

RESEARCH

Open Access



Scale adaptive compressive tracking

Pengpeng Zhao*, Shaohui Cui, Min Gao and Dan Fang

*Correspondence:
zhaopengpeng5258@sina.com
Electronic Engineering
Department, Shijiazhuang
Mechanical Engineering College,
No. 97 Heping West Road,
Shijiazhuang 050003, China

Abstract

Recently, the compressive tracking (CT) method (Zhang et al. in Proceedings of European conference on computer vision, pp 864–877, 2012) has attracted much attention due to its high efficiency, but it cannot well deal with the scale changing objects due to its constant tracking box. To address this issue, in this paper we propose a scale adaptive CT approach, which adaptively adjusts the scale of tracking box with the size variation of the objects. Our method significantly improves CT in three aspects: Firstly, the scale of tracking box is adaptively adjusted according to the size of the objects. Secondly, in the CT method, all the compressive features are supposed independent and equal contribution to the classifier. Actually, different compressive features have different confidence coefficients. In our proposed method, the confidence coefficients of features are computed and used to achieve different contribution to the classifier. Finally, in the CT method, the learning parameter λ is constant, which will result in large tracking drift on the occasion of object occlusion or large scale appearance variation. In our proposed method, a variable learning parameter λ is adopted, which can be adjusted according to the object appearance variation rate. Extensive experiments on the CVPR2013 tracking benchmark demonstrate the superior performance of the proposed method compared to state-of-the-art tracking algorithms.

Keywords: Visual tracking, Compressive tracking, Feature template, Model update

Background

Visual object tracking is a significant research hotspot in computer vision because of its numerous applications such as intelligent monitor system, precision guide system, intelligent medical diagnosis, etc. However, it remains a challenging task to develop robust tracking algorithms due to the appearance change caused by illumination, motion, occlusion, and so on. Aimed at this issue, numerous algorithms have been proposed, which can be divided into algorithms based on generative appearance models (Babu et al. 2007; Lim et al. 2004; Adam et al. 2006; Mei and Ling 2009, 2011; Li et al. 2011; Liu et al. 2013) and algorithms based on discriminative appearance models (Avidan 2004; Grabner and Bischof 2006; Grabner et al. 2008; Babenko et al. 2011; Zhang et al. 2013, 2014a, b; Liu et al. 2015).

Generative algorithms typically learn a representative object model, which is utilized to search for the most similar region in image according to a certain similarity principle. Babu et al. (2007) make use of a linear subspace model to represent object appearance for tracking. Lim et al. (2004) utilizes an incremental learning method to update both subspace model and samples average. Adam et al. (2006) use the intensity histograms of

multiple fragments to represent object appearance, which can be computed by integral images efficiently. Recently, Mei and Ling (2009, 2011) proposed a robust object tracking method based on the sparse representation theory, named l_1 tracker, which introduces the sparse representation theory into object tracking at the first time. Li et al. (2011) further improved the l_1 tracker by using the orthogonal matching pursuit algorithm for solving the optimization problems efficiently. Liu et al. (2013) propose robust visual tracking method using local sparse appearance model and k-selection, which introduces block cording coefficient into mean shift to search for the optimal tracking result. Despite much progress has been achieved, there are still several problems to be solve in generative tracking algorithms. First, numerous training samples are required for learning an object appearance model at the start. However, if the object appearance change significantly during this period, the drift problem is likely to occur. Secondly, all of these generative algorithms don't use the background information, which is likely helpful to improve the tracking results.

Discriminative algorithms regard object tracking as a binary classification task, the goal of which is to find the optimal classify function between different classes. Avidan (2004) makes use of an off-line support vector machine (SVM) classifier to design a tracker. Grabner and Bischof (2006) propose an on-line features selected visual tracking method by using Adboost algorithm to select features on line. Soon afterwards, Grabner et al. (2008) propose semi-supervised on-line boosting for robust tracking, and the key of the method combines the advantage of both on-line and off-line classifier. Babenko et al. (2011) treat the tracking task as a multiple instance learning (MIL) problem, and propose a robust object tracking method with online MIL. Zhang et al. (2013) point out the shortcoming of on-line MIL, and propose a new tracking method, named ODFS, by introducing features selection into on-line MIL system. Recently, Zhang et al. (2012) propose a real-time compressive tracking (CT) algorithm that employs a very sparse random matrix to achieve a low-dimensional object appearance representation. Soon afterwards, Zhang et al. (2014a, b) further improve CT algorithm by reducing computational complexity. Liu et al. (2015) point out the shortness of CT algorithm, and propose adaptive compressive tracking method via online vector boosting feature selection.

In this paper, we propose a scale adaptive CT method which can adaptively adjusts the scale of tracking box with the size variation of the objects. Furthermore, the confidence coefficients of features are computed and used to achieve different contribution to the classifier. Finally, a variable learning parameter λ is adopted, which can be adjusted according to the object appearance variation rate. Extensive experiments on the CVPR2013 tracking benchmark demonstrate the superior performance of the proposed method compared to state-of-the-art tracking algorithms in terms of efficiency, accuracy and robustness.

Compressive tracking

The idea of CT is motivated by the compressive sensing theory, in which the random projections of a high dimensional signal can keep the original information to a great extent (Candes and Tao 2005, 2006). The main components of CT are shown by Fig. 1. At the t -th frame, both positive samples and negative samples are represented by high-dimensional multi-scale vectors via convolving each patch with some rectangle filters.

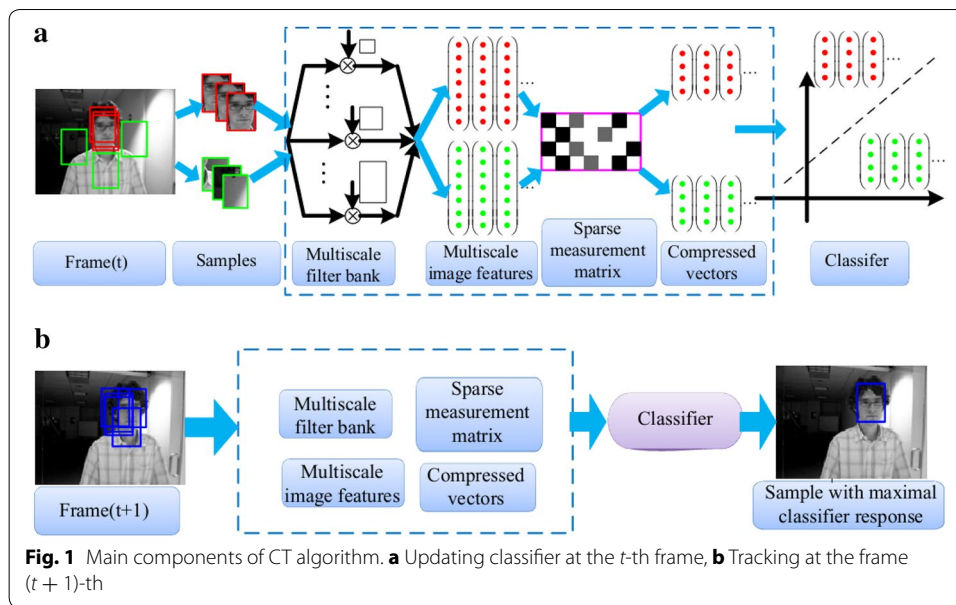


Fig. 1 Main components of CT algorithm. **a** Updating classifier at the t -th frame, **b** Tracking at the frame $(t + 1)$ -th

Then, each vector is projected onto a low-dimensional space by employing a very sparse random projection matrix that satisfies the restricted isometry property (RIP). And then, the compressed vectors are utilized to train the classifier. At the $(t + 1)$ -th frame, each candidate sample is similarly processed, and then the trained classifier is utilized to search for sample with maximal classifier response. In order to analyze, we divide the CT algorithm into several steps as follows.

Step 1: Sample two sets of image patches $D^\alpha = \{z | \|I(z) - I_t\| < \alpha\}$ and $D^{\zeta, \beta} = \{z | \zeta < \|I(z) - I_t\| < \beta\}$ with $\alpha < \zeta < \beta$, where I_t is the tracking location at the t -th frame, D^α and $D^{\zeta, \beta}$ represent the positive and negative samples respectively.

Step 2: Each patch is transformed into a high-dimensional multi-scale vector \mathbf{x} via convolving each patch with some rectangle filters at multiple scales $\{h_{1,1}, \dots, h_{w,h}\}$ defined as

$$h_{i,j}(x, y) = \begin{cases} 1, & 0 < x \leq i, 0 < y \leq j \\ 0, & \text{otherwise} \end{cases}, \quad (1)$$

where i and j are the width and height of a rectangle filter, respectively. Each vector $\mathbf{x} \in R^m$ represents the multi-scale features of each patch.

Step 3: A random matrix $\mathbf{R} \in R^{n \times m}$ is employed to project the high-dimensional vector \mathbf{x} onto a low-dimensional vector $\mathbf{v} \in R^n$ as $\mathbf{v} = \mathbf{R}\mathbf{x}$, where the entry of \mathbf{R} is represented by

$$r_{ij} = \sqrt{\rho} \times \begin{cases} 1 & \text{with probability } 1/2\rho \\ 0 & \text{with probability } 1 - 1/\rho \\ -1 & \text{with probability } 1/2\rho \end{cases}, \quad (2)$$

where $\rho = 2$ or 3 , and Achlioptas (2003) has proved that this type of matrix in such a case satisfies the Johnson-Lindenstrauss lemma. Thus, each low-dimensional vector $\mathbf{v} = \{v_i\}$, $i \in [1, n]$ represents the compressive features of each sampled patch, and it can be efficiently computed using the integral image method.

Step 4: Each compressive feature's conditional distributions in positive samples and negative samples both are assumed to be Gaussian distributed as $p(v_i|y=1.) \sim N(\mu_i^1, \sigma_i^1)$, $p(v_i|y=0.) \sim N(\mu_i^0, \sigma_i^0)$. And the low-dimensional vector $\mathbf{v} = \{v_i, i \in [1, n]\}$ is utilized to update Gaussian parameters

$$\begin{aligned}\mu_i^1 &= \lambda\mu_i^1 + (1-\lambda)\mu^1 \\ \sigma_i^1 &= \sqrt{\lambda(\sigma_i^1)^2 + (1-\lambda)(\sigma^1)^2 + \lambda(1-\lambda)(\mu_i^1 - \mu^1)^2},\end{aligned}\quad (3)$$

where $\sigma^1 = \sqrt{1/n \sum_{k=0}^{n-1} (v_i(k) - \mu^1)^2}$, $\mu^1 = 1/n \sum_{k=0}^{n-1} v_i(k)$, λ is a constant learning parameter, its value depends on particular situations. When the object appearance change significantly, λ takes smaller value; otherwise, λ takes bigger value.

Step 5: Sample a set of image patches in the $(t+1)$ -th frame, $D^\gamma = \{z | \|I(z) - I_t\| < \gamma\}$, where I_t is the tracking location at the t -th frame, and extract the features with low dimensionality. In this step, the sliding window method is used to traverse the whole candidate region to sample the patches, the sizes of which are all same to the object at t -th frame, illustrated in Fig. 2. It is worth mentioning that the search radius γ and one pixel distance in the figure are enlarged for show.

Step 6: A native Bayes classifier is utilized to distinguish the classes of each patch,

$$H(\mathbf{v}) = \log \left(\frac{\prod_{i=1}^n p(v_i|y=1)p(y=1)}{\prod_{i=1}^n p(v_i|y=0)p(y=0)} \right) = \sum_{i=1}^n \log \left(\frac{p(v_i|y=1)}{p(v_i|y=0)} \right). \quad (4)$$

In this step, all compressive features v_i in the vector \mathbf{v} are assumed independent and equal contribution to the classifier (Zhang et al. 2012; Ng and Jordan 2002). By using the classifier $H(\mathbf{v})$, we find the tracking location I_{t+1} with the maximal classifier response.

Although CT algorithm is demonstrated efficient by several experiments in Zhang et al. (2012), it has some limitations that makes CT perform unfavorably in some cases: First, in the classical CT algorithm, the estimation of scale changes of the target is not solved. Second, its constant learning parameter λ and uniform weights of Haar features are likely to bring drift when the object appearance changes significantly. In the following section, we will propose a scale adaptive CT that can deal with these issues well.

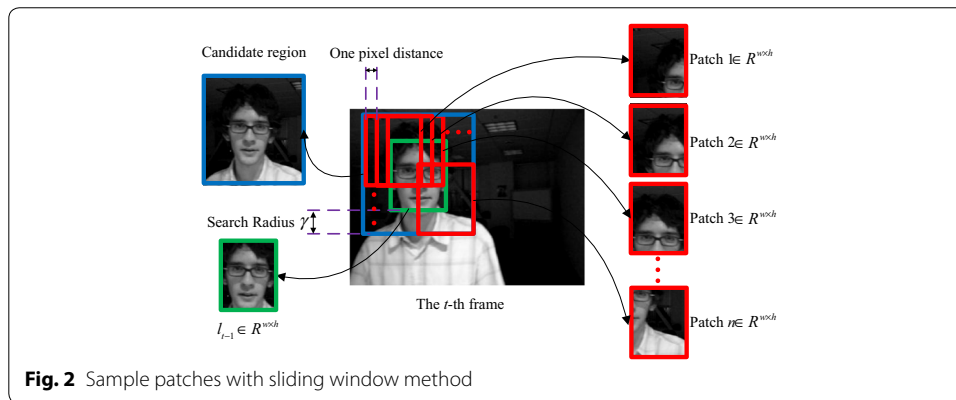


Fig. 2 Sample patches with sliding window method

Scale adaptive compressive tracking

Algorithm overview

Algorithm 1. Scale Adaptive Compressive Tracking (SACT)

Input: The $(t+1)$ -th image frame

1. Sample a set of image patches $D = \{\mathbf{z} | p(l_{t+1}(\mathbf{z}) | l_t) \sim N(0, Q)\}$ with different location and sizes via (4), and extract the compressive features via (5).
2. Use the new classifier $H(\mathbf{v})$ in (9) to each feature vector $\mathbf{v}(\mathbf{z})$, and find the tracking location \mathbf{l}_{t+1} with the maximal classifier response.
3. Sample two sets of image patches $D^\alpha = \{\mathbf{z} | \|\mathbf{l}(z) - \mathbf{l}_{t+1}\| < \alpha\}$ and $D^{\zeta, \beta} = \{\mathbf{z} | \zeta < \|\mathbf{l}(z) - \mathbf{l}_{t+1}\| < \beta\}$ with $\alpha < \zeta < \beta$, and extract the features with low dimensionality.
4. Use the low-dimensional vectors of positive and negative samples update Gaussian parameters and via (2), where the learning parameter λ is computed via (10).

Output: Tracking location \mathbf{l}_{t+1} , and classifier parameters

The proposed adaptive compressive tracking is summarized in Algorithm 1, which improved CT algorithm mainly in three aspects. Firstly, both the location and size of the object are regarded as variable parameters, which make up a vector $s_t = (x_p, y_p, w_p, h_p)$. The vector is assumed to be Gaussian distributed under the assumption of Brownian motion model $p(s_t | s_{t-1}) \sim N(0, Q)$, where $Q = \text{diag}(\sigma_s^2)$ is the covariance matrix containing diagonal elements, each corresponding to the variance of individual parameter x_t, y_t, w_t and h_t . In this way, a series of patches with different size and location are sampled instead of all the patches in CT being in the same size. Secondly, the weights of the Haar features are defined by computing each feature's ability of discriminating the object from background. These different weights are used in the classifier model instead of all the Haar features in CT having the same weight. Finally, a novel performance metric is applied to distinguish whether the current frame is reliable and low possibility of occlusion from the background or intersection from other objects. Only when the metric is satisfied, are the parameters $(\mu_i^1, \sigma_i^1, \mu_i^0, \sigma_i^0)$ incrementally updated. But the parameters in CT are updated at every frame instead.

Multi scale patches sampling and their features extraction

As illustrated by Fig. 2, the patches in CT are sampled by using sliding window method to traverse the whole candidate region. In this way, the sizes of all the patches are all same to the object at t -th frame. However, if the size of the object changes significantly in tracking, the drift problem is likely to occur. To handle this problem, a multi scale patches sampling method is proposed in this section, and the integral image method is still utilized to compute the compressive features efficiently.

In our algorithm, both the location and size of the object are regarded as variable parameters, which make up a vector $l_t = (x_p, y_p, w_p, h_p)$, where x_t and y_t represent the center coordinates of the object at t -th frame, w_t and h_t are the width and height of the object at t -th frame. The vector l_t is assumed to be Gaussian distributed under the assumption of Brownian motion model

$$p(l_{t+1} | l_t) \sim N(0, Q), \quad (5)$$

where $Q = \text{diag}(\sigma_l^2)$ is the covariance matrix containing diagonal elements, each corresponding to the variance of individual parameter x_t, y_t, w_t and h_t . In this way, a series of patches with different size and location are sampled instead of all the patches in CT being in the same size.

As shown in Fig. 3, in CT algorithm, the t -th compressive feature v_i in the compressed vector \mathbf{v} is constructed by several feature templates, whose sizes and locations are set randomly and fixed during tracking. While in our proposed method, the sizes and locations of the feature templates cannot be fixed during tracking, because the sizes of sampled patches are various. The parameters of the feature templates are computed as

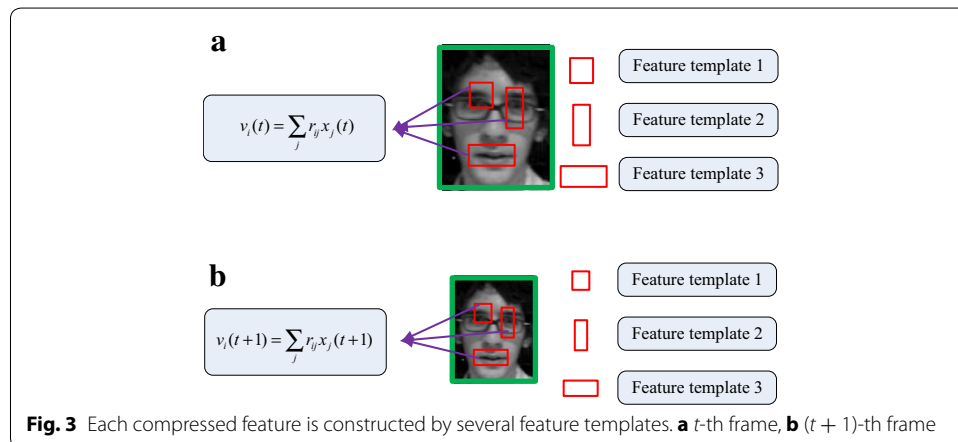
$$\begin{aligned} bx_{t+1}^{(n)} &= \frac{bx_t}{w_t} \cdot w_{t+1}^{(n)}, by_{t+1}^{(n)} = \frac{by_t}{h_t} \cdot h_{t+1}^{(n)} \\ bw_{t+1}^{(n)} &= \frac{bw_t}{w_t} \cdot w_{t+1}^{(n)}, bh_{t+1}^{(n)} = \frac{bh_t}{h_t} \cdot h_{t+1}^{(n)} \end{aligned} \quad (6)$$

where $(bx_{t+1}^{(n)}, by_{t+1}^{(n)}, bw_{t+1}^{(n)}, bh_{t+1}^{(n)})$ represent the locations and sizes of future templates in the n -th sampled patches at $(t+1)$ -th frame, (bx_t, by_t, bw_t, bh_t) represent the locations and sizes of future templates at t -th frame, $w_{t+1}^{(n)}$ and $h_{t+1}^{(n)}$ are the width and height of the n -th sampled patches at $(t+1)$ -th frame, w_t and h_t are the width and height of the object at t -th frame. The integral image method is still utilized to compute each rectangular feature efficiently.

Evaluation and application of features' confidence

In CT, all the compressive features are supposed independent and equal contribution to the classifier (Zhang et al. 2012; Ng and Jordan 2002). Actually, different compressive features have different confidence coefficients. In our proposed algorithm, the confidence coefficients of features are computed and used to achieve different contribution to the classifier.

As the references (Abraham et al. 2013; Jing et al. 2011; Zhang et al. 2014a, b) referred to, the confidence of a feature can be represented by computing the feature's ability of discriminating the object from background, which is computed though using the Hellinger distance between a feature's distributions of positive and negative samples in our method



$$h^2(p(v_i|y=0), p(v_i|y=1)) = \frac{1}{2} \int (\sqrt{f_1(x)} - \sqrt{f_0(x)})^2 dx, \quad (7)$$

where $f_1(x)$ and $f_0(x)$ are the feature's probability density functions (PDF) of positive samples and negative samples. Similar to CT, the distributions are assumed to be Gaussian distributed as

$$p(v_i|y=1) \sim N(\mu_i^1, \sigma_i^1), p(v_i|y=0) \sim N(\mu_i^0, \sigma_i^0), \quad (8)$$

Substituting (7) into (6), we can get

$$h^2 = 1 - \sqrt{\frac{2\sigma_1\sigma_0}{\sigma_1^2 + \sigma_0^2}} \exp\left(-\frac{1}{4} \frac{(\mu_1 - \mu_0)^2}{\sigma_1^2 + \sigma_0^2}\right). \quad (9)$$

It is obvious that h satisfies $0 \leq h \leq 1$, and the bigger value h takes, the stronger ability of discriminating the object from background. And afterwards, the Hellinger distance h is utilized in the classifier to achieve the goal that features with stronger ability make more contribution to the classifier

$$H(v) = \sum_{i=1}^n h_i \log\left(\frac{p(v_i|y=1)}{p(v_i|y=0)}\right). \quad (10)$$

Online learning of features' conditional distribution

After the tracking location has been found in a new frame, its positive and negative samples are used to update the Gaussian distribution parameters with introducing a learning parameter λ in CT, as (2) illustrated. However, CT suffers drift when the object appearance changes much due to its fixed learning rate λ . In our proposed method, a variable learning parameter λ is adopted, which can be adjusted according to the object appearance variation rate. To achieve this, $\rho = \sum_u \sqrt{q_u^t p_u^{t+1}}$ is utilized to compute the Bhattacharyya coefficient between the object being tracked and the last object at last frame (q_u^t implies the histogram of the object at the t -th frame, p_u^{t+1} implies the histogram of the object at the $(t+1)$ -th frame). It is obvious that ρ satisfies $0 \leq \rho \leq 1$. And a larger ρ means the object appearance changes rapidly, consequently the Gaussian distribution parameters need a larger learning rate. On the contrary, a smaller learning rate is needed. However, when $\rho < \Theta$, which means the current location of the object is not accurate or the occlusion has occur, the Gaussian distribution model stop update. In conclusion, the new learning parameter can be represented as

$$\begin{cases} 1, & \rho < \Theta \\ \lambda'/\rho = \frac{\lambda'}{\sum_u \sqrt{q_u^t p_u^{t+1}}}, & \rho \geq \Theta \end{cases}, \quad (11)$$

where λ' is the given constant learning parameter, ρ is the Bhattacharyya coefficient, λ is our new learning parameter, which can be adaptively adjusted according to the object appearance variation rate. Then λ in Eq. (3) will be instead by our new learning parameter, which is defined by (11).

Experiments

We evaluate the proposed algorithm with 7 state-of-the-art methods on 50 challenging sequences, which are all among the CVPR2013 tracking benchmark (Wu et al. 2013). The 7 contrastive trackers are summarized in literature (Wu et al. 2013), containing the CSK method, the VTS method, the SCM tracker, the VTD tracker, the TLD tracker, the Struck method, and the CT method. The reason of choosing these 7 trackers is that all of them except CT has been demonstrated much better performance than other trackers, like OAB, Frag, DFT, for example. We also choose CT method to verify if the proposed tracker can improve it greatly. For fair comparison, we use the source or binary codes provided by the authors with tuned parameters for best performance. For our compared trackers, we either use the tuned parameters from the source codes or empirically set them for best results.

Setup

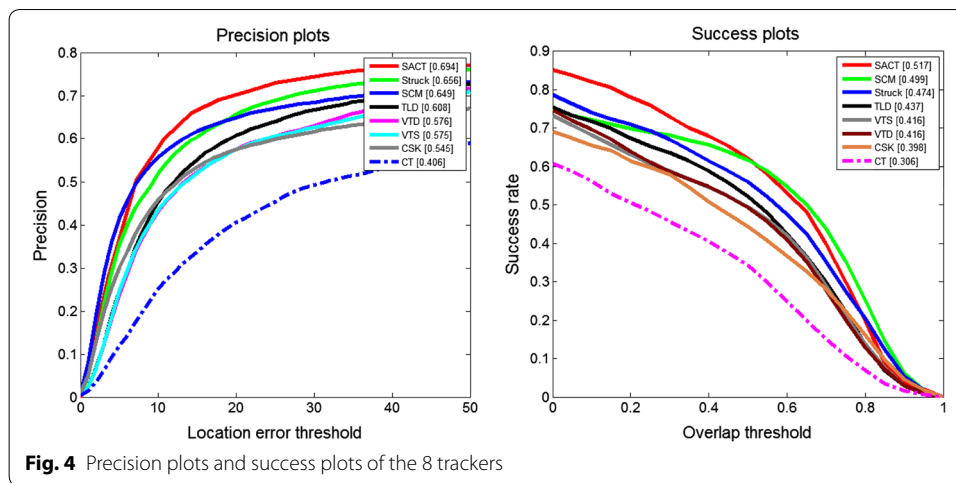
The search radius of sampling positive samples is set to $\alpha = 3$, where 50 positive samples are extracted. The inner and outer radii of sampling positive samples are set to $\zeta = 6$ and $\beta = 25$, where 40 negative samples are extracted randomly. The dimensionality of projected space is set to $n = 50$, and the given constant learning parameter λ' is set to 0.8, and the threshold value is set to $\Theta = 0.5$. The empirically determined parameters $\sigma_x, \sigma_y, \sigma_w, \sigma_h$ in Q are empirically chosen depending on the motion and attributes of the target in different videos. Table 1 lists the parameter values of some sequences in our experiments.

Experimental Results

We use the precision plot and success plot defined in Wu et al. (2013) to evaluate the proposed algorithm with 7 state-of-the-art trackers. The precision plot shows the percentage of frames whose estimated average center location errors are within the given threshold distance to the ground truth. The score at the threshold 20 pixels is defined as the precision score. The success plot shows the percentage of frames whose overlap score are more than a threshold value, where the overlap score is defined as $SCORE = \frac{area(ROT_t \cap ROT_a)}{area(ROT_t \cup ROT_a)}$ with the tracking bounding box ROT_t and the ground truth bounding box ROT_a . The threshold value ranges from 0 to 1, and the area under curve is used to measure the success score. Figure 4 shows the overall performance of the 7 evaluated tracking algorithms and the proposed algorithm SACT in terms of precision plot and success plot. Table 2 lists the precision score and success score for the 7 state-of-the-art trackers and SACT.

Table 1 Parameter Values used in the tests

Video	$(\sigma_x, \sigma_y, \sigma_w, \sigma_h)$	Video	$(\sigma_x, \sigma_y, \sigma_w, \sigma_h)$
Dudek	(5, 5, 0.3, 0.3)	Football	(3, 3, 0.1, 0.1)
Car scale	(3, 3, 0.5, 0.5)	Faceocc	(5, 5, 0.1, 0.1)
Fish	(1, 1, 0.05, 0.05)	Basketball	(8, 8, 0.05, 0.05)
Car dark	(1, 1, 0.01, 0.01)	Soccer	(3, 3, 0.2, 0.2)

**Table 2** Precision score and success score of the 8 trackers

	SACT	Stuck	SCM	TLD	VTD	VTS	CSK	CT
Precision plots	0.694	0.656	0.649	0.608	0.576	0.575	0.545	0.406
Success plots	0.517	0.474	0.499	0.437	0.416	0.416	0.398	0.306

The proposed SACT achieves the best tracking results in terms of both precision score and success score: the precision score of SACT is 0.694, which outperforms the STRUCK algorithm (ranking 2nd) by 5.79 %; meanwhile, the success score of SACT is 0.517, which outperforms the SCM algorithm (0.499 ranking 2nd). We note that the simple Haar-like features is employed to represent the object and background in the proposed algorithm SACT and the simple naive Bayesian classifier with low computational complexity is adopted in SACT. Thus, the proposed algorithm SACT outperforms STRUCK and SCM that resort to complicate learning techniques in terms of both accuracy and efficiency. Besides, one can be seen from Table 2 that the proposed SACT improves CT to a large extent: the precision score of SACT outperforms 0.406 (the precision score of CT) by 70.9 %; meanwhile, the success score of SACT outperforms 0.306 (the precision score of CT) by 68.9 %. Figure 5 shows screenshots of some tracking results.

Scale and pose change

For the *Dudek* sequence shown in Fig. 5a, the scale and pose of the object both change gradually. The tracking results of the front half images indicate that all of these algorithms have a certain ability of dealing with pose variation (e.g., #125). But we also observe that CT, CSK and Stuck cannot deal with scale variation well due to the error caused by their constant tracking box. On contrary, other methods concluding the proposed SACT can adjust their tracking boxes according to the scale of the object (e.g., #565). Furthermore, it is obviously that the tracking box of SACT is tighter and more accurate than TLD, SCM, VTS and VTD, especially when the object gets smaller in size (e.g., #1080). The proposed SACT can deal with scale and pose variation due to the



Fig. 5 Screenshots of some sampled tracking results. **a** Dudek, **b** Car scale, **c** Fish, **d** Car dark, **e** Football, **f** Faceocc, **g** Basketball, **h** Soccer

Gaussian distributed tracking box and random features selection that has been proved to handle pose variation well.

For the *Car scale* sequence shown in Fig. 5b, the object suffers from great scale change. Challenges also come from the interference caused by the tree when the object goes through it. We observed that CT, CSK and TLD drift when the object goes through the tree in the video. VTS, VTD and Struck only track a certain part of the object as it gets larger in size. On contrary, SCM and our SACT can track the object accurately in the whole sequence.

Illumination change

For the *Fish* sequence shown in Fig. 5c, the object undergoes several times of illumination change. The tracking result indicates that illumination getting stronger will have little effect on the tracking results of each algorithm (e.g., #156). But all the algorithms except SACT drift once the illumination get weaker (e.g., #160 and #437). The proposed SACT can deal with illumination Change in terms of its adaptive local appearance model, that is to say, different compressive features have different confidence coefficients in our tracker. For the *Car dark* sequence shown in Fig. 5d, the object undergoes large changes in environmental illumination with the car running along the street. CT, VTS, VTD and TLD drift gradually (320–388) as illumination changing while SCM, STRUCK, VTS and the proposed SACT achieve much better performance.

Background clutters or occlusion

The object in the *football* sequence (Fig. 5e) suffers from background clutters. Furthermore, the object also suffers from occlusion by other players, which make the sequence challenging. Overall, our tracker shows favorable performance to deal with the challenging sequence. The target in *faceocc* sequence in Fig. 5f undergoes heavy occlusion. The proposed tracker SACT achieves the best performance in terms of precision score and success score. Our tracker can handle occlusion variations and background clutters well as its adaptive appearance model and the online classifier update strategy. When the object appearance changes rapidly, a larger learning rate is applied. On the contrary, a smaller learning rate is applied. However, when $p < \Theta$, which means the current location of the object is not accurate or the occlusion has occurred, the classifier stops updating. In this way, the tracker is prevented from drifting due to avoiding adding inaccurate samples.

Multiple challenges

The objects in the *basketball* (Fig. 5g) and *soccer* (Fig. 5h) sequences both suffer from multiple challenges, such as fast motion, motion blur, background clutters, occlusion and other challenges, which make these two sequences much challenging. Consequently, all the trackers drift to the background or other objects gradually except our tracker. Overall, SACT achieves the best performance in these two challenging sequences due to its adaptive appearance model and the online classifier update strategy.

Conclusions

In this paper, we proposed a novel scale adaptive compressive tracking method, which improves the CT algorithm by a significantly large margin on the CVPR2013 tracking benchmark. Our method significantly improves CT in three aspects: Firstly, the scale of tracking box is adaptively adjusted according to the size of the objects. Secondly, the confidence coefficients of features are computed and used to achieve different contribution to the classifier. Finally, a variable learning parameter λ is adopted in our method, which can be adjusted according to the object appearance variation rate. Numerous experiments have shown that the superior performance of the proposed method over other 7 state-of-the-art tracking algorithms in dealing with scale and pose change, illumination change, background clutters, occlusion and multiple challenges.

Authors' contributions

The work presented here was carried out in collaboration between all authors. PZ and SC defined the research theme, designed methods and experiments, carried out the experiments, analyzed the data, interpreted the results and wrote the paper. MG and DF co-designed experiments, discussed analyses and presentation. All authors have contributed to, seen and approved the manuscript. All authors read and approved the final manuscript.

Acknowledgements

We would like to express our gratitude to all those who helped us during the writing of this paper. We gratefully acknowledge the help of Dr. Kai Chang and Xiaoran Guo. We do appreciate their patience, encouragement, and professional writing services during our paper writing. Also, we would like to thank Ph.D Max Haring and the reviewers of the manuscript. We would like to express our sincere appreciation to you for their professional handling and constructive comments on our manuscript.

Competing interests

The authors declare that they have no competing interests.

Received: 15 January 2016 Accepted: 13 May 2016

Published online: 23 June 2016

References

- Abraham Glincoy et al (2013) Sparsity based single object tracking. *Int J Comput Technol* 9(2):1004–1011
- Achlioptas D (2003) Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *J Comput Syst Sci* 66(4):671–687
- Adam A, Rivlin E, Shimshoni I (2006) Robust fragments-based tracking using the integral histogram. *Proc IEEE Conf Comput Vis Pattern Recognit* 1:798–805
- Avidan S (2004) Support vector tracking. *IEEE Trans Pattern Anal Mach Intell* 26:1064–1072
- Babenko B, Yang M-H, Belongie S (2011) Robust object tracking with online multiple instance learning. *IEEE Trans Pattern Anal Mach Intell* 33:1619–1632
- Babu RV, Perez P, Bouthemy P (2007) Robust tracking with motion estimation and local kernel-based color modeling. *Image Vis Comput* 25:1205–1216
- Candes EJ, Tao T (2005) Decoding by linear programming. *IEEE Trans Inf Theory* 51(12):4203–4215
- Candes EJ, Tao T (2006) Near-optimal signal recovery from random projections: universal encoding strategies. *IEEE Trans Inf Theory* 52(12):5406–5425
- Grabner H, Bischof H (2006) On-line boosting and vision. In: *Proceedings of IEEE computer society conference on computer vision and pattern recognition*, pp 260–267
- Grabner H, Leistner C, Bischof H (2008) Semi-supervised on-line boosting for robust tracking. *ECCV*, pp 234–247
- Jing J, Li Z, Zhu Q (2011) Influence of the outer secondary air vane angle on the gas particle flow characteristics near the double swirl flow burner region. *Energy* 36(1):258–267
- Li H, Shen C, Shi Q (2011) Real-time visual tracking using compressive sensing. *CVPR*, pp 1305–1312
- Lim J, Ross DA, Lin RS, Yang M-H (2004) Incremental learning for visual tracking. *Nips*, pp 793–800
- Liu B, Huang J, Kulikowski C, Yang L (2013) Robust visual tracking using local sparse appearance model and k-selection. *IEEE Trans Pattern Anal Mach Intell* 35:2968–2981
- Liu QS, Yang J, Zhang K, Wu Y (2015) Preprint submitted to pattern recognition: adaptive compressive tracking via online vector boosting feature selection. *arXiv: 1504.05451v2*
- Mei X, Ling H (2009) Robust visual tracking using l_1 minimization. In: *Proceedings of IEEE 12th international conference on computer vision*, pp 1436–1443
- Mei X, Ling H (2011) Robust visual tracking and vehicle classification via sparse representation. *IEEE Trans Pattern Anal Mach Intell* 33:2259–2272
- Ng A, Jordan M (2002) On discriminative vs. generative classifier: a comparison of logistic regression and naive Bayes. *NIPS*, pp 841–848
- Wu Y, Lim J, Yang M-H (2013) Online object tracking: a benchmark. In: *Proceedings of IEEE conference on computer vision and pattern recognition*, pp 2411–2418
- Zhang K, Zhang L, Yang M-H (2012) Real-time compressive tracking. In: *Proceedings of European conference on computer vision*, pp 864–877
- Zhang K, Lei Z, Yang M-H (2013) Real-time object tracking via online discriminative feature selection. *IEEE Trans Image Process* 22(12):4664–4677
- Zhang K, Lei Z, Yang M-H (2014a) Fast compressive tracking. *IEEE Trans Pattern Anal Mach Intell* 36(10):2002–2015
- Zhang W, Xiang Z, Yu H, Liu J (2014b) Object compressive tracking based on adaptive multi-feature appearance model. *J Zhejiang Univ* 48(12):2132–2138