong object or background, or tracks the right but seriously or even completely occluded target, the images are not fitting for update. From Figs. 2 and 3, we can see that the mask of an occluded or wrongly tracked object image is dense while one of the complete images is sparse. So, the number of the elements in the mask which approach 1 (dense)

could be used as a kind of evaluation criterion. Only if the number is below a threshold ($0.5 \times$ size of the mask) could the object image be updated into the model.

Finally, we will illustrate in the following text that the dynamical target template used here is more effective and stable than the templates designed in [22].

$$p(I_t|X_t) = p_{d_t}(I_t|X_t)p_{w_t}(I_t|X_t) = N(I_t; \mu, UU^T + \varepsilon I)N(I_t; \mu, U\Sigma^{-2}U^T). \quad (7)$$

$$p(I_t|X_t) = \exp(-\| (I_t - \mu) * (i - c_0) - UU^T((I_t - \mu) * (i - c_0)) \|^2). \quad (8)$$

Obviously, our update scheme is simpler than [22] and can tolerate several wrong tracking results with sparse coefficients. In some very complex background, there exists probably an image block in background which is very similar to the target image, so it will be updated into the target template as its sparsity. But this will not bring fatal disaster to the tracker because the template integrates a lot of target images

in previous frames. In [22], a wrong template in the target template set highly probably causes the tracker locks onto itself, and this wrong template is hard to be removed from the set according to the proposed algorithm. The tracking results in Fig. 5 can demonstrate what we say.

3.3 Particle filter and its observation model

In particle filter, a number of particles are propagated to catch up with the moving object. In this work, the particle state at time t consists of the six parameters of an affine transformation $X_t = (x_t, y_t, \theta_t, s_t, \alpha_t, \phi_t)$ where $x_t, y_t, \theta_t, s_t, \alpha_t, \phi_t$, denote x, y translation, rotation angle, scale, aspect ratio and skew direction at time t . Dynamical model of X_t is that each parameter is modeled independently by a Gaussian distribution around its counterpart in X_{t-1} . Observation model is the key module which gives a similarity weight to each candidate, i.e., accomplishes the task of matching the candidates.

Table 1 The speed of four trackers in general condition (for indication only)

Tracker	Speed (s/frame)	Framework	Code
IPCA	0.08	Particle filter (300 particles used)	Matlab
ℓ^1 -tracker	4.5	Particle filter (300 particles used)	Matlab
MIL	1.2	Classifier	C
Our tracker	0.08	Particle filter (300 particles used)	Matlab

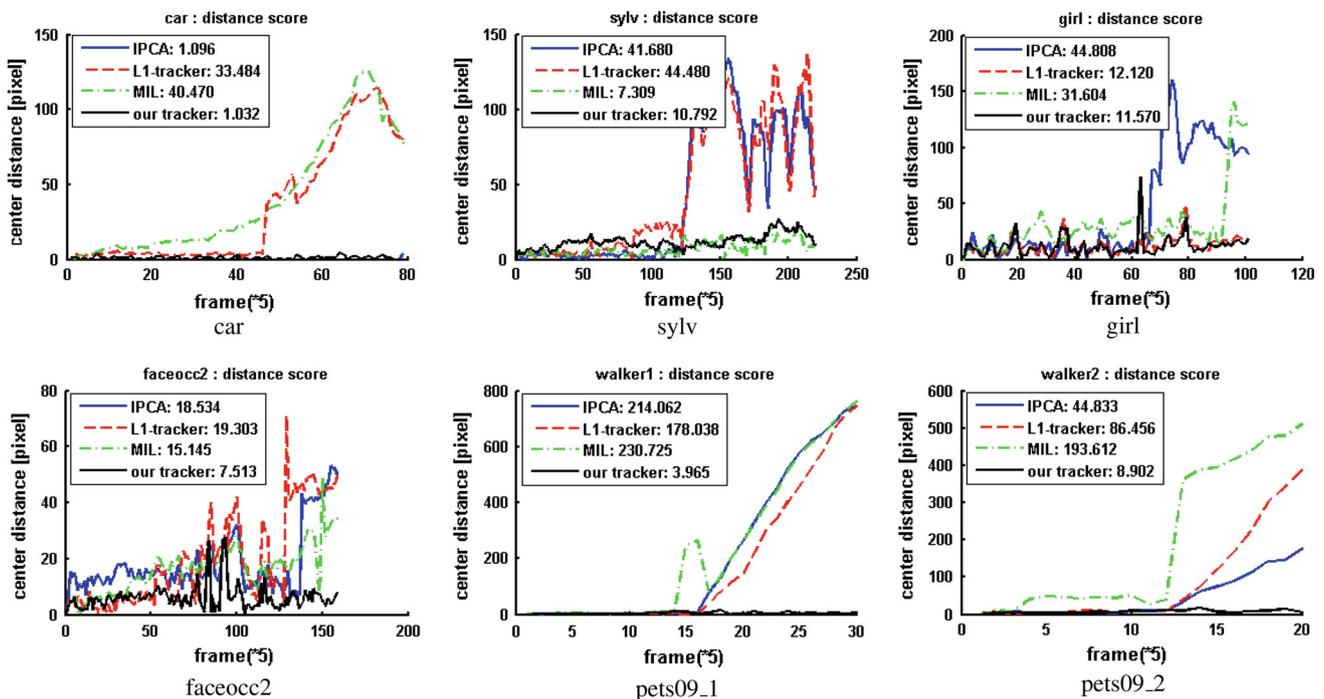


Fig. 4 the quantitative results (the center distance at intervals of five frames and the mean distance) of car [23], sylv [2], girl [2], faceocc2 [2], pets09_1 [8] and pets09_2 [8]

Observation model: in [23], the probability of a candidate being generated from the subspace spanned by U and centered at μ is inversely proportional to the distance d from the candidate to the reference point, i.e., μ of the subspace, which can be decomposed into the distance to the subspace, d_t , and the distance-within-subspace from the projected candidate to the subspace center, d_w . The equation is formulated in (7), where I_t is an image patch predicated by X_t . For simplicity, we only use d_t to measure p which is reasonable on the assumption as detailed in subsection 3.2.3 of [23]. Also, our work below is easy to be extended to incorporate d_w .

It can be shown [1] that the negative exponential distance from I_t to the subspace spanned by U , i.e., $\exp(-\|I_t - \mu - UU^T(I_t - \mu)\|^2)$, is proportional to $p(I_t|X_t)$ as $\varepsilon \rightarrow 0$. So, the coefficient vector c_0 (mask) we obtain in Sect. 3.2.2 can be used to modify this observation model as formulated in Eqs. (7) and (8).

As detailed before, the mask is computed for the tracked target image in the previous frame. Then, it can be used to strengthen the matching process in the current frame as the appearance change of the target between neighboring frames

is not too large. The mask acts as a weight map that indicates the occluded or noised fractions with regard to each candidate. The formulation is displayed in (8), in which $(i - c_0)$ helps eliminate the effect of turbulence, i and \cdot^* symbolize a vector full of 1 and dot product, respectively.

The experiments demonstrate the significant improvement of our tracker in cope with various challenging cases such as serious occlusion and noise. Also, we provide the code in the supplementary material, and it can be run to verify the performance. Thanks a lot for the authors of [23] and [22] for we take use of their code to write ours.

4 Experiments

We compare our algorithm with three state-of-the-art tracking methods on several well-chosen sequences. The methods for comparison include IPCA tracker [23], ℓ^1 -minimization tracker [22] and multiple instance learning (MIL) tracker [2]. All the experiments are done with a computer of 2.8 GHz P4 CPU and 2GB memory. The speeds of the trackers are listed

Table 2 The mean distance and the percentage of frames correctly tracked

Bold values indicate first ranking and underlined values indicate second ranking

Sequence	IPCA	ℓ^1 -tracker	MIL	Our tracker
car [23]	<u>1.096</u> / 1.0	33.484 / 0.60	40.474 / 0.41	1.032 / 1.0
sylv [2]	41.680 / 0.55	44.480 / 0.41	7.309 / 0.78	<u>10.792</u> / <u>0.76</u>
girl [2]	44.808 / 0.66	<u>12.120</u> / 0.97	31.604 / 0.71	11.570 / <u>0.96</u>
faceocc2 [2]	18.534 / 0.80	19.303 / 0.69	<u>15.145</u> / <u>0.96</u>	7.513 / 0.97
pets09_1 [8]	214.062 / 0.67	<u>178.038</u> / 0.67	230.725 / 0.46	3.965 / 0.92
pets09_2 [8]	<u>44.833</u> / 0.71	86.456 / 0.71	193.612 / 0.65	8.902 / 0.91

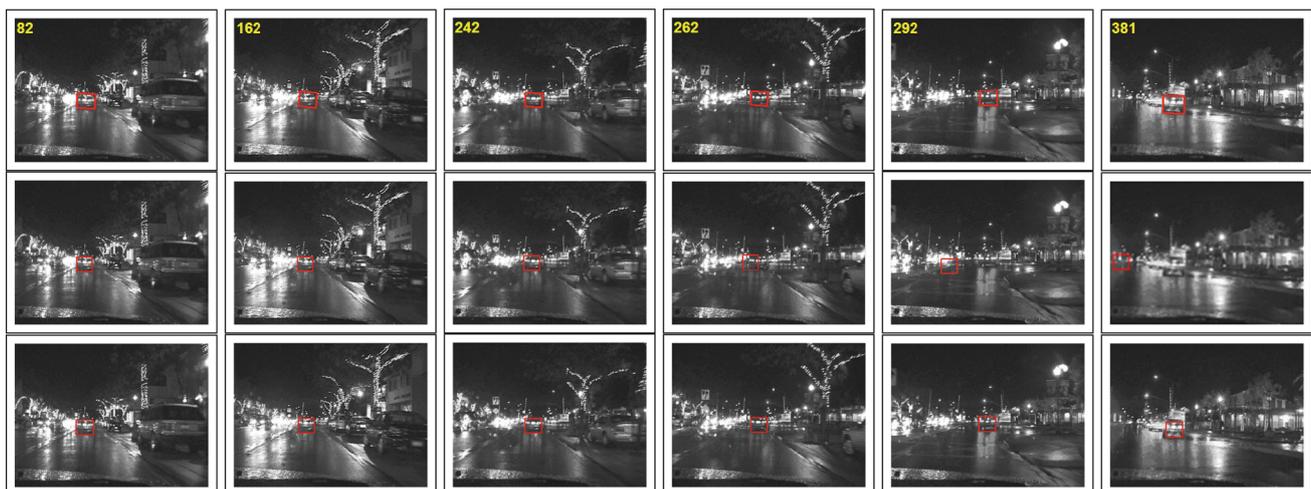


Fig. 5 Picture results of the “car” in Fig. 4. The *first* row is the result of IPCA, the *second* row is the result of ℓ^1 -tracker and the *third* row is our tracker. We can see that ℓ^1 -tracker (the *second* row) loses the target

because wrong templates is updated into its template set, whereas IPCA tracker and our tracker achieve success

in Table 1. Our tracker is almost the same fast with IPCA because the time consumed by sparse representation of just one target is very trivial. In the following experiments, we adopt two evaluation methods to compare the performance of four trackers, in which one is the center distance of the tracking rectangle to the ground truth, and the other is the percentage of frames tracked correctly. Also, we display some results of sequences with pictures for intuitive feelings in the end.

4.1 Evaluation methods

Two evaluation methods mentioned before are detailed below. One is the center distance of the tracking rectangle

to the ground truth, which is computed every five frames (as the ground truth is labeled at intervals of five frames). Also, we compute a mean value of the distances to show the performance of each tracker. The other is the percentage of frames tracked correctly. In order to evaluate performance, we use the PASCAL challenge [11] object detection score. Given the detected bounding box ROI_D and the ground truth bounding box ROI_{GT} , the overlap score is evaluated as

$$\text{score} = \frac{\text{area}(ROI_D \cap ROI_{GT})}{\text{area}(ROI_D \cup ROI_{GT})}. \tag{9}$$

We can interpret the frame as true positive if the score exceeds 0.5.



Fig. 6 Picture results of the “faceocc2” in Fig. 4. The *first row* is the result of IPCA, the *second row* is the result of ℓ^1 -tracker and the *third row* is our tracker which behaves the best. As use of sparse representation,

our tracker can eliminate the effect of turbulence caused by occluded area and find the accurate object



Fig. 7 Picture results of the “pets09_1” in Fig. 4. A women is walking in the opposite direction with the stream of people and encounters serious occlusion sometimes. Only our tracker (the *third row*) achieves

success while IPCA tracker (the *first row*) and ℓ^1 -tracker(the *second row*) fail to track the target

4.2 Performance of four trackers

We select seven sequences involving different kinds of challenging situations in which the former six ones are publicly available, and the last one is a moving car video taken on an urban road. In Fig. 4, we depict the quantitative results of the former six sequences. Table 2 indicates that our tracker achieves the best performance, and it can cope with serious occlusions caused by moving objects.

Nextly, we depict some results of the sequences with pictures and add a video shot in the urban road where a car is occluded seriously by another one. In Figs. 5, 6, 7, 8

and 9, the three rows are results of IPCA, ℓ^1 -tracker and our tracker, respectively. For MIL-tracker, we do not get the picture results but only the coordinates of tracking trajectories.

Finally, we will illustrate the results of Fig. 5 to demonstrate what we mentioned in the ending of Sect. 3.2.3, i.e., our updating scheme of the templates works better than that of ℓ^1 -tracker. We can see that ℓ^1 -tracker loses the object because some wrong templates come into the template set; however, our tracker succeeds because it can tolerate some inaccurate templates (with sparse coefficients) and rejects definitely wrong templates (with dense coefficients) and consequently prevents drifting.

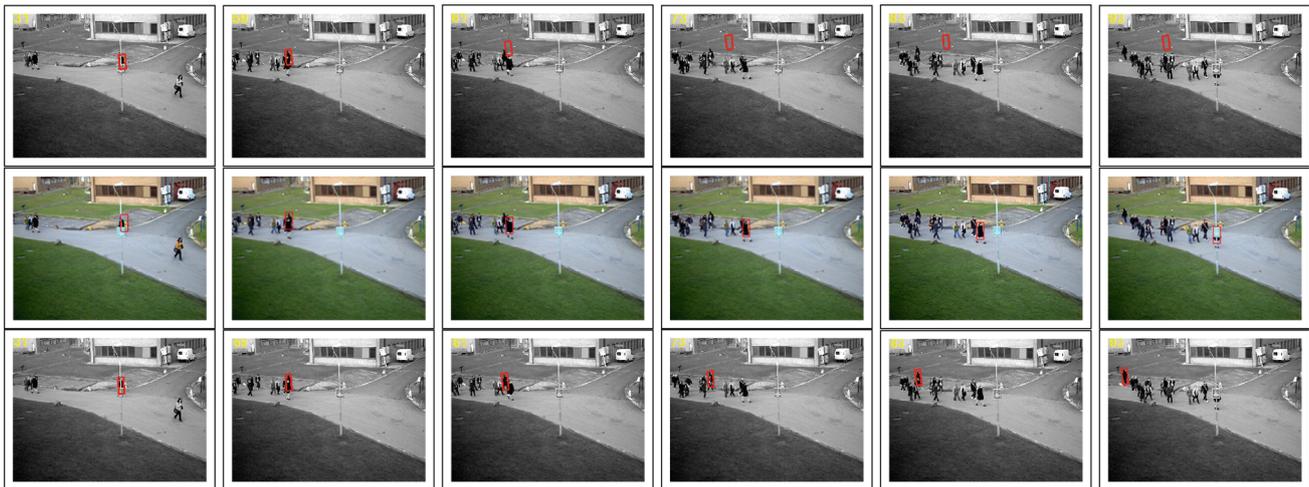


Fig. 8 Picture results of the “pets09_2” in Fig. 4. The *first row* is the result of IPCA, the *second row* is the result of ℓ^1 -tracker and the *third row* is our tracker. In the scene, a man is walking across people flow with serious occlusions and appearance changes. Only our tracker (the *third*

row) achieves success. IPCA could find the man while serious occlusion occurs and ℓ^1 -tracker locks to a wrong object (its discrimination ability is weaker than IPCA in this case)

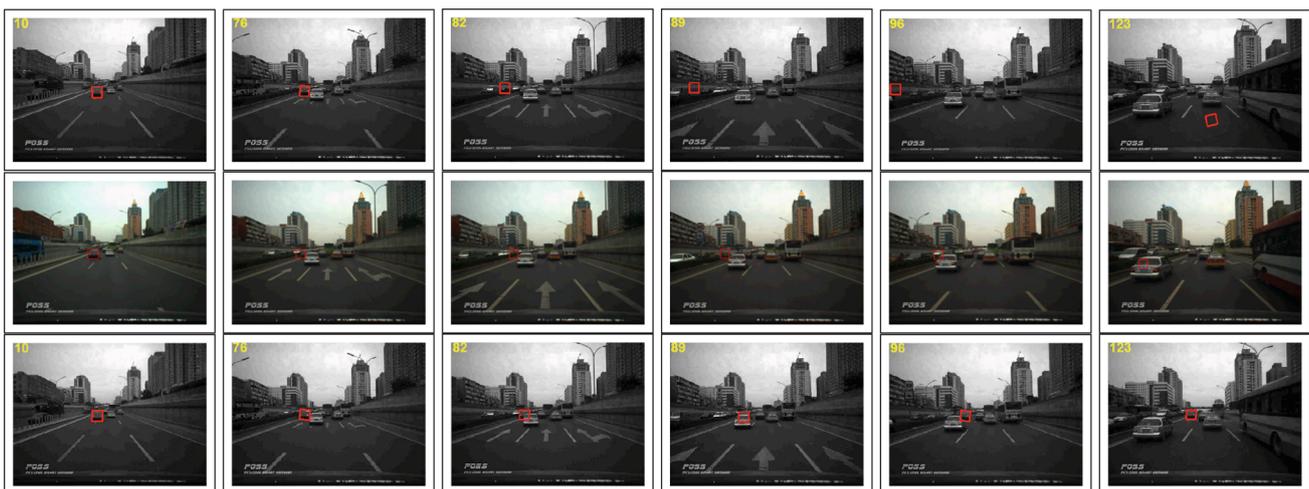


Fig. 9 Picture results of the video taken on an urban road in which the black car tracked is occluded seriously by a white car and then reappears. The *first row* is the result of IPCA, the *second row* is the result of

ℓ^1 -tracker and the *third row* is our tracker. Our tracker (the *third row*) could track the target car and refind it while the other two fail

5 Conclusions and the future work

In this paper, we propose a tracking method based on IPCA and sparse representation. The mask derived from sparse representation is used to eliminate or alleviate the effect of noise, occlusion, etc, during tracking and help update the IPCA appearance model so as to make it more fitting to capture the appearance change. Our tracker works stably when coping with challenging situations as cluttered background, serious or even complete occlusion and large appearance change. In the future, we can combine a detector with the tracker. So, it will be more robust, especially for dealing with the case that the object disappears in one place and reappears in another.

Acknowledgments This work was supported by National Key Basic Research Project of China (973Program) 2011CB302400 and National Nature Science Foundation of China (NSFC Grant No. 61071156).

References

- Ahn, J.H., Choi, S., Oh, J.H.: A new way of PCA: integrated-squared-error and EM algorithms. In: Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on, vol. 5, pp. V-777–V-780 (2004). doi:[10.1109/ICASSP.2004.1327226](https://doi.org/10.1109/ICASSP.2004.1327226)
- Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: CVPR 2009. IEEE Conference on, pp. 983–990 (2009). doi:[10.1109/CVPR.2009.5206737](https://doi.org/10.1109/CVPR.2009.5206737)
- Babenko, B., Yang, M.H., Belongie, S.: Robust object tracking with online multiple instance learning. *Pattern Anal. Mach. Intell. IEEE Trans.* **33**(8), 1619–1632 (2011). doi:[10.1109/TPAMI.2010.226](https://doi.org/10.1109/TPAMI.2010.226)
- Berclaz, J., Fleuret, F., Fua, P.: Robust people tracking with global trajectory optimization. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 1, pp. 744–750 (2006). doi:[10.1109/CVPR.2006.258](https://doi.org/10.1109/CVPR.2006.258)
- Candes, E.J., Romberg, J.K., Tao, T.: Stable signal recovery from incomplete and inaccurate measurements. *Commun. Pure Appl. Math.* **59**(8), 1207–1223 (2006)
- Collins, R.: Mean-shift blob tracking through scale space. In: Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, vol. 2, pp. II-234–II-240 (2003). doi:[10.1109/CVPR.2003.1211475](https://doi.org/10.1109/CVPR.2003.1211475)
- Collins, R., Liu, Y., Leordeanu, M.: Online selection of discriminative tracking features. *Pattern Anal. Mach. Intell. IEEE Trans.* **27**(10), 1631–1643 (2005). doi:[10.1109/TPAMI.2005.205](https://doi.org/10.1109/TPAMI.2005.205)
- Ferryman, J., Crowley, J.L.: Crowd-pets09. <http://www.cvg.cs.rdg.ac.uk/PETS2001/> (2009)
- Dagher, I., Nachar, R.: Face recognition using IPCA-ICA algorithm. *Pattern Anal. Mach. Intell. IEEE Trans.* **28**(6), 996–1000 (2006). doi:[10.1109/TPAMI.2006.118](https://doi.org/10.1109/TPAMI.2006.118)
- Decker, D., Punch, W., Sticklen, J.: IPCA, an architecture for intelligent control. In: Intelligent Control, 1996, Proceedings of the 1996 IEEE International Symposium on, pp. 80–85 (1996). doi:[10.1109/ISIC.1996.556181](https://doi.org/10.1109/ISIC.1996.556181)
- Everingham, M., Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The Pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **88**(2), 303–338 (2010). doi:[10.1007/s11263-009-0275-4](https://doi.org/10.1007/s11263-009-0275-4)
- Gu, S., Zheng, Y., Tomasi, C.: Linear time offline tracking and lower envelope algorithms. In: Computer Vision (ICCV), 2011 IEEE International Conference on, pp. 1840–1846 (2011). doi:[10.1109/ICCV.2011.6126451](https://doi.org/10.1109/ICCV.2011.6126451)
- Hargrave, P.: A tutorial introduction to Kalman filtering. In: Kalman Filters: Introduction, Applications and Future Developments, IEE Colloquium on, pp. 1/1–1/6 (1989)
- Herman, M., Strohmer, T.: General deviants: an analysis of perturbations in compressed sensing. *Sel. Top. Signal Process. IEEE J.* **4**(2), 342–349 (2010). doi:[10.1109/JSTSP.2009.2039170](https://doi.org/10.1109/JSTSP.2009.2039170)
- Kalal, Z., Matas, J., Mikolajczyk, K.: Online learning of robust object detectors during unstable tracking. In: Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on, pp. 1417–1424 (2009). doi:[10.1109/ICCVW.2009.5457446](https://doi.org/10.1109/ICCVW.2009.5457446)
- Kalal, Z., Matas, J., Mikolajczyk, K.: P-n learning: Bootstrapping binary classifiers by structural constraints. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 49–56 (2010). doi:[10.1109/CVPR.2010.5540231](https://doi.org/10.1109/CVPR.2010.5540231)
- Kim, S.J., Koh, K., Lustig, M., Boyd, S., Gorinevsky, D.: An interior-point method for large-scale ℓ_1 -regularized least squares. *Sel. Top. Signal Process. IEEE J.* **1**(4), 606–617 (2007). doi:[10.1109/JSTSP.2007.910971](https://doi.org/10.1109/JSTSP.2007.910971)
- Leibe, B., Schindler, K., Van Gool, L.: Coupled detection and trajectory estimation for multi-object tracking. In: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, pp. 1–8 (2007). doi:[10.1109/ICCV.2007.4408936](https://doi.org/10.1109/ICCV.2007.4408936)
- Levy, A., Lindenbaum, M.: Sequential karhunen-loeve basis extraction and its application to images. In: Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on, vol. 2, pp. 456–460 (1998). doi:[10.1109/ICIP.1998.723422](https://doi.org/10.1109/ICIP.1998.723422)
- Liu, B., Ji, C., Zhang, Y., Hao, C., Wong, K.K.: Multi-target tracking in clutter with sequential monte carlo methods. *IET Radar, Sonar Navig.* **4**(5), 662–672 (2010). doi:[10.1049/iet-rsn.2009.0051](https://doi.org/10.1049/iet-rsn.2009.0051)
- Liu, L., Fieguth, P.: Texture classification from random features. *Pattern Anal. Mach. Intell. IEEE Trans.* **34**(3), 574–586 (2012). doi:[10.1109/TPAMI.2011.145](https://doi.org/10.1109/TPAMI.2011.145)
- Mei, X., Ling, H.: Robust visual tracking and vehicle classification via sparse representation. *Pattern Anal. Mach. Intell. IEEE Trans.* **33**(11), 2259–2272 (2011). doi:[10.1109/TPAMI.2011.66](https://doi.org/10.1109/TPAMI.2011.66)
- Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. *Int. J. Comput. Vis.* **77**(1–3), 125–141 (2008). doi:[10.1007/s11263-007-0075-7](https://doi.org/10.1007/s11263-007-0075-7)
- Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y.: Robust face recognition via sparse representation. *Pattern Anal. Mach. Intell., IEEE Trans.* **31**(2), 210–227 (2009). doi:[10.1109/TPAMI.2008.79](https://doi.org/10.1109/TPAMI.2008.79)
- Yang, C., Duraiswami, R., Davis, L.: Efficient mean-shift tracking via a new similarity measure. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1, pp. 176–183 (2005). doi:[10.1109/CVPR.2005.139](https://doi.org/10.1109/CVPR.2005.139)
- Zhang, D., Van Gool, L., Oosterlinck, A.: Generalized predictive control of a vision-based tracking system using kalman filtering technique. In: Control and Applications, 1989. Proceedings. ICCON '89. IEEE International Conference on, pp. 731–733 (1989). doi:[10.1109/ICCON.1989.770616](https://doi.org/10.1109/ICCON.1989.770616)
- Zhou, H., Yuan, Y., Shi, C.: Object tracking using sift features and mean shift. *Comput. Vis. Image Underst.* **113**(3), 345–352 (2009). doi:[10.1016/j.cviu.2008.08.006](https://doi.org/10.1016/j.cviu.2008.08.006)
- Zhou, S.K., Chellappa, R., Moghaddam, B.: Visual tracking and recognition using appearance-adaptive models in particle filters. *Image Process. IEEE Trans.* **13**(11), 1491–1506 (2004). doi:[10.1109/TIP.2004.836152](https://doi.org/10.1109/TIP.2004.836152)