Transfer Learning Based Visual Tracking with Gaussian Processes Regression

Jin Gao[1,2], Haibin Ling[2], Weiming Hu[1], and Junliang Xing[1]

[1] National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing, China
{jin.gao,wmhu,jlxing}@nlpr.ia.ac.cn
[2] Department of Computer and Information Sciences, Temple University, Philadelphia, USA
hbling@temple.edu

Abstract. Modeling the target appearance is critical in many modern visual tracking algorithms. Many tracking-by-detection algorithms formulate the probability of target appearance as exponentially related to the confidence of a classifier output. By contrast, in this paper we directly analyze this probability using Gaussian Processes Regression (GPR), and introduce a latent variable to assist the tracking decision. Our observation model for regression is learnt in a semi-supervised fashion by using both labeled samples from previous frames and the unlabeled samples that are tracking candidates extracted from the current frame. We further divide the labeled samples into two categories: *auxiliary samples* collected from the very early frames and *target samples* from most recent frames. The auxiliary samples are dynamically re-weighted by the regression, and the final tracking result is determined by fusing decisions from two individual trackers, one derived from the auxiliary samples and the other from the target samples. All these ingredients together enable our tracker, denoted as TGPR, to alleviate the drifting issue from various aspects. The effectiveness of TGPR is clearly demonstrated by its excellent performances on three recently proposed public benchmarks, involving 161 sequences in total, in comparison with state-of-the-arts.

1 Introduction

Visual tracking is a fundamental problem in computer vision with a wide range of applications such as augmented reality, event detection and human-computer interaction, to name a few. Due to the challenges in tracking arbitrary objects, especially the drastic object appearance changes caused by lighting conditions, object pose variations, and occlusion, a tracking system needs to adaptively update the observation model on-the-fly. A well-known danger of this updating over time, however, is the tendency to "drift".

There are several popular strategies in previous studies toward alleviating drift (§2). First, background information should be take into consideration to develop a discriminative tracker, as followed by many tracking-by-detection methods. Second, unlabeled samples from the current frame provide rich information in a semi-supervised manor, and can be used for enhancing the tracking inference. Third, re-weighting the training samples appropriately may help reduce the impact of the noisy and potential sample misalignment during model updating. Fourth, training samples should be adaptively to avoid the loss of sample diversity. Fifth, using the auxiliary data to assist the current online tracking task (e.g., using a transfer learning strategy) is preferable, because it can

D. Fleet et al. (Eds.): ECCV 2014, Part III, LNCS 8691, pp. 188–203, 2014.

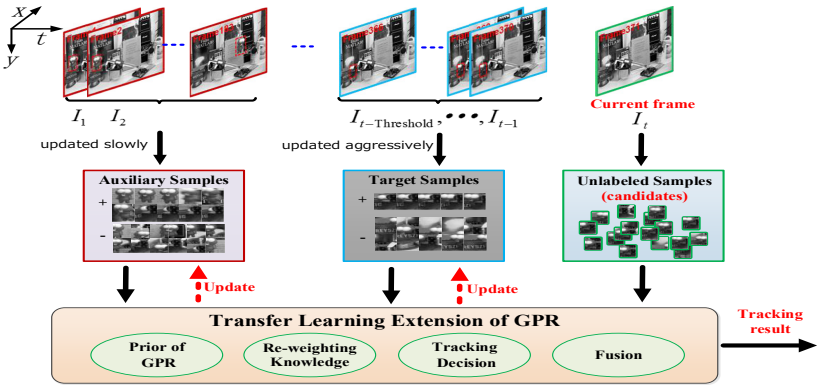

Fig. 1. Overview of the proposed TGPR tracking algorithm

reduce the drift resulting from the direct *Maximum a Posterior* (MAP) estimation over the noisy observation. Sixth, some part-based local representation methods are robust to the partial occlusion and small non-rigid deformation. Although these strategies have been exploited before, integrating all of them together remains challenging.

In this paper, we attack the challenge by proposing a new transfer learning based visual tracker using *Gaussian Processes Regression* (GPR). The new tracker, denoted as TGPR, naturally addresses the drifting issue from six aforementioned aspects.

First, we explicitly model the probability of target appearances in a GPR framework, and then a latent variable is naturally introduced to locate the best tracking candidates. In this process, the background information consists of the negative samples for regression. Also, the unlabeled samples (tracking candidates) are exploited when the prior of GPR is defined, so that the observation model is inferred in a semi-supervised fashion.

Second, we divide the training samples into two categories and treat them differently: the *auxiliary samples* (collected from the very early frames) are updated slowly and carefully; the *target samples* (from most recent frames) are updated quickly and aggressively. Such strategy allows us to re-weight the auxiliary samples, which is closely related to the current tracking status. The re-weighting helps to reduce the impact of the noisy and potential sample misalignment when the auxiliary samples are locate the best tracking candidates.

Third, the re-weighting of the auxiliary samples can be viewed as the knowledge that can be effectively exploited in a transfer learning framework. In particular, we adopt the task-transfer strategy [38], where the tracking decision using the re-weighted auxiliary samples assists the decision using target samples by fusing these two decisions. Their collaboration circumvents the direct *Maximum a Posterior* (MAP) estimation over the most likely noisy observation model, and allows the use of a new strategy similar to the *Minimum Uncertainty Gap* (MUG) estimation [19]. In addition, we define the prior of GPR by a local patch representation method to achieve robustness against occlusion.

Figure 1 overviews the proposed approach. For fairly evaluating the proposed tracker and reducing subjective bias as suggested by [28], we test TGPR on three recently proposed online tracking benchmarks: the CVPR2013 Visual Tracker Benchmark [35],

the Princeton Tracking Benchmark [30], and the VOT2013 Challenge Benchmark [16]. On all three benchmarks, involving in total 161 sequences, TGPR has achieved very promising results and outperforms previously tested state-of-the-arts.

2 Related Work

Model-Free Tracking. Single target visual tracking has long been attracting large amounts of research efforts [39]. It is impractical to enumerate all previous work, instead we sample some recent interests related to our work: i) linear representation with a dictionary, e.g., a set of basis vectors based on subspace learning [29,12] or least soft-threshold squares linear regression [32], a series of raw pixel templates based on sparse coding [25,24,44,43,36] or non-sparse linear representation [22]; ii) collaboration of multiple tracking models, e.g., Interacting *Markov Chain Monto Carlo* (MCMC) based [17,18,19], local/global combination based [45]; iii) part-based models, e.g., fragments voting based [1,9,5], incorporating spatial constraints between the parts [42,37], alignment-pooling across the local patches [14]; iv) and the widely followed tracking-by-detection (or discriminative) methods [6,7,20,2,8,21,31,45], which treat the tracking problem as a classification task. All these trackers adaptively update tracking models to accommodate the appearance changes and new information during tracking.

Alleviate Drifts. Much progress has been made in alleviating drifts. Previous strategies mainly consist of following aspects. i) Some studies [14,36,23] observe that straightforward and frequent update of new observations may cause gradual drifting due to accumulated errors and loss of sample diversity. So some strategies, e.g., slow update of old templates and quick update of new ones by assigning different update probability to them [14], multi-lifespan setting [36,23], are adopted. ii) Some studies [7,41,45,19] notice that appearance models are often updated with noisy and potentially misaligned samples, which often leads to drifting. So their solutions incorporate the data-independent knowledge, e.g., a fixed prior classifier trained by the labeled samples from the first frame [7], a measurement matrix for compressive sensing [41], a fixed dictionary for histogram generation [45], or utilize the MUG estimation instead of the MAP estimation [19]. iii) Some work [9,14], based on the part-based model, focuses on selectively updating the parts of the object to handle the tracking drift caused by heavy occlusion; other work [26,3,45] use occlusion detection strategy to determine whether the template should be updated with the new observation. iv) Many tracking-by-detection methods and some others [43,22] reduce the drifting effects by incorporating background samples.

Re-weight the Training Samples. Re-weighting tracking samples has been widely used in the sparse coding based tracking methods (e.g., [44,43]), however the importance of re-weighting the training samples is hardly observed in the tracking-by-detection methods with a few exceptions such as [22,8,31]. In [22] larger weights are assigned to the recently added samples while smaller weights to old ones using a time-weighted reservoir sampling strategy. Their re-weighting method is prone to drifting when the recently added samples are noisy or misaligned with the current tracking. In [8] the focus

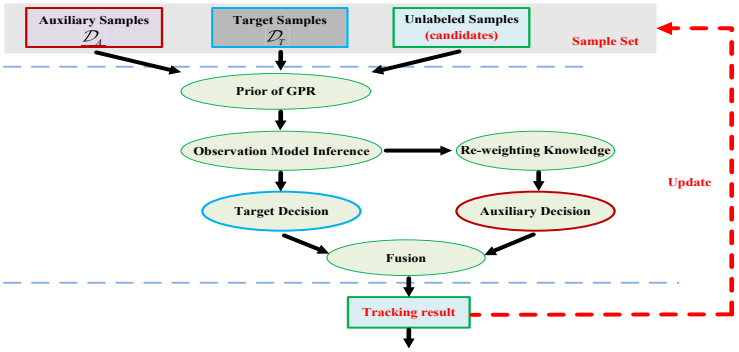

Fig. 2. The relationship among the components of the proposed TGPR tracker

is on re-weighting the support vectors by taking into account the current learner and a bounding box overlap based loss function. In [31] "good" frames are selected to learn a new model while revisiting past frames to correct mistakes made by previous models, which means that past frames are re-weighted to learn a new model. By contrast, our GPR-based solution re-weights all auxiliary samples by considering distances between all pairs of samples. Thus, distribution of unlabeled samples collected from the current frame strongly influences the modelling process.

Transfer Learning Based Tracking. Transfer learning has recently been applied to visual tracking (e.g., [20,34,33]). In [20], the "Covariate Shift" extension of the semi-supervised on-line boosting tracker [7] is proposed. Different than in our work, the auxiliary samples' re-weighting in [20] is based on the online boosting classifier. The methods in [34,33] transfer the prior knowledge from offline training on the real-world natural images to the current online target tracking task. By contrast, in our algorithm the prior knowledge is based on the online regression on the auxiliary samples.

3 The Proposed Tracking Approach

In this section, we first analyze the probability of the observation model in the Bayesian tracking framework and re-formulate it as a new objective. Then, we use GPR to solve this new formulation. Fig. 2 depicts the whole process.

3.1 New Objective of the Observation Model

Visual tracking can be cast as a sequential Bayesian inference problem [13]. Given a set of observed image patches $\mathcal{I}_t$ up to the t-th frame, we aim to estimate the value of the state variable $\boldsymbol{\ell}_t$, which describes the target location at time t. The true posterior state distribution $\Pr(\boldsymbol{\ell}_t|\mathcal{I}_t)$ is commonly approximated by a set of n_U samples, called tracking candidates, $\{\boldsymbol{\ell}_t^i, i = 1, 2, \ldots, n_U\}$, and $\boldsymbol{\ell}_t$ is estimated by MAP:

$$\hat{\boldsymbol{\ell}}_t = \arg\max_{\boldsymbol{\ell}_t^i} \Pr\left(\boldsymbol{\ell}_t^i|\mathcal{I}_t\right), \tag{1}$$

where $\boldsymbol{\ell}_t^i$ indicates the state of the i-th candidate of the state $\boldsymbol{\ell}_t$ on the t-th frame. The posterior probability $\Pr(\boldsymbol{\ell}_t|\mathcal{I}_t)$ can be inferred recursively,

$$\Pr(\boldsymbol{\ell}_t|\mathcal{I}_t) \propto \Pr(\mathbf{X}_t|\boldsymbol{\ell}_t) \int \Pr(\boldsymbol{\ell}_t|\boldsymbol{\ell}_{t-1}) \Pr(\boldsymbol{\ell}_{t-1}|\mathcal{I}_{t-1})\, d\boldsymbol{\ell}_{t-1}, \tag{2}$$

where $\Pr(\boldsymbol{\ell}_t|\boldsymbol{\ell}_{t-1})$ denotes the dynamic model, $\Pr(\mathbf{X}_t|\boldsymbol{\ell}_t)$ the observation model, and $\mathbf{X}_t$ the observation on the t-th frame. We use the same dynamic model as in [29], while focusing on the observation model.

Suppose we have stochastically generated a set of samples to model the distribution of the object location, i.e., $\mathcal{X}_U = \{\mathbf{X}_t^i, i = 1, 2, \ldots, n_U\}$ at the states (tracking candidates) $\{\boldsymbol{\ell}_t^i, i = 1, 2, \ldots, n_U\}$. We use an indicator variable $y_i \in \{1, -1\}$ to indicate "same" ($y_i = +1$) or "completely different" ($y_i = -1$) for $\mathbf{X}_t^i$. We call $\mathcal{X}_U$ as the unlabeled sample set. Then, we can re-formulate the observation model as

$$\Pr\left(\mathbf{X}_t^i|\boldsymbol{\ell}_t^i\right) \propto \Pr\left(y_i = +1|\mathbf{X}_t^i\right) \tag{3}$$

where the right hand is the likelihood that an observed image patch $\mathbf{X}_t^i$ having the "same" observation of the tracking object.

From the tracking results $\{\hat{\boldsymbol{\ell}_f}, f = 1, 2, \ldots, t-1\}$ up to the $(t-1)$-th frame, we extract n_L labeled training samples with the labels in $\{-1, +1\}$. Furthermore, we divide these samples into two categories and treat them differently: the *auxiliary samples* (from the very early frames) are updated slowly and carefully; the *target samples* (from most recent frames) are updated quickly and aggressively. Hereafter we denote $\mathcal{D}_T = \{(\mathbf{X}^j, y_j), j = 1, 2, \ldots, n_T\}$ as the target sample set, and $\mathcal{D}_A = \{(\mathbf{X}^j, y_j), j = n_T + 1, n_T + 2, \ldots, n_T + n_A\}$ the auxiliary sample set, where $n_L = n_T + n_A$ and y_j is the label in the sense of Eq. (3). Let $\mathbf{1} = [+1, +1, \ldots, +1]^\top$, the regression function for the indicators of the unlabeled samples $\mathbf{y}_U = [y_1, y_2, \ldots, y_{n_U}]^\top$ can be written as

$$\mathcal{R} = \Pr(\mathbf{y}_U = \mathbf{1}|\mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T) \quad . \tag{4}$$

3.2 Analyses

To analyze the regression $\mathcal{R}$ directly, we introduce two *real valued* latent vectors $\mathbf{z}_A \in \mathbb{R}^{n_A}$ and $\mathbf{z}_U \in \mathbb{R}^{n_U}$, underpinning the labels in $\mathbf{y}_A$ and $\mathbf{y}_U$, respectively. This way, $\mathcal{R}$ can be derived as marginalize over $\mathbf{z}_A$, $\mathbf{z}_U$:

$$\begin{aligned} \Pr(\mathbf{y}_U = \mathbf{1}|\mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T) &= \int\int \Pr(\mathbf{y}_U = \mathbf{1}|\mathbf{z}_A, \mathbf{z}_U, \mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T)\ d\mathbf{z}_A\, d\mathbf{z}_U \\ &= \int\int \Pr(\mathbf{y}_U = \mathbf{1}|\mathbf{z}_U)\, \mathbf{f}(\mathbf{z}_A, \mathbf{z}_U|\mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T)\ d\mathbf{z}_A\, d\mathbf{z}_U \ , \end{aligned} \tag{5}$$

where $\mathbf{f}(\mathbf{z}_A, \mathbf{z}_U|\mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T)$ is the joint probability density.

Analysis 1. Let $\mathbf{z}_U = [z_1, z_2, \ldots, z_{n_U}]^\top$, we model $\Pr(\mathbf{y}_U|\mathbf{z}_U)$ as a noisy label generation process $\mathcal{X}_U \to \mathbf{z}_U \to \mathbf{y}_U$ with the following sigmoid noise output model:

$$\Pr(y_i|z_i) = \frac{e^{\gamma z_i y_i}}{e^{\gamma z_i y_i} + e^{-\gamma z_i y_i}} = \frac{1}{1 + e^{-2\gamma z_i y_i}} \quad , \qquad \forall i = 1, 2, \ldots, n_U \tag{6}$$

where γ is a parameter controlling the steepness of the sigmoid.

The label generation process is similar for the auxiliary data, i.e., $\mathcal{X}_A \rightarrow \mathbf{z}_A \rightarrow \mathbf{y}_A$, where $\mathcal{X}_A = \{\mathbf{X}^j, j = n_T+1, n_T+2, \ldots, n_T+n_A\}$, $\mathbf{z}_A = [z_{n_T+1}, z_{n_T+2}, \ldots, z_{n_T+n_A}]^\top$, and $\mathbf{y}_A = [y_{n_T+1}, y_{n_T+2}, \ldots, y_{n_T+n_A}]^\top$. In this case, $\mathbf{z}_A$ can be viewed as the re-weighting knowledge extracted from the regression $\mathcal{R}$. Thus, $\mathbf{z}_A$ bridges the gap between the regression of the current tracking task and the indicators of the auxiliary samples. $\mathbf{z}_A$ can also be viewed as a soft substitution of $\mathbf{y}_A$, and is therefore less sensitive to noisy and potential sample misalignment.

Analysis 2. Applying the Bayes' theorem to $\mathbf{f}(\mathbf{z}_A, \mathbf{z}_U|\mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T)$, we have

$$\begin{aligned}\mathbf{f}(\mathbf{z}_A, \mathbf{z}_U|\mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T) &= \mathbf{f}(\mathbf{z}_A, \mathbf{z}_U|\mathcal{X}_U, \mathcal{X}_A, \mathbf{y}_A, \mathcal{D}_T) \\ &= \frac{\Pr(\mathbf{y}_A|\mathbf{z}_A, \mathbf{z}_U, \mathcal{X}_A, \mathcal{X}_U, \mathcal{D}_T) \bullet \mathbf{f}(\mathbf{z}_A, \mathbf{z}_U|\mathcal{X}_A, \mathcal{X}_U, \mathcal{D}_T)}{\Pr(\mathbf{y}_A|\mathcal{X}_A, \mathcal{X}_U, \mathcal{D}_T)} \\ &\propto \Pr(\mathbf{y}_A|\mathbf{z}_A) \bullet \mathbf{f}(\mathbf{z}_A, \mathbf{z}_U|\mathcal{X}_A, \mathcal{X}_U, \mathcal{D}_T) \quad . \end{aligned} \tag{7}$$

We model $\mathbf{f}(\mathbf{z}_A, \mathbf{z}_U|\mathcal{X}_A, \mathcal{X}_U, \mathcal{D}_T)$ with a Gaussian process, which can be specified by the mode $\boldsymbol{\mu}$ and the covariance matrix $\mathbf{G} \in \mathbb{R}^{(n_A+n_U)\times(n_A+n_U)}$, i.e.,

$$\Pr(\mathbf{z}_A, \mathbf{z}_U|\mathcal{X}_A, \mathcal{X}_U, \mathcal{D}_T) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{G}) \quad . \tag{8}$$

The non-Gaussianity of $\Pr(\mathbf{y}_A|\mathbf{z}_A)$ (see Analysis 1) makes the $\mathbf{f}(\mathbf{z}_A, \mathbf{z}_U|\mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T)$ no longer Gaussian, consequently Eq. (5) becomes analytically intractable. According to [11], assuming $\mathbf{f}(\mathbf{z}_A, \mathbf{z}_U|\mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T)$ to be uni-modal, we can consider instead its *Laplace approximation*. In place of the correct density we use an $(n_A + n_U)$-dimensional Gaussian measure with mode $\boldsymbol{\mu}' \in \mathbb{R}^{n_A+n_U}$ and covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{(n_A+n_U)\times(n_A+n_U)}$, where $\boldsymbol{\mu}' = \arg\max_{\mathbf{z}_A\in\mathbb{R}^{n_A}, \mathbf{z}_U\in\mathbb{R}^{n_U}} \mathbf{f}(\mathbf{z}_A, \mathbf{z}_U|\mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T)$. In the next we decompose this maximization over $\mathbf{z}_A$ and $\mathbf{z}_U$ separately.

Taking the logarithm of Eq. (7), we get the following objective function to maximize

$$\mathcal{J}(\mathbf{z}_A, \mathbf{z}_U) = \underbrace{\ln(\Pr(\mathbf{y}_A|\mathbf{z}_A))}_{Q_1(\mathbf{z}_A)} + \underbrace{\ln(\mathbf{f}(\mathbf{z}_A, \mathbf{z}_U|\mathcal{X}_A, \mathcal{X}_U, \mathcal{D}_T))}_{Q_2(\mathbf{z}_A, \mathbf{z}_U)} \quad . \tag{9}$$

Denote $\mathbf{z}^\top = \left(\mathbf{z}_A^\top\ \mathbf{z}_U^\top\right)$, $\mathbf{y}^\top = \left(\mathbf{y}_T^\top\ \mathbf{z}_A^\top\right)$, where $\mathbf{y}_T = [y_1, y_2, \ldots, y_{n_T}]^\top$. According to Eq. (8), we define Q_2 as

$$\begin{aligned} Q_2(\mathbf{z}_A, \mathbf{z}_U) &= -\frac{1}{2}\big(\ln(2\pi)^{n_A+n_U} + \ln|\mathbf{G}| + \big(\mathbf{z}-\boldsymbol{\mu}\big)^\top \mathbf{G}^{-1}\big(\mathbf{z}-\boldsymbol{\mu}\big)\big) \\ &= -\frac{1}{2}\big(\ln|\mathbf{G}_{\text{all}}| + \left(\mathbf{y}_T^\top\ \mathbf{z}^\top\right)\mathbf{G}_{\text{all}}^{-1}\begin{pmatrix}\mathbf{y}_T \\ \mathbf{z}\end{pmatrix}\big) + c_1 \qquad (10) \\ &= -\frac{1}{2}\big(\ln|\mathbf{G}_{\text{all}}| + \left(\mathbf{y}^\top\ \mathbf{z}_U^\top\right)\mathbf{G}_{\text{all}}^{-1}\begin{pmatrix}\mathbf{y} \\ \mathbf{z}_U\end{pmatrix}\big) + c_1 \quad , \qquad (11) \end{aligned}$$

where $\mathbf{G}_{\text{all}} = \begin{pmatrix}\mathbf{G}_{LL}\ \mathbf{G}_{LU} \\ \mathbf{G}_{UL}\ \mathbf{G}_{UU}\end{pmatrix}$ and $\mathbf{G}_{\text{all}}^{-1} = \begin{pmatrix}\mathbf{A}\ \mathbf{B} \\ \mathbf{B}^\top\ \mathbf{M}\end{pmatrix}$ are the $(n_L+n_U)\times(n_L+n_U)$ Gram matrix (symmetric, non-singular) and its inverse, and $c_1 \in \mathbb{R}$ summarizes all terms independent of $\mathbf{z}$. As the prior of GPR for our observation model, the matrix $\mathbf{G}_{\text{all}}$ is defined over all samples. $\boldsymbol{\mu}$ and $\mathbf{G}$ in Eq. (8) can be derived from $\mathbf{G}_{\text{all}}$ as follows.

Proposition 1. *By defining the prior Gram matrix* $\boldsymbol{G}_{\text{all}}$ *over all samples, we can determine* $\boldsymbol{\mu}$ *and* $\boldsymbol{G}$ *in Eq.* (8) *by* $\boldsymbol{\mu} = -\boldsymbol{M}^{-1}\boldsymbol{B}^{\top}\mathbf{y}_T$ *and* $\boldsymbol{G} = \boldsymbol{M}^{-1}$.

The derivation is based on Eq. (10) and can be found in the supplementary material[1].

Note $\mathbf{z}_U$ appears only in Q_2, and we can independently optimize $Q_2(\mathrm{z}_A, \bullet)$ w.r.t. z_U given $\hat{\mathbf{z}}_A$, where $(\hat{\mathbf{z}}_A, \hat{\mathbf{z}}_U) = \arg\max_{\mathbf{z}_A, \mathbf{z}_U} \mathcal{J}$. According to [11,47], by taking derivative of $Q_2(\mathrm{z}_A, \bullet)$ w.r.t. z_U, the optimal value $\hat{\mathrm{z}}_U$ can be derived as:

$$\hat{\mathrm{z}}_U = \mathbf{G}_{UL}\mathbf{G}_{LL}^{-1}\begin{pmatrix}\mathrm{y}_T \\ \hat{\mathrm{z}}_A\end{pmatrix} \quad . \tag{12}$$

Then, let $\mathrm{z}_U = \mathbf{G}_{UL}\mathbf{G}_{LL}^{-1}\begin{pmatrix}\mathrm{y}_T \\ \mathrm{z}_A\end{pmatrix}$ in Eq. (11), we can derive $\hat{\mathbf{z}}_A$ by Proposition 2.

Proposition 2. *The optimal value* $\hat{z}_A$ *is given by*

$$\hat{z}_A = \arg\max_{\mathbf{z}_A \in \mathbb{R}^{n_A}} \mathcal{J} = \arg\max_{\mathbf{z}_A \in \mathbb{R}^{n_A}} \sum_{j=n_T+1}^{n_L} \ln\left(\Pr\left(y_i|z_i\right)\right) - \frac{1}{2}\left(y_T^{\top}\ z_A^{\top}\right)\boldsymbol{G}_{LL}^{-1}\begin{pmatrix}y_T \\ z_A\end{pmatrix} + c_2 \ , \tag{13}$$

where $Q_1(\mathbf{z}_A) = \sum_{j=n_T+1}^{n_L} \ln\left(\Pr\left(y_i|z_i\right)\right)$ *and* $c_2 = c_1 - \frac{1}{2}\ln|\boldsymbol{G}_{\text{all}}|$.

The derivation is based on Eq. (11) and can be found in the supplementary material[1].

The above derivations in (12) and (13) help us to estimate the mode $\boldsymbol{\mu}'$. In fact, we can also estimate the covariance $\boldsymbol{\Sigma}$ and thus Eq. (5) is computationally feasible. That is because determining Eq. (5) reduces to computing $\Pr\left(\mathbf{y}_U = \mathbf{1}|\mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T\right) = \int \Pr\left(\mathbf{y}_U = \mathbf{1}|\mathbf{z}_U\right)\mathbf{f}\left(\mathbf{z}_U|\hat{\mathbf{z}}_A, \mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T\right)d\mathbf{z}_U$, and $\mathbf{f}\left(\mathbf{z}_U|\hat{\mathbf{z}}_A, \mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T\right)$ is approximated by a Gaussian parameterized by $\boldsymbol{\mu}'$ and $\boldsymbol{\Sigma}$ (see [11] for more details).

Analysis 3. We use an iterative Newton-Raphson scheme to find the optimal value $\hat{\mathrm{z}}_A$ in Proposition 2. Let $\rho(z_j) = (1+e^{-2\gamma z_j})^{-1}$, where $j = n_T+1, n_T+2, \ldots, n_T+n_A$. Since $y_j \in \{-1, +1\}$, the auxiliary data generation model can be written as

$$\Pr\left(y_j|z_j\right) = \frac{e^{\gamma z_j y_j}}{e^{\gamma z_j y_j} + e^{-\gamma z_j y_j}} = \rho(z_j)^{\frac{y_j+1}{2}}(1-\rho(z_j))^{\frac{1-y_j}{2}} \quad , \tag{14}$$

therefore

$$Q_1(\mathbf{z}_A) = \gamma\left(\mathbf{y}_A - 1\right)^{\top}\mathbf{z}_A - \sum_{j=n_T+1}^{n_L}\ln\left(1+e^{-2\gamma z_j}\right) \quad . \tag{15}$$

Let $\mathbf{G}_{LL}^{-1} = \begin{pmatrix}\mathbf{F}_{TT} & \mathbf{F}_{TA} \\ \mathbf{F}_{AT} & \mathbf{F}_{AA}\end{pmatrix}$, we can estimate $\hat{\mathrm{z}}_A$ by taking derivative of $\mathcal{J}$ w.r.t. z_A,

$$\frac{\partial\mathcal{J}}{\partial\mathrm{z}_A} = \gamma(\mathrm{y}_A - \mathbf{1}) + 2\gamma\left(\mathbf{1} - \boldsymbol{\rho}(\mathrm{z}_A)\right) - \mathbf{F}_{AA}\mathrm{z}_A - \frac{1}{2}\mathbf{F}_{TA}^{\top}\mathrm{y}_T - \frac{1}{2}\mathbf{F}_{AT}\mathrm{y}_T \ , \tag{16}$$

[1] `http://www.dabi.temple.edu/~hbling/code/TGPR.htm`

where $\boldsymbol{\rho}(\mathrm{z}_A) = [\rho(z_{n_T+1}), \rho(z_{n_T+2}), \ldots, \rho(z_{n_L})]^\top$. The term $\boldsymbol{\rho}(\mathrm{z}_A)$ makes it impossible to compute $\hat{\mathrm{z}}_A$ in a closed form. Instead we use Newton-Raphson algorithm,

$$\mathrm{z}_A^{m+1} \leftarrow \mathbf{z}_A^m - \eta \mathbf{H}^{-1} \frac{\partial \mathcal{J}}{\partial \mathrm{z}_A}\Big|_{\mathrm{z}_A^m} \tag{17}$$

where $\eta \in \mathbb{R}^+$ is chosen so that $\mathcal{J}^{m+1} > \mathcal{J}^m$, and $\mathbf{H}$ is the Hessian matrix defined as

$$\mathbf{H} = \left[\frac{\partial^2 \mathcal{J}}{\partial z_i \partial z_j}\Big|_{\mathrm{z}_A}\right] = -\mathbf{F}_{AA} - \mathbf{P} \tag{18}$$

where $\mathbf{P}$ is a diagonal matrix with elements $P_{ii} = 4\gamma^2 \rho(z_i)(1 - \rho(z_i))$.

Analysis 4. An important aspect of GPR in our model lies in constructing the prior Gram or kernel matrix $\mathbf{G}_{\mathrm{all}}$ in (11). A popular way is to define the matrix entries in a "local" manner. For example, in a radial basis function (RBF) kernel $\mathbf{K}$, the matrix element $k_{ij} = \exp(-d_{ij}^2/\alpha^2)$ depends only on the distance d_{ij} between the i, j-th items. Such definition ignores the information encoded in unlabeled samples. Addressing this issue, we define the Gram matrix $\mathbf{G}_{\mathrm{all}}$ based on a weighted graph to explore the manifold structure of all samples (both labelled and unlabeled), as suggested in [46,47] following the intuition that similar samples often share similar labels.

Consider a graph $\mathcal{G} = (V, E)$ with the node set $V = T \cup A \cup U$ corresponding to all $n = n_L + n_U$ samples, where $T = \{1, \ldots, n_T\}$ denotes labeled target samples, $A = \{n_T+1, \ldots, n_T+n_A\}$ the labeled auxiliary samples, and $U = \{n_L+1, \ldots, n_L+n_U\}$ the unlabeled samples. We define weight matrix $\mathbf{W} = [w_{ij}] \in \mathbb{R}^{n \times n}$ on the edges of the graph using the local patch representation in [12]. This benefits the robust tracking, especially under partial occlusion. For the i-th and j-th samples, the weight w_{ij} is defined by the spatially weighted log-Euclidean Riemannian Metric over block-based covariance descriptors. Specifically, for the i-th sample, we first divide its image patch into $N_{\mathrm{r}} \times N_{\mathrm{c}}$ blocks, and then describe its (p, q)-th block by a covariance matrix $\mathbf{C}_i^{pq}$. Specifically, w_{ij} is defined as

$$w_{ij} = \frac{1}{\sum_{p,q} \beta_{p,q}} \sum_{p,q} \beta_{p,q} \exp\Big(-\frac{\|\log \mathbf{C}_i^{pq} - \log \mathbf{C}_j^{pq}\|^2}{\sigma_i^{pq} \sigma_j^{pq}}\Big) \tag{19}$$

where σ_i^{pq} is a local scaling factor proposed by [40]; $\beta_{p,q} = \exp(-\frac{\|\mathrm{pos}^{pq} - \mathrm{pos}^o\|^2}{2\sigma_{\mathrm{spatial}}^2})$ is the spatial weight, in which pos^{pq} indicates the position of the (p, q)-th block, pos^o the position of the block center, and $\sigma_{\mathrm{spatial}}$ the scaling factor.

Instead of connecting all pairs of nodes in V, we restrict the edges to be within the k-nearest-neighborhood, where k controls the density of the graph and the sparsity of $\mathbf{W}$. We can hence define the combinatorial Laplacian $\boldsymbol{\Delta}$ of $\mathcal{G}$ in the matrix form as $\boldsymbol{\Delta} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D} = \mathrm{diag}(D_{ii})$ is the diagonal matrix with $D_{ii} = \sum_j w_{ij}$.

Finally, we define the Gram matrix as $\mathbf{G}_{\mathrm{all}} = (\boldsymbol{\Delta} + \mathbf{I}/\lambda^2)^{-1}$, where the regularization term $\mathbf{I}/\lambda^2$ guards $\boldsymbol{\Delta} + \mathbf{I}/\lambda^2$ from being singular. From the definition of $\mathbf{G}_{\mathrm{all}}$ we can see that, the prior covariance in Eq. (11) between any two samples i, j in general depends

Algorithm 1. Transfer with GPR for Tracking

Input: Target sample set $\mathcal{D}_T$, auxiliary sample set $\mathcal{D}_A$, and unlabeled sample dataset $\mathcal{X}_U$

Output: The node set V_{res} (with size limit n_V) of the unlabeled samples that are most likely to belong to the tracking object.

if $n_A \leq \text{Threshold}$ **then**
  Calculate $\mathbf{W}_t$ over the target and unlabeled samples from Eq. (19);
  Construct $\mathbf{G}_{\text{all}}^t$ according to Analysis 4;
  Target tracking: $\hat{\mathbf{z}_U^t} = \mathbf{G}_{UT}\mathbf{G}_{TT}^{-1}\mathbf{y}_T$;
  $[\bullet, \text{Idx}_t] = \text{sort}(\hat{\mathbf{z}_U^t}, \text{'descend'})$;
  $V_{\text{res}} = \text{Idx}_t(1 : n_V)$;
else
  Calculate $\mathbf{W}$ over all the target, auxiliary and unlabeled samples from Eq. (19);
  Construct $\mathbf{G}_{\text{all}}$ according to Analysis 4;
  Calculate $\hat{\mathbf{z}_A}$ from Eq. (17) until convergence;
  Let $\mathbf{W}_a = \mathbf{W}(n_T + 1 : n,\ n_T + 1 : n)$ and construct $\mathbf{G}_{\text{all}}^a$ according to Analysis 4;
  Auxiliary tracking: $\hat{\mathbf{z}_U^a} = \mathbf{G}_{UA}\mathbf{G}_{AA}^{-1}\hat{\mathbf{z}}_A$;
  Construct $\mathbf{W}_t = \begin{pmatrix} \mathbf{W}(1 : n_T,\ 1 : n_T) & \mathbf{W}(1 : n_T,\ n_L + 1 : n) \\ \mathbf{W}(n_L + 1 : n,\ 1 : n_T) & \mathbf{W}(n_L + 1 : n,\ n_L + 1 : n) \end{pmatrix}$;
  Construct $\mathbf{G}_{\text{all}}^t$ according to Analysis 4;
  Target tracking: $\hat{\mathbf{z}_U^t} = \mathbf{G}_{UT}\mathbf{G}_{TT}^{-1}\mathbf{y}_T$;
  /* Fusing two trackers, 'pool' is the size of candidate pool */
  $[\bullet, \text{Idx}_a] = \text{sort}(\hat{\mathbf{z}_U^a}, \text{'descend'})$;
  $[\bullet, \text{Idx}_t] = \text{sort}(\hat{\mathbf{z}_U^t}, \text{'descend'})$;
  $V_A = \text{Idx}_a(1 : \text{pool}) \setminus \{i : \text{Idx}_a(i) \notin \text{Idx}_t(1 : \text{pool})\}$;
  $V_T = \text{Idx}_t(1 : \text{pool}) \setminus \{i : \text{Idx}_t(i) \notin \text{Idx}_a(1 : \text{pool})\}$;
  if $|V_A| > \text{pool}/2$ **then**
    $V_{\text{res}} = V_A(1 : \min(n_V, \text{pool}/2))$;
  else if $|V_A| = 0$ **then**
    $V_{\text{res}} = \text{Idx}_a(1 : n_V)$;
  else
    $V_{\text{res}} = V_T(1 : \min(n_V, |V_A|))$;
  end if
end if

on all samples – all the target and unlabeled samples are used to define the prior. Thus, distribution of target and unlabeled samples may strongly influence the kernel, which is desired when we extract the re-weighting knowledge $\mathbf{z}_A$.

3.3 Fusion Based Transfer Learning Extension

The value of a latent variable in $\hat{\mathrm{z}}_U$ can be viewed as a soft version of tracking decision. Consequently, our tracker can be based on using $\hat{\mathrm{z}}_U$ to decide which samples most likely have the "same" observations to the object. The larger the value of $\hat{z}_i$ in $\hat{\mathrm{z}}_U$, the more likely the sample has the "same" observation. However, we do not directly use Eq. (12) to compute $\hat{\mathrm{z}}_U$ for tracking. This is because the unlabeled samples relate more to the target samples than to the auxiliary ones, and direct use of Eq. (12) may

overfit the target samples and is vulnerable to the misaligned target samples or occlusion. Alternatively, we use the re-weighted auxiliary samples and the target samples to build two individual trackers. Then, the auxiliary decision (made by the re-weighted auxiliary samples) assists the target decision (made by the target samples) by fusing the two trackers. This can be thought as a task-transfer process, in which the re-weighting knowledge is transferred from the auxiliary decision to the target decision.

These two trackers can be derived based on §3.2. Given all the labeled (auxiliary and target) and unlabeled samples, i.e., $(\mathcal{X}_L, \mathbf{y}_L)$ and $\mathcal{X}_U$, Eq. (5) can be reduced to $\Pr(\mathbf{y}_U = \mathbf{1}|\mathcal{X}_U, \mathcal{X}_L, \mathbf{y}_L) = \int \Pr(\mathbf{y}_U = \mathbf{1}|\mathbf{z}_U)\ \mathbf{f}(\mathbf{z}_U|\mathcal{X}_U, \mathcal{X}_L, \mathbf{y}_L)\, d\mathbf{z}_U$. Meanwhile, the Gaussian distribution in Eq. (8) is reduced to $\Pr(\mathbf{z}_U|\mathcal{X}_U, \mathcal{X}_L, \mathbf{y}_L) \sim \mathcal{N}(\boldsymbol{\mu}_L, \mathbf{G}_L)$. According to Proposition 1, let $\mathbf{y}_T = \mathbf{y}_L$ and $\mathbf{z} = \mathbf{z}_U$ in Eq. (11), we can find the optimal estimation of $\mathbf{z}_U$ by $\hat{\mathbf{z}}_U = \boldsymbol{\mu}_L = -\mathbf{M}_L^{-1}\mathbf{B}_L^\top \mathbf{y}_L$, where $\mathbf{G}_{\text{all}} = \begin{pmatrix} \mathbf{G}_{LL} & \mathbf{G}_{LU} \\ \mathbf{G}_{UL} & \mathbf{G}_{UU} \end{pmatrix}$ and $\mathbf{G}_{\text{all}}^{-1} = \begin{pmatrix} \mathbf{A}_L & \mathbf{B}_L \\ \mathbf{B}_L^\top & \mathbf{M}_L \end{pmatrix}$ are the Gram matrix and its inverse over all samples. The blocks in $\mathbf{G}_{\text{all}}^{-1}$ can be derived as $\mathbf{B}_L^\top = -\mathbf{M}_L\mathbf{G}_{UL}\mathbf{G}_{LL}^{-1}$. Consequently, we have $\hat{\mathbf{z}}_U = \mathbf{G}_{UL}\mathbf{G}_{LL}^{-1}\mathbf{y}_L$. This is consistent to the harmonic property proposed in [46,47], which shows that the value of soft label $\hat{z}_i$ at each unlabeled sample is the average of label values from its neighborhood.

With the above derivation, we can perform the two tracking algorithms respectively using the re-weighted auxiliary samples and the target samples:

- **Auxiliary Tracking Using $\hat{\mathbf{z}}_U^a$**: use the auxiliary samples $\mathcal{X}_A$ as labeled samples with labels $\hat{\mathbf{z}}_A$; construct the prior Gram matrix $\mathbf{G}_{\text{all}}^a = \begin{pmatrix} \mathbf{G}_{AA} & \mathbf{G}_{AU} \\ \mathbf{G}_{UA} & \mathbf{G}_{UU} \end{pmatrix}$ according to Analysis 4; then the soft labels of unlabeled samples can be determined by the auxiliary samples as $\hat{\mathbf{z}}_U^a = \mathbf{G}_{UA}\mathbf{G}_{AA}^{-1}\hat{\mathbf{z}}_A$.
- **Target Tracking Using $\hat{\mathbf{z}}_U^t$**: use the target samples $\mathcal{X}_T$ as labeled samples with labels $\mathbf{y}_T$; construct the prior Gram matrix $\mathbf{G}_{\text{all}}^t = \begin{pmatrix} \mathbf{G}_{TT} & \mathbf{G}_{TU} \\ \mathbf{G}_{UT} & \mathbf{G}_{UU} \end{pmatrix}$ according to Analysis 4; then the soft labels of unlabeled samples can be determined by the target samples as $\hat{\mathbf{z}}_U^t = \mathbf{G}_{UT}\mathbf{G}_{TT}^{-1}\mathbf{y}_T$.

Finally, we use a heuristic fusion method to regularize the target decision with the assistance of the auxiliary decision. Specifically, when obtaining two positive candidate sets according to these two decisions separately, we check the two sets' coincidence degree, e.g., $|V_A|$ in Algorithm 1. When the degree is high, it does not matter whether we rely on the auxiliary decision or the target decision; when the degree is small, we rely more on the target decision to ensure the consistency of the tracking results; when the degree is zero, we rely more on the auxiliary decision to recover from the severe appearance variation and heavy occlusion. We detail this procedure in Algorithm 1. When the node set V_{res} in Algorithm 1 is obtained, the object location can be determined by the average over locations of the samples indexed by these nodes.

4 Experiments

It is not easy to thoroughly evaluate a tracking algorithm without subjective bias [28], due to the influence from many factors such as sequence selection and parameter tuning. Several notable recent efforts [35,30,16] have been devoted to address this issue by proposing tracking benchmarks. Aligning with these efforts, we evaluate the proposed TGPR tracker over these benchmarks by following rigorously their evaluation protocols. In summary, TGPR is run on a total of 161 sequences and has achieved excellent performances in all the benchmarks.

4.1 Implementation Details

The proposed algorithm is implemented in C++ and evaluated on a desktop with a 3.40GHz CPU and 8GB RAM. The running time is about 3∼4 frames per second. This C++ implementation of TGPR is publicly available[1].

Samples Collection. We use the dynamic model proposed by [29] for collecting unlabeled samples $\mathcal{X}_U$ from the current frame I_t, where we only consider the variations of 2D translation $(\vec{x}_t, \vec{y}_t)$ and scale (s_t) in the affine transformation, and set the number n_U of particles to 300. When the conditions of lines 22 and 24 in Algorithm 1 are met, the parameter settings of $(\vec{x}_t, \vec{y}_t)$ and n_U are increased by a factor of 1.5. As for $\mathcal{D}_T$, we use the tracking results of past 10 frames $I_{t-10}, \ldots, I_{t-1}$ (or less than 10 in the beginning of tracking) as the positive target samples; the negative target samples are sampled from the frame I_{t-1} around its tracking result $(\vec{x}^*_{t-1}, \vec{y}^*_{t-1}, s^*_{t-1})$, using dense sampling method similar to [20] (overlap ratio is 0.11) in the sliding region, i.e., $\{\mathbf{X} : \boldsymbol{\ell}(\mathbf{X}) \in (\mathrm{R}(\vec{x}^*_{t-1}, \vec{y}^*_{t-1}, 2s^*_{t-1}) - R(\vec{x}^*_{t-1}, \vec{y}^*_{t-1}, s^*_{t-1}))\}$, where $\boldsymbol{\ell}(\mathbf{X})$ denotes the location of negative target sample $\mathbf{X}$, $\in$ means the center location of $\mathbf{X}$ lies in a certain image region, and $R(\vec{x}, \vec{y}, s)$ denotes the image region corresponding to the affine transformation $(\vec{x}, \vec{y}, s)$. Then, we randomly sample 64 negative target samples. For the purpose of updating the auxiliary set slowly, we collect the auxiliary samples $\mathcal{D}_A$ from the frames before $t - 10$ at intervals of 3 (or 6 for long-term tracking) frames, if these frames are available. The collection in such frames is the same to the collection of labeled samples in [20]. We set the size limit of positive auxiliary sample buffer to 50, and negative auxiliary sample buffer to 200.

Parameter Settings. Note that these settings are fixed for all experiments. In Analysis 4, the weight (Eq. (19)) of $\mathbf{W}$ is calculated by setting $N_\mathrm{r} = N_\mathrm{c} = 3$, $\sigma_\mathrm{spatial} = 3.9$ and σ_i^{pq} calculated from the $7th$ nearest neighbor. The hyperparameter k for controlling the sparsity of $\mathbf{W}$ is set to 50. The Gram matrix is defined by setting $\lambda = 1000$. In Analysis 3, γ in Eq. (6) is set to be 10, η in Eq. (17) is 0.4, and the number of iterations for calculating $\hat{\mathbf{z}_A}$ from Eq. (17) is 15. In Algorithm 1, the size limit n_V of the output V_res is set to be 5, $\mathrm{Threshold}$ is 30, and pool is 20.

4.2 Experiment 1: CVPR2013 Visual Tracker Benchmark

The CVPR2013 Visual Tracker Benchmark [35] contains 50 fully annotated sequences. These sequences include many popular sequences used in the online tracking literature

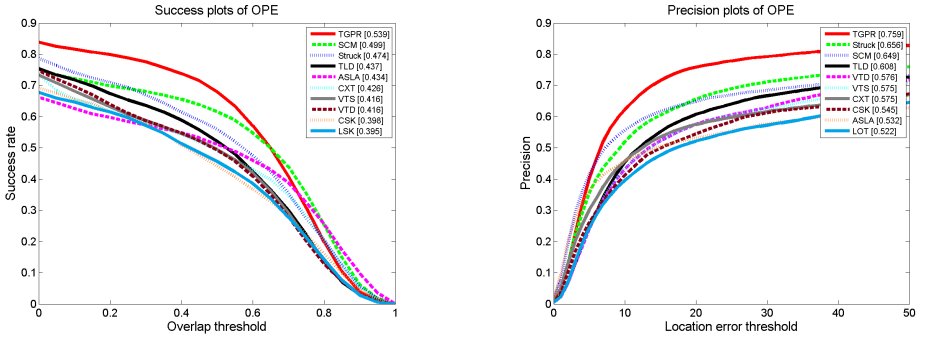

Fig. 3. Plots of OPE on the CVPR2013 Visual Tracker Benchmark. The performance score for each tracker is shown in the legend. For each figure, the top 10 trackers are presented for clarity (best viewed on high-resolution display.

over the past several years. For better evaluation and analysis of the strength and weakness of tracking approaches, these sequences are annotated with the 11 attributes including illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutters, and low resolution.

The providers have evaluated 29 tracking algorithms and released their results along with the sequences. To analyze the performances of different algorithms, the **precision plots** based on the location error metric and the **success plots** based on the overlap metric are adopted. In addition, the providers propose three kinds of robustness evaluation strategies: **OPE** (one-pass evaluation), **TRE** (temporal robustness evaluation), **SRE** (spatial robustness evaluation).

Results. Due to space limitations, we only show the overall performance of OPE for our tracker and compare it with some other state-of-the-arts (ranked within top 10) as shown in Fig. 3. These trackers include Struck [8], SCM [45], TLD [15], ASLA [14], VTD [17], VTS [18], CXT [4], LSK [24], CSK [10], MTT [44] and LOT [27]. Note that all the plots are automatically generated by the code library supported by the benchmark providers. From Fig. 3, we see that: (1) in the success plot, our proposed tracker TGPR outperforms the second best tracker SCM by 8.0%; and (2) in the precision plot, TGPR outperforms the second best tracker Struck by 15.7%.

Note that due to space limitation, we only include the above representative results and leaves more details in the supplementary material. It worth pointing out that, as shown in [35], the results (especially the top ones) in OPE are in general consistent with those in TRE and SRE.

4.3 Experiment 2: Princeton Tracking Benchmark

In the Princeton Tracking Benchmark [30], the providers captured a new benchmark by recording 100 video clips with both RGB and depth data using a standard Microsoft Kinect 1.0. In spite of some constraints due to acquisition (e.g., captured indoors, with

Table 1. Results on the Princeton Tracking Benchmark: successful rates and rankings (in parentheses) for different categorizations. The best results are in **red** and the second best in blue.

Alg.	Avg.	target type			target size		movement		occlusion		motion type	
	Rank	human	animal	rigid	large	small	slow	fast	yes	no	passive	active
TGPR	**1.09**	**0.46(1)**	0.49(2)	**0.67(1)**	**0.56(1)**	**0.53(1)**	**0.66(1)**	**0.50(1)**	**0.44(1)**	**0.69(1)**	**0.67(1)**	**0.50(1)**
Struck	2.82	0.35(2)	0.47(3)	0.53(4)	0.45(2)	0.44(4)	0.58(2)	0.39(2)	0.30(4)	0.64(2)	0.54(4)	0.41(2)
VTD	3.18	0.31(5)	**0.49(1)**	0.54(3)	0.39(4)	0.46(2)	0.57(3)	0.37(3)	0.28(5)	0.63(3)	0.55(3)	0.38(3)
RGBdet	4.36	0.27(7)	0.41(5)	0.55(2)	0.32(7)	0.46(3)	0.51(5)	0.36(4)	0.35(2)	0.47(6)	0.56(2)	0.34(5)
CT	5.36	0.31(4)	0.47(4)	0.37(7)	0.39(3)	0.34(7)	0.49(6)	0.31(5)	0.23(8)	0.54(4)	0.42(7)	0.34(4)
TLD	5.64	0.29(6)	0.35(7)	0.44(5)	0.32(6)	0.38(5)	0.52(4)	0.30(7)	0.34(3)	0.39(7)	0.50(5)	0.31(7)
MIL	5.82	0.32(3)	0.37(6)	0.38(6)	0.37(5)	0.35(6)	0.46(7)	0.31(6)	0.26(6)	0.49(5)	0.40(8)	0.34(6)
SemiB	7.73	0.22(8)	0.33(8)	0.33(8)	0.24(8)	0.32(8)	0.38(8)	0.24(8)	0.25(7)	0.33(8)	0.42(6)	0.23(8)
OF	9.00	0.18(9)	0.11(9)	0.23(9)	0.20(9)	0.17(9)	0.18(9)	0.19(9)	0.16(9)	0.22(9)	0.23(9)	0.17(9)

object depth values ranging from 0.5 to 10 meters), the dataset is valuable for evaluating the state-of-the-art visual tracking algorithms (only use the RGB data). This benchmark dataset presents varieties in the following aspects: target type, scene type, presence of occlusion, bounding box location and size distribution, and bounding box variation over time.

Along with the dataset, the providers also provide the evaluation results of the success rates measured by overlap ratio for eight state-of-the-art trackers (with RGB input) and eight RGBD competitors (with RGBD input). For fair comparison, we only compare the proposed TGPR tracker with the eight RGB competitors, including Struck [8], VTD [17], CT [41], TLD [15], MIL [2], SemiB [7] and the other 2 RGB baseline algorithms provided by the benchmark providers, RGBdet [30] and OF [30].

Results. The groundtruth of 95 out of the 100 sequences is reserved by the providers to reduce the chance for data-specific tuning. Following the instruction in `http://tracking.cs.princeton.edu/submit.php`, we submitted our tracking results online and obtained the evaluation results compared with the other RGB trackers as shown in Table 1. The results show that TGPR again outperforms other state-of-the-arts in almost all categories.

4.4 Experiment 3: VOT2013 Challenge Benchmark

The visual object tracking VOT2013 Challenge Benchmark [16] provides an evaluation kit and the dataset with 16 fully annotated sequences for evaluating tracking algorithms in realistic scenes subject to various common conditions. Following the protocol, we integrate our tracker TGPR into the VOT2013 evaluation kit, which automatically performs a standardized experiment on the tracking algorithm.

The tracking performance in the VOT2013 Challenge Benchmark is primarily evaluated by the following measures with a different view from the common evaluation criteria. **Accuracy (acc.):** This measure is the average of the overlap ratios over the valid frames of each sequence. The possible values are in the range of $[0, 1]$. **Robustness (rob.):** The tracker's robustness is evaluated by the total number of failures over

Table 2. The results of our tracker TGPR on the VOT2013 Challenge Benchmark. All the values are averaged by running each test on each sequence 15 times.

		bicycle	bolt	car	cup	david	diving	face	gym.	hand	iceskater	juice	jump	singer	sunshade	torus	woman
A	acc.	0.60	0.57	0.45	0.83	0.58	0.33	0.85	0.57	0.56	0.60	0.76	0.59	0.65	0.73	0.78	0.74
	rob.	0	1.27	0.40	0	0.27	2.87	0	2.87	1.67	0	0	0	0.60	0.20	0.13	1.00
B	acc.	0.57	0.57	0.41	0.75	0.58	0.32	0.77	0.53	0.53	0.57	0.73	0.57	0.45	0.64	0.65	0.67
	rob.	0	1.27	0.20	0	0.27	2.87	0.07	3.00	2.07	0	0	0	0.33	0.07	0.60	1.00

15 runs. In particular, a failure is detected once the overlap ratio measure drops to zero. When a failure happens, an operator re-initializes the tracker so it can continue. An equivalent of the number of required manual interventions per sequence is recorded and used as a comparative score.

We run TGPR in two types of test following the benchmark protocol. **Test A:** TGPR was run on each sequence in the dataset 15 times by initializing it on the ground truth bounding box, obtaining average statistic scores of the measures. **Test B:** TGPR was run, initialized with 15 noisy bounding boxes in each sequence, i.e., bounding boxes randomly perturbed in order of ten percent of the ground truth bounding box size. Then, average statistic scores of the measures are obtained.

Results. Because the VOT does not provide their ranking-based evaluation systems to public, we can report the results of our tracker in Table 2. That said, the table shows the great effectiveness of TGPR, with the failure rate often equal to 0, and most overlapping ratios above 0.5. Meanwhile, Table 2 shows that our tracker is not that sensitive to different initializations.

5 Conclusion

We proposed a new transfer learning based tracking algorithm with Gaussian Processes Regression (GPR). Specifically, GPR is innovatively exploited to make a new objective of the observation model, and then a simple but effective task-transfer tracking framework is extended so that drift problems can be alleviated from various aspects. We have also used a local patch representation method based graph Laplacian to define the prior Gram matrix in GPR, so that the distribution of target and unlabeled samples may strongly influence the transferred re-weighting knowledge. We have performed thorough evaluations on three public benchmarks and TGPR has generated very promising results by outperforming many state-of-the-arts.

Acknowledgement. Ling is supported in part by the US NSF Grant IIS-1218156 and the US NSF CAREER Award IIS-1350521. The others are partially supported by NSFC (Grant No. 60935002, Grant No. 61303178), the National 863 High-Tech R&D Program of China (Grant No. 2012AA012504), the Natural Science Foundation of Beijing (Grant No. 4121003), and The Project Supported by Guangdong Natural Science Foundation (Grant No. S2012020011081).

References

1. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: CVPR (2006)
2. Babenko, B., Yang, M.-H., Belongie, S.: Visual tracking with online multiple instance learning. In: CVPR (2009)
3. Bao, C., Wu, Y., Ling, H., Ji, H.: Real time robust ℓ_1 tracker using accelerated proximal gradient approach. In: CVPR (2012)
4. Dinh, T.B., Vo, N., Medioni, G.: Context tracker: Exploring supporters and distracters in unconstrained environments. In: CVPR (2011)
5. Erdem, E., Dubuisson, S., Bloch, I.: Fragments based tracking with adaptive cue integration. Computer Vision and Image Understanding 116(7), 827–841 (2012)
6. Grabner, H., Grabner, M., Bischof, H.: Real-time tracking via on-line boosting. In: BMVC (2006)
7. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised on-line boosting for robust tracking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 234–247. Springer, Heidelberg (2008)
8. Hare, S., Saffari, A., Torr, P.H.S.: Struck: Structured output tracking with kernels. In: ICCV (2011)
9. He, S., Yang, Q., Lau, R., Wang, J., Yang, M.-H.: Visual tracking via locality sensitive histograms. In: CVPR (2013)
10. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part IV. LNCS, vol. 7575, pp. 702–715. Springer, Heidelberg (2012)
11. Herbrich, R.: Kernel classifiers from a bayesian perspective. Learning Kernel Classifiers: Theory and Algorithms. MIT Press (2002)
12. Hu, W., Li, X., Luo, W., Zhang, X., Maybank, S., Zhang, Z.: Single and multiple object tracking using log-euclidean riemannian subspace and block-division appearance model. Trans. on PAMI 34(12), 2420–2440 (2012)
13. Isard, M., Blake, A.: Condensation - conditional density propagation for visual tracking. International Journal of Computer Vision 29(1), 5–28 (1998)
14. Jia, X., Lu, H., Yang, M.-H.: Visual tracking via adaptive structural local sparse appearance model. In: CVPR (2012)
15. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. Trans. on PAMI 34(7), 1409–1422 (2012)
16. Kristan, M., Pflugfelder, R., et al.: The visual object tracking vot2013 challenge results. In: Vis. Obj. Track. Challenge VOT 2013, In conjunction with ICCV (2013)
17. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: CVPR (2010)
18. Kwon, J., Lee, K.M.: Tracking by sampling trackers. In: ICCV (2011)
19. Kwon, J., Lee, K.M.: Minimum uncertainty gap for robust visual tracking. In: CVPR (2013)
20. Li, G., Qin, L., Huang, Q., Pang, J., Jiang, S.: Treat samples differently: object tracking with semi-supervised online covboost. In: ICCV (2011)
21. Li, X., Shen, C., Dick, A., van den Hengel, A.: Learning compact binary codes for visual tracking. In: CVPR (2013)
22. Li, X., Shen, C., Shi, Q., Dick, A., van den Hengel, A.: Non-sparse linear representations for visual tracking with online reservoir metric learning. In: CVPR (2012)
23. Li, Y., Ai, H., Yamashita, T., Lao, S., Kawade, M.: Tracking in low frame rate video: A cascade particle filter with discriminative observers of different lifespans. In: CVPR (2007)

24. Liu, B., Huang, J., Yang, L., Kulikowsk, C.: Robust tracking using local sparse appearance model and k-selection. In: CVPR (2011)
25. Mei, X., Ling, H.: Robust visual tracking and vehicle classification via sparse representation. Trans. on PAMI 33(11), 2259–2272 (2011)
26. Mei, X., Ling, H., Wu, Y., Blasch, E., Bai, L.: Minimum error bounded efficient ℓ_1 tracker with occlusion detection. In: CVPR (2011)
27. Oron, S., Bar-Hillel, A., Levi, D., Avidan, S.: Locally orderless tracking. In: CVPR (2012)
28. Pang, Y., Ling, H.: Finding the best from the second bests - inhibiting subjective bias in evaluation of visual tracking algorithms. In: ICCV (2013)
29. Ross, D.A., Lim, J., Lin, R., Yang, M.-H.: Incremental learning for robust visual tracking. Int. J. Comp. Vis. 77(1), 125–141 (2008)
30. Song, S., Xiao, J.: Tracking revisited using rgbd camera: Unified benchmark and baselines. In: ICCV (2013)
31. Supančič, J.S., Ramanan, D.: Self-paced learning for long-term tracking. In: CVPR (2013)
32. Wang, D., Lu, H., Yang, M.-H.: Least soft-thresold squares tracking. In: CVPR (2013)
33. Wang, N., Yeung, D.Y.: Learning a deep compact image representation for visual tracking. In: NIPS (2013)
34. Wang, Q., Chen, F., Yang, J., Xu, W., Yang, M.-H.: Transferring visual prior for online object tracking. Trans. on IP 21(7), 3296–3305 (2012)
35. Wu, Y., Lim, J., Yang, M.-H.: Online object tracking: A benchmark. In: CVPR (2013)
36. Xing, J., Gao, J., Li, B., Hu, W., Yan, S.: Robust object tracking with online multi-lifespan dictionary learning. In: ICCV (2013)
37. Yao, R., Shi, Q., Shen, C., Zhang, Y., van den Hengel, A.: Part-based visual tracking with online latent structural learning. In: CVPR (2013)
38. Yao, Y., Doretto, G.: Boosting for transfer learning with multiple sources. In: CVPR (2010)
39. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Comput. Surv. 38(4) (2006)
40. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: NIPS (2005)
41. Zhang, K., Zhang, L., Yang, M.-H.: Real-time compressive tracking. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 864–877. Springer, Heidelberg (2012)
42. Zhang, L., van der Maaten, L.: Structure preserving object tracking. In: CVPR (2013)
43. Zhang, T., Ghanem, B., Liu, S., Ahuja, N.: Low-rank sparse learning for robust visual tracking. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 470–484. Springer, Heidelberg (2012)
44. Zhang, T., Ghanem, B., Liu, S., Ahuja, N.: Robust visual tracking via multi-task sparse learning. In: CVPR (2012)
45. Zhong, W., Lu, H., Yang, M.-H.: Robust object tracking via sparsity-based collaborative model. In: CVPR (2012)
46. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions. In: ICML (2003)
47. Zhu, X., Lafferty, J., Ghahramani, Z.: Semi-supervised learning: From gaussian fields to gaussian processes. Tech. Rep. CMU-CS-03-175, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania (August 2003)