



cess is independently carried out frame by frame, and hence ignores the intrinsic cross-frame interaction information on temporal model coherence, leading to instable tracking results in complicated scenarios.

In this work, we propose a joint learning approach that is capable of well balancing the following three important parts: temporal model coherence across frames, spatial model consistency within frames, and discriminative feature construction. As illustrated in Figure 1, the joint learning approach ensures the temporal model coherence by building a multi-task structured model learning scheme, which encodes the cross-frame interaction information by simultaneously optimizing a set of mutually correlated learning subtasks (i.e., a common model plus different biases) over several successive frames. As a result, the interaction information induced by multi-task learning can guide the tracker to produce stable tracking results. Moreover, the proposed approach explores the keypoint-specific structural information on spatial model consistency by performing geometric verification based structured output learning, which aims to estimate a geometric transformation while associating cross-frame keypoints. In order to make the keypoint descriptors well adapt to time-varying tracking situations, the proposed approach naturally embeds metric learning to the structured SVM learning process, which enhances the discriminative power of inter-class separability.

In summary, we propose a keypoint tracking approach that learns an effective and robust keypoint model through metric learning-driven multi-task structured output optimization. The main contributions of this work are as follows:

1. We propose a multi-task joint learning scheme to learn structured keypoint models by simultaneously considering spatial model consistency, temporal model coherence, and discriminative feature learning. An online optimization algorithm is further presented to efficiently and effectively solve the proposed scheme. To our knowledge, it is the first time that such a joint learning scheme is proposed for learning-based keypoint tracking.
2. We create and release a new benchmark video dataset containing four challenging video sequences (covering several complicated scenarios) for experimental evaluations. In these video sequences, the keypoint tracking results are manually annotated as ground truth. Besides, the quantitative results on them are also provided in the experimental section.

## 2 Approach

Our tracking approach is mainly composed of two parts: learning part and prediction part. Namely, an object model is first learned by a multi-task structured learning scheme in a discriminative feature space (induced by metric learning). Based on the learned object model, our approach subsequently produces the tracking results through structured prediction. Using the tracking results, a set of training samples are further collected for structured learning. The above process is repeated as tracking proceeds.

### 2.1 Preliminary

Let the template image  $O$  be represented as a set of keypoints  $O = \{(u_i, \mathbf{q}_i)\}_{i=1}^{N^O}$ , where each keypoint is defined by a location  $u_i$  and associated descriptor  $\mathbf{q}_i$ . Similarly, let  $I = \{(v_j, \mathbf{d}_j)\}_{j=1}^{N^I}$  denote the input frame with keypoints. Typically, the traditional approaches construct the correspondences between the template keypoints and the input frame keypoints. The correspondences are scored by calculating the distances between  $\{\mathbf{q}_i\}_{i=1}^{N^O}$  and  $\{\mathbf{d}_j\}_{j=1}^{N^I}$ . Following the model learning approaches (Hare, Saffari, and Torr 2012; Stückler and Behnke 2012), we learn a model parameterized by a weight vector  $\mathbf{w}_i$  for the template keypoint  $u_i$  to score each correspondence. The set of the hypothetical correspondences is defined as  $C = \{(u_i, v_j, s_{ij}) | (u_i, \mathbf{q}_i) \in O, (v_j, \mathbf{d}_j) \in I, s_{ij} = \langle \mathbf{w}_i, \mathbf{d}_j \rangle\}$ , where  $s_{ij}$  is a correspondence score and  $\langle \cdot, \cdot \rangle$  is the inner product.

Similar to (Hare, Saffari, and Torr 2012; Pernici and Del Bimbo 2013), we estimate the homography transformation for planar object tracking as the tracking result based on the hypothetical correspondences.

### 2.2 Multi-task Structured Learning

During the tracking process, the keypoints in the successive frames  $\{I_1, I_2, \dots\}$  corresponding to the  $i$ -th keypoint  $u_i$  in the template image form a tracklet  $\{v^1, v^2, \dots\}$ . Based on the observation that the adjacent keypoints in a tracklet are similar to each other, the models learned for the frames  $\{\mathbf{w}_i^1, \mathbf{w}_i^2, \dots\}$  should be mutually correlated. So we construct  $K$  learning tasks over several adjacent frames. For example, task  $k$  learns a model  $\mathbf{w}^k$  over the training samples collected from the frames  $I_1$  to  $I_{t+k}$ , where  $\mathbf{w}^k = [\mathbf{w}_1^k, \dots, \mathbf{w}_{N^O}^k]^T$  is the column concatenation of the model parameter vectors. We model each  $\mathbf{w}^k$  as a linear combination of a common model  $\mathbf{w}^0$  and an unique part  $\mathbf{v}^k$  (Zheng and Ni 2013):

$$\mathbf{w}^k = \mathbf{w}^0 + \mathbf{v}^k, \quad k = 1, \dots, K \quad (1)$$

where all the vectors  $\{\mathbf{v}^k\}_{k=1}^K$  are “small” when the tasks are similar to each other.

To consider the spatial model consistency in the model learning process, the transformation which maps the template to the location of the input frame is regarded as a structure, which can be learned in a geometric verification based structured learning framework. In our approach, the expected transformation  $\hat{\mathbf{y}}$  is expressed as  $\hat{\mathbf{y}} = \arg \max_{\mathbf{y} \in \mathcal{Y}} F(C, \mathbf{y})$ , where  $F$  is a compatibility function, scoring all possible transformations generated by using the RANSAC (Fischler and Bolles 1981) method. Before introducing the compatibility function, we give the definition of the inlier set with a specific transformation  $\mathbf{y}$ :

$$H(C, \mathbf{y}) = \{(u_i, v_j) | (u_i, v_j) \in C, \|\mathbf{y}(u_i) - v_j\| < \tau\} \quad (2)$$

where  $\mathbf{y}(u_i)$  is the transformed location in the input frame of the template keypoint location  $u_i$ ,  $\tau \in \mathbb{R}$  is a spatial distance threshold, and  $\|\cdot\|$  denotes the Euclidean norm.

The compatibility function with respect to task  $k$  is then defined as the total score of the inliers:

$$F^k(C, \mathbf{y}) = \sum_{(u_i, v_j) \in H(C, \mathbf{y})} \langle \mathbf{w}_i^k, \mathbf{d}_j \rangle = \langle \mathbf{w}^k, \Phi(C, \mathbf{y}) \rangle \quad (3)$$

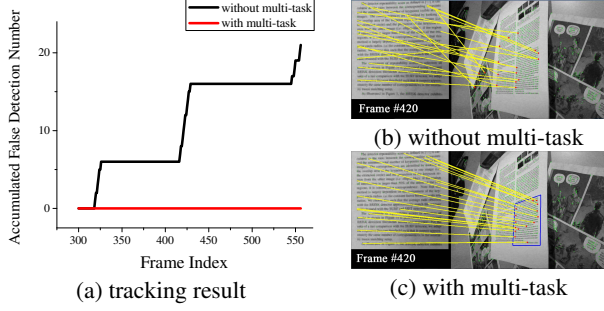


Figure 2: Example tracking results. Figure (a) shows the quantitative results of the trackers with and without multi-task learning in the accumulated number of falsely detected frames (lower is better), the tracker with multi-task learning produces a stable tracking result. Figure (b) and (c) show the qualitative tracking results. The blue bounding box represents the location of the detected object, and the yellow line represents a keypoint correspondence. In figure (b), the tracker without multi-task learning fails to match keypoints correctly.

where  $\Phi(C, \mathbf{y})$  is a joint feature mapping vector concatenated by  $\phi_i(C, \mathbf{y})$  which is defined as:

$$\phi_i(C, \mathbf{y}) = \begin{cases} \mathbf{d}_j & \exists (u_i, v_j) \in C : \|\mathbf{y}(u_i) - v_j\| < \tau \\ \mathbf{0} & \text{otherwise} \end{cases} \quad (4)$$

Given training samples  $\{(C_t, \mathbf{y}_t)\}_{t=1}^T$  (each  $C_t$  is the hypothetical correspondences of the frame  $I_t$ , and  $\mathbf{y}_t$  is the predicted transformation), a structured output maximum margin framework (Taskar, Guestrin, and Koller 2003; Tsochantaridis et al. 2005) is used to learn all the multi-task models, which can be expressed by the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w}^0, \mathbf{v}^k, \xi} & \frac{1}{2} \|\mathbf{w}^0\|^2 + \frac{\lambda_1}{2K} \sum_{k=1}^K \|\mathbf{v}^k\|^2 + \nu_1 \sum_{k=1}^K \sum_{t=k}^T \xi_{kt} \\ \text{s.t. } & \forall k, t, \xi_{kt} \geq 0 \\ & \forall k, t, \forall \mathbf{y} \neq \mathbf{y}_t : \delta F_t^k(\mathbf{y}) \geq \Delta(\mathbf{y}_t, \mathbf{y}) - \xi_{kt} \end{aligned} \quad (5)$$

where  $\delta F_t^k(\mathbf{y}) = \langle \mathbf{w}^k, \Phi(C_t, \mathbf{y}_t) \rangle - \langle \mathbf{w}^k, \Phi(C_t, \mathbf{y}) \rangle$  and  $\Delta(\mathbf{y}_t, \mathbf{y})$  is a loss function which measures the difference of two transformations (in our case, the loss function  $\Delta(\mathbf{y}_t, \mathbf{y}) = |\#H(C, \mathbf{y}_t) - \#H(C, \mathbf{y})|$  is the difference in number of two inlier sets). The nonnegative  $\lambda_1$  is the weight parameter for multiple tasks, and the weighting parameter  $\nu_1$  determines the trade-off between accuracy and regularization.

To better describe the contribution of the multi-task learning, example tracking results of the trackers with and without multi-task learning are shown in Figure 2. From Figure 2(b), we observe that the independent model fails to match the keypoints in the case of drastic rotations, while the multi-task model enables the temporal model coherence to capture the information of rotational changes, thus produces a stable tracking result.

### 2.3 Discriminative Feature Space

In order to make the keypoint descriptors well adapt to time-varying tracking situations, we wish to learn a mapping

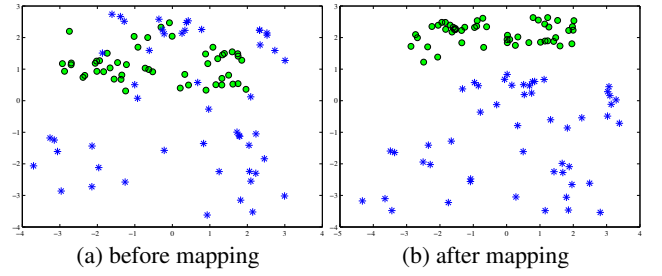


Figure 3: Visualization of keypoint features using PCA. The 50 green circle points represent the keypoints from 50 successive frames corresponding to the same keypoint in the template (semantically similar keypoints). The blue asterisk points represent any other keypoints, which are dissimilar to the above 50 keypoints. In figure (a), all the keypoints mix together in the original feature space. In figure (b), there is a large margin between dissimilar keypoints in current learned feature space.

function  $f(\mathbf{d})$  that maps the original feature space to another discriminative feature space, in which the semantically similar keypoints are close to each other while the dissimilar keypoints are far away from each other, that can be formulated as a metric learning process (Weinberger and Saul 2009; Park et al. 2011). We then use the mapped feature  $f(\mathbf{d})$  to replace the original feature  $\mathbf{d}$  in the structured learning process, to enhance its discriminative power of inter-class separability.

Figure 3 shows an example of such feature space transformation. Before the mapping procedure, the object keypoints and the background keypoints can not be discriminated in the original feature space. After the transformation, the keypoints in different frames corresponding to the same keypoint in the template, which are semantically similar, get close to each other in the mapped feature space, while the features of the other keypoints have a distribution in another side with a large margin.

The following describes how to learn the mapping function. For a particular task  $k$ , given the learned model  $\mathbf{w}_i^k$ , the distance between a doublet  $(\mathbf{d}_j, \mathbf{d}_{j'})$  is defined as follows:

$$D_i^k(\mathbf{d}_j, \mathbf{d}_{j'}) = \langle \mathbf{w}_i^k, f(\mathbf{d}_j) - f(\mathbf{d}_{j'}) \rangle \quad (6)$$

We assume that the binary matrix  $p_{jj'} \in \{0, 1\}$  indicates whether or not the features  $\mathbf{d}_j$  and  $\mathbf{d}_{j'}$  are semantically similar (if they are similar,  $p_{jj'} = 1$ ). Therefore, the hinge loss function on a doublet is defined as:

$$\ell_i^k(\mathbf{d}_j, \mathbf{d}_{j'}) = [(-1)^{p_{jj'}} (1 - D_i^k(\mathbf{d}_j, \mathbf{d}_{j'}))]_{+} \quad (7)$$

where  $[z]_{+} = \max(z, 0)$ .

To learn the effective feature consistently in our mapping process, we wish to find the group-sparsity of the features. So we utilize  $\ell_{2,1}$ -norm (Cai et al. 2011; Li et al. 2012) to learn the discriminative information and feature correlation consistently. Since we use a linear transformation  $f(\mathbf{d}) = \mathbf{M}^T \mathbf{d}$  as our mapping function, the  $\ell_{2,1}$ -norm for the mapping matrix  $\mathbf{M}$  is defined as:  $\|\mathbf{M}\|_{2,1} = \sum_i \sqrt{\sum_j \mathbf{M}_{ij}^2}$ .

Given all the keypoint features from the video frames  $\{I_t\}_{t=1}^T$ , we collect all possible combinations of the features

as the training set, which is denoted as  $\mathcal{A} = \{(\mathbf{d}_j, \mathbf{d}_{j'}) | \mathbf{d}_j \in \{I_t\}_{t=1}^T, j' \neq j, \mathbf{d}_{j'} \in \{I_t\}_{t=1}^T\}$ . We obtain the binary matrix  $p_{jj'}$  by using the tracking results (if  $\mathbf{d}_j$  and  $\mathbf{d}_{j'}$  from different frames correspond to the same keypoint in the template,  $p_{jj'}$  is set to 1; otherwise,  $p_{jj'}$  is set to 0). We wish to minimize the following cost function consisting of the empirical loss term and the  $\ell_{2,1}$ -norm regularization term:

$$\sum_{i,k,(\mathbf{d}_j, \mathbf{d}_{j'}) \in \mathcal{A}} \ell_i^k(\mathbf{d}_j, \mathbf{d}_{j'}) + \lambda \|\mathbf{M}\|_{2,1} \quad (8)$$

The cost function is incorporated into our multi-task structured learning framework, and then a unified joint learning scheme for object tracking is obtained. The final optimization problem of our approach is expressed in the following form:

$$\begin{aligned} \min_{\mathbf{w}^0, \mathbf{v}^k, \mathbf{M}, \xi, \gamma} & \frac{1}{2} \|\mathbf{w}^0\|^2 + \frac{\lambda_1}{2K} \sum_{k=1}^K \|\mathbf{v}^k\|^2 + \lambda_2 \|\mathbf{M}\|_{2,1} \\ & + \sum_{k=1}^K \left( \nu_1 \sum_{t=k}^T \xi_{kt} + \nu_2 \sum_{t=k}^T \sum_{(u_i, v_j) \in H(C_t, \mathbf{y}_t)} \gamma_{kti} \right) \\ \text{s.t. } & \forall k, t, \xi_{kt} \geq 0 \\ & \forall k, t, \forall \mathbf{y} \neq \mathbf{y}_t : \delta F_t^k(\mathbf{y}) \geq \Delta(\mathbf{y}_t, \mathbf{y}) - \xi_{kt} \\ & \forall k, t, i : \gamma_{kti} \geq 0 \\ & \forall k, t, (u_i, v_j), \forall j' \neq j : D_i^k(\mathbf{d}_j, \mathbf{d}_{j'}) \geq 1 - \gamma_{kti} \end{aligned} \quad (9)$$

After all the models  $\mathbf{w}^1, \mathbf{w}^2, \dots, \mathbf{w}^K$  are learned, we use the last model  $\mathbf{w} = \mathbf{w}^K$  to predict the result of new frame  $I_t$ . We use the RANSAC method to generate hypothetical transformations. Based on the model  $\mathbf{w}$ , we predict the expected transformation  $\mathbf{y}_t$  from all hypothetical transformations by maximizing Eq. (3). The hypothetical correspondence set  $\hat{C}_t$  of the frame  $I_t$  and the predicted transformation  $\mathbf{y}_t$  are then added to our training set. We use all the training samples collected from the results of previous  $K$  frames ( $I_{t-K+1}$  to  $I_t$ ) to update our model. Then the above process is repeated as tracking proceeds.

## 2.4 Online Optimization

The optimization problem presented in Eq. (9) can be solved online effectively. We adopt an alternating optimization algorithm to solve the optimization problem.

**Unconstrained form** Let  $\alpha_{kt} = [\max_{\mathbf{y} \neq \mathbf{y}_t} \{\Delta(\mathbf{y}_t, \mathbf{y}) - \delta F_t^k(\mathbf{y})\}]_+$  and  $\beta_{kti} = [\max_{j' \neq j} \{1 - D_i^k(\mathbf{d}_j, \mathbf{d}_{j'})\}]_+$ . Therefore, Eq. (9) can be rewritten to an unconstrained form:

$$\begin{aligned} \min_{\mathbf{w}^0, \mathbf{v}^k, \mathbf{M}} & \frac{1}{2} \|\mathbf{w}^0\|^2 + \frac{\lambda_1}{2K} \sum_{k=1}^K \|\mathbf{v}^k\|^2 + \lambda_2 \|\mathbf{M}\|_{2,1} \\ & + \sum_{k=1}^K \left( \nu_1 \sum_{t=k}^T \alpha_{kt} + \nu_2 \sum_{t=k}^T \sum_{(u_i, v_j) \in H(C_t, \mathbf{y}_t)} \beta_{kti} \right) \end{aligned} \quad (10)$$

For descriptive convenience, let  $J$  denote the term of  $\nu_1 \sum_{t=k}^T \alpha_{kt} + \nu_2 \sum_{t=k}^T \sum_{(u_i, v_j) \in H(C_t, \mathbf{y}_t)} \beta_{kti}$ .

**Fix  $\{\mathbf{v}^k\}_{k=1}^K$  and  $\mathbf{w}^0$ , solve  $\mathbf{M}$**  Firstly, we fix all  $\{\mathbf{v}^k\}_{k=1}^K$  and  $\mathbf{w}^0$ , and learn the transformation matrix  $\mathbf{M}$  by solving the following problem:

$$\min_{\mathbf{M}} \|\mathbf{M}\|_{2,1} + \frac{1}{\lambda_2} \sum_{k=1}^K J \quad (11)$$

Let  $\mathbf{M}^i$  denote the  $i$ -th row of  $\mathbf{M}$ , and  $Tr(\cdot)$  denote the trace operator. In mathematics, the Eq. (11) can be converted to the following form:

$$\min_{\mathbf{M}} Tr(\mathbf{M}^T \mathbf{D} \mathbf{M}) + \frac{1}{\lambda_2} \sum_{k=1}^K J \quad (12)$$

where  $\mathbf{D}$  is the diagonal matrix of  $\mathbf{M}$ , and each diagonal element is  $\mathbf{D}_{ii} = \frac{1}{2\|\mathbf{M}^i\|_2}$ . We use an alternating algorithm to calculate  $\mathbf{D}$  and  $\mathbf{M}$  respectively. We calculate  $\mathbf{M}$  with the current  $\mathbf{D}$  by using gradient descent method, and then update  $\mathbf{D}$  according to the current  $\mathbf{M}$ . The details of solving Eq. (12) are shown in the supplementary file.

**Fix  $\mathbf{M}$  and  $\{\mathbf{v}^k\}_{k=1}^K$ , solve  $\mathbf{w}^0$**  Secondly, after  $\mathbf{M}$  is learned, let  $\{\mathbf{v}^k\}_{k=1}^K$  have been the optimal solution of Eq. (10). Then  $\mathbf{w}^0$  can be obtained by the combination of  $\mathbf{v}^k$  according to (Evgeniou and Pontil 2004):

$$\mathbf{w}^0 = \frac{\lambda_1}{K} \sum_{k=1}^K \mathbf{v}^k \quad (13)$$

The proof can be found in our supplementary material.

**Fix  $\mathbf{M}$  and  $\mathbf{w}^0$ , solve  $\{\mathbf{v}^k\}_{k=1}^K$**  Finally,  $\{\mathbf{v}^k\}_{k=1}^K$  can be learned one by one using gradient descent method. In fact, we learn  $\mathbf{w}^k = \mathbf{w}^0 + \mathbf{v}^k$  instead of  $\mathbf{v}^k$  for convenience. Let  $\bar{\mathbf{w}} = \frac{1}{K} \sum_{k=1}^K \mathbf{w}^k$  be the average vector of all  $\mathbf{w}^k$ . Then the optimization problem for each  $\mathbf{w}^k$  can be rewritten as:

$$\min_{\mathbf{w}^k} \rho_1 \|\mathbf{w}^k\|^2 + \rho_2 \|\mathbf{w}^k - \bar{\mathbf{w}}\|^2 + J \quad (14)$$

where  $\rho_1 = \lambda_1/(\lambda_1 + 1)$  and  $\rho_2 = \lambda_1^2/(\lambda_1 + 1)$  (the derivation proof is given in the supplementary material).

Given training samples  $\{(C_{t-k}, \mathbf{y}_{t-k})\}_{k=0}^{K-1}$  at time  $t$ , the subgradient of Eq. (14) with respect to  $\mathbf{w}^k$  is calculated, and we perform a gradient descent step according to:

$$\mathbf{w}^k \leftarrow (1 - \frac{1}{t}) \mathbf{w}^k + \eta \rho_2 \bar{\mathbf{w}} - \eta \frac{\partial J}{\partial \mathbf{w}^k} \quad (15)$$

where  $\eta = 1/(\rho_1 t + \rho_2 t)$  is the step size (the details of the term  $J$  is described in the supplementary material). We repeat the procedure to obtain an optimal solution until the algorithm converges (on average converges after 5 iterations).

All the above is summarized in Algorithm 1, and the details are described in the supplementary material.

## 3 Experiments and Results

### 3.1 Experimental Settings

**Dataset** The video dataset used in our experiments consists of nine video sequences. Specifically, the first five sequences are from (Hare, Saffari, and Torr 2012), and the last

**Algorithm 1:** Online Optimization for Tracking

---

**Input:** Input frame  $I_t$  and previous models  $\{\mathbf{w}^1, \dots, \mathbf{w}^K\}$   
**Output:** The predicted transformation  $\mathbf{y}_t$ , updated models and mapping matrix for metric learning

---

```

/* The structured prediction part */
1 Calculate the correspondences  $C_t$  based on the model  $\mathbf{w}^K$ ;
2 Estimate hypothetical transformations  $\mathbf{y}$  using RANSAC;
  Calculate the inlier set of each  $\mathbf{y}$  using Eq. (2);
3 Predict the expected  $\mathbf{y}_t$  by maximizing Eq. (3);
/* The structured learning part */
4 Collect the training samples  $\{(C_{t-k}, \mathbf{y}_{t-k})\}_{k=0}^{K-1}$ ;
5 repeat
6   Calculate  $J$  according to Section 2.4;
7   for  $k = 1, \dots, K$  do
8     Update each model  $\mathbf{w}^k$  using Eq. (15);
9   end
10  Update the mapping matrix  $\mathbf{M}$  by solving Eq. (12);
11 until Alternating optimization convergence;
12 return  $\mathbf{y}_t, \{\mathbf{w}^1, \dots, \mathbf{w}^K\}$  and  $\mathbf{M}$ ;

```

---

four sequences (i.e., “chart”, “keyboard”, “food”, “book”) are recorded by ourselves. All these sequences cover several complicated scenarios such as background clutter, object zooming, object rotation, illumination variation, motion blurring and partial occlusion (example frames can be found in the supplementary material).

**Implementation Details** For keypoint feature extraction, we use FAST keypoint detector (Rosten and Drummond 2006) with 256-bit BRIEF descriptor (Calonder et al. 2010). For metric learning, the linear transformation matrix  $\mathbf{M}$  is initialized to be an identity matrix. For multi-task learning, the number of tasks  $K$  is chosen as 5 and we update all the multi-task models frame by frame. All weighting parameters  $\lambda_1, \lambda_2, \nu_1, \nu_2$  are set to 1, and remain fixed throughout all the experiments. Similar to (Hare, Saffari, and Torr 2012), we consider the tracking process of estimating homography transformation on the planar object as a tracking-by-detection task.

**Evaluation Criteria** We use the same criteria as (Hare, Saffari, and Torr 2012) with a scoring function between the predicted homography  $\mathbf{y}$  and the ground-truth homography  $\mathbf{y}^*$ :

$$S(\mathbf{y}, \mathbf{y}^*) = \frac{1}{4} \sum_{i=1}^4 \|\mathbf{y}(c_i) - \mathbf{y}^*(c_i)\|_2 \quad (16)$$

where  $\{c_i\}_{i=1}^4 = \{(-1, -1)^\top, (1, -1)^\top, (-1, 1)^\top, (1, 1)^\top\}$  is a normalized square. For each frame, it is regarded as a successfully detected frame if  $S(\mathbf{y}, \mathbf{y}^*) < 10$ , and a falsely detected frame otherwise. The average success rate is defined as the number of successfully detected frames divided by the length of the sequence, which is used to evaluate the performance of the tracker. To provide the tracking result frame by frame, we present a criterion of the accumulated false detection number, which is defined as the accumulated number of falsely detected frames as tracking proceeds.

Sequence	Average Success Rate(%)			
	Static	Boosting	SSVM	Ours
barbapapa	19.7138	89.0302	94.1176	<b>94.4356</b>
comic	42.5000	57.6042	98.1250	<b>98.8542</b>
map	81.1295	82.0937	<b>98.7603</b>	<b>98.7603</b>
paper	05.0267	03.8502	82.7807	<b>88.2353</b>
phone	88.1491	84.9534	96.6711	<b>98.4021</b>
chart	13.1461	01.9101	53.0337	<b>77.5281</b>
keyboard	27.8607	57.7114	62.3549	<b>94.5274</b>
food	32.8173	67.4923	85.7585	<b>99.6904</b>
book	08.5616	08.9041	55.8219	<b>81.6781</b>

Table 1: Comparison with state-of-the-art approaches in the average success rate (higher is better). The best result on each sequence is shown in bold font. We observe that our approach performs best on all the sequences.

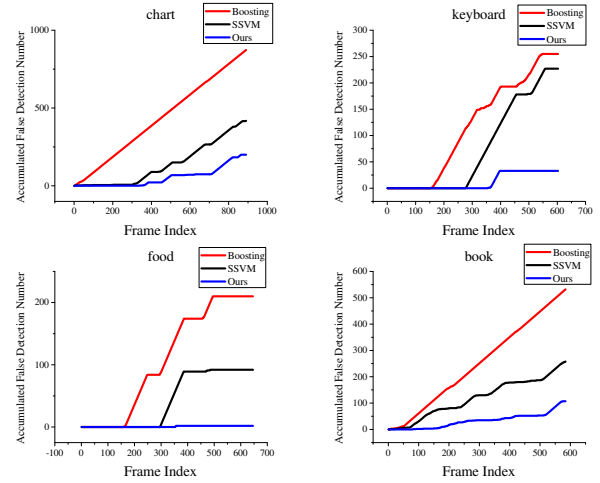


Figure 4: Comparison of three approaches in the accumulated number of falsely detected frames (lower is better). The curve corresponding to our approach grows slowly and is almost horizontal, which means that our tracking result is stable.

### 3.2 Experimental Results

**Comparison with State-of-the-art Methods** We compare our approach with some state-of-the-art approaches, including boosting based approach (Grabner, Grabner, and Bischof 2007), structured SVM (SSVM) approach (Hare, Saffari, and Torr 2012) and a baseline static tracking approach (without model updating). All these approaches are implemented by making use of their publicly available code. We also implement our approach in C++ and OPENCV. On average, our algorithm takes 0.0746 second to process one frame with a quad-core 2.4GHz Intel Xeon E5-2609 CPU and 16GB memory. Table 1 shows the experimental results of all four approaches in the average success rate. As shown in this table, our approach performs best on all sequences.

To provide an intuitive illustration, we report the detection result on each frame in Figure 4. We observe that both the “Boosting” and “SSVM” approaches obtain a number of incorrect detection results on some frames of the test sequences, while our approach achieves stable tracking results in most situations (the curve corresponding to our approach grows slowly and is almost horizontal).



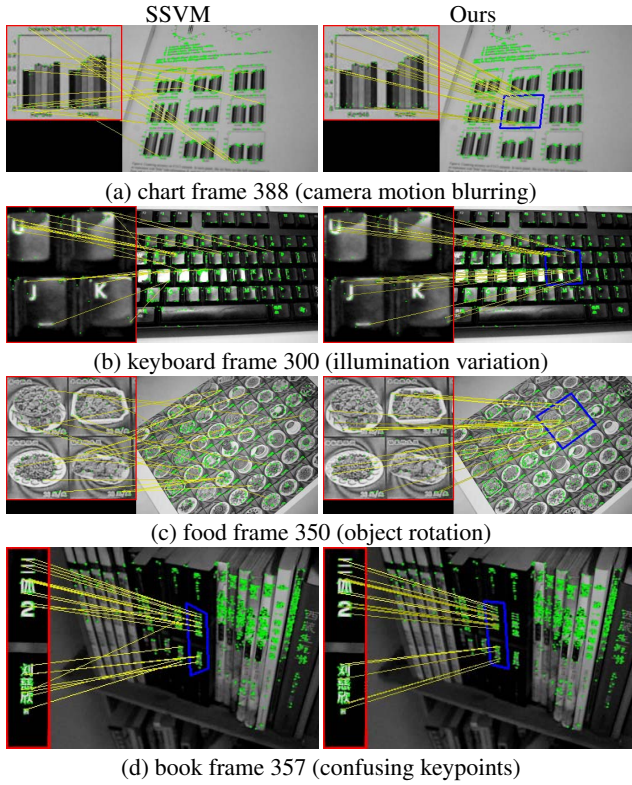


Figure 5: Example tracking results on our test video sequences. In each picture, the left part highlighted in red bounding box is the template image. The blue box shows the location of the detected object in the frame. Our model has adapted to obtain correct detection results in the complicated scenarios with drastic object appearance changes.

Figure 5 shows the tracking results on some sample frames (more experimental results can be found in our supplementary materials). These sequences containing background clutter are challenging for keypoint based tracking. In terms of metric learning and multi-task learning, our approach still performs well in some complicated scenarios with drastic object appearance changes.

**Evaluation of Our Individual Components** To explore the contribution of each component in our approach, we compare the performances of the approaches with individual parts, including SSVM(structured SVM), SML(SSVM + metric learning), SMT(SSVM + multi-task learning), and SMM (SSVM + ML + MT, which is exactly our approach). The experimental results of all these approaches in the average success rate are reported in Table 2.

From Table 2, we find that the geometric verification based structured learning approach achieves good tracking results in most situations. Furthermore, we observe from Figure 6 that multi-task structured learning guides the tracker to produce a stable tracking result in the complicated scenarios, and metric learning enhances the capability of the tracker to separate keypoints from background clutter. Our approach consisting of all these components then generates a robust tracker.

Sequence	Average Success Rate(%)			
	SSVM	SML	SMT	SMM
barbapapa	94.1176	<b>94.4356</b>	94.2766	<b>94.4356</b>
comic	98.1250	98.5417	98.6458	<b>98.8542</b>
map	<b>98.7603</b>	98.6226	<b>98.7603</b>	<b>98.7603</b>
paper	82.7807	86.2032	87.3797	<b>88.2353</b>
phone	96.6711	97.2037	97.6032	<b>98.4021</b>
chart	53.0337	62.0225	61.1236	<b>77.5281</b>
keyboard	62.3549	73.6318	76.6169	<b>94.5274</b>
food	85.7585	88.0805	99.3808	<b>99.6904</b>
book	55.8219	71.5753	74.8288	<b>81.6781</b>

Table 2: Evaluation of our individual components in the average success rate (higher is better). The best result on each sequence is shown in bold font. We find that both metric learning and multi-task learning based approach obtain a higher success rate than the structured SVM approach, and our joint learning approach achieves the best performance.

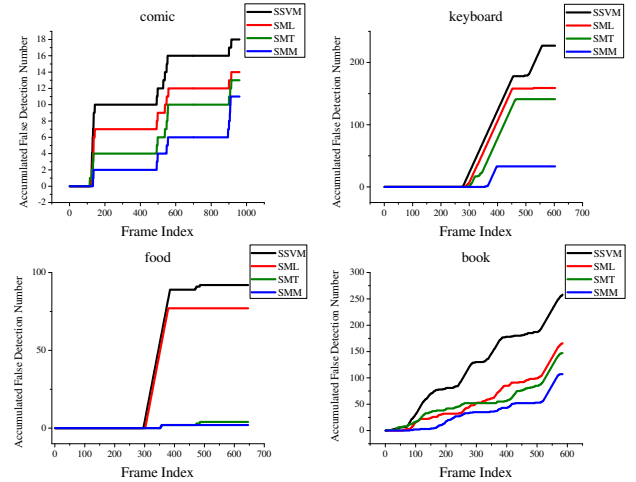


Figure 6: Evaluation of our individual components in the accumulated number of falsely detected frames (lower is better). We observe that both metric learning and multi-task learning can improve the robustness of the tracker.

## 4 Conclusion

In this paper, we have presented a novel and robust keypoint tracker by solving a multi-task structured output optimization problem driven by metric learning. Our joint learning approach have simultaneously considered spatial model consistency, temporal model coherence, and discriminative feature construction during the tracking process.

We have shown in extensive experiments that geometric verification based structured learning has modeled the spatial model consistency to generate a robust tracker in most scenarios, multi-task structured learning has characterized the temporal model coherence to produce stable tracking results even in complicated scenarios with drastic changes, and metric learning has enabled the discriminative feature construction to enhance the discriminative power of the tracker. We have created a new benchmark video dataset consisting of challenging video sequences, and experimental results performed on the dataset have shown that our tracker outperforms the other state-of-the-art trackers.

## 5 Acknowledgments

All correspondence should be addressed to Prof. Xi Li. This work is in part supported by the National Natural Science Foundation of China (Grant No. 61472353), National Basic Research Program of China (2012CB316400), NSFC (61472353), 863 program (2012AA012505), China Knowledge Centre for Engineering Sciences and Technology and the Fundamental Research Funds for the Central Universities.

## References

- Alahi, A.; Ortiz, R.; and Vanderghenst, P. 2012. Freak: Fast retina keypoint. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Bay, H.; Ess, A.; Tuytelaars, T.; and Gool, L. V. 2008. Speeded-up robust features (surf). *Computer Vision and Image Understanding (CVIU)* 110(3):346–359.
- Bouachir, W., and Bilodeau, G.-A. 2014. Structure-aware keypoint tracking for partial occlusion handling. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*.
- Cai, X.; Nie, F.; Huang, H.; and Ding, C. 2011. Multi-class  $l_2, l_1$ -norm support vector machine. In *Proceedings of the IEEE Conference on Data Mining (ICDM)*.
- Calonder, M.; Lepetit, V.; Strecha, C.; and Fua, P. 2010. Brief: Binary robust independent elementary features. In *Proceedings of the European Conference on Computer Vision (ECCV)*.
- Evgeniou, T., and Pontil, M. 2004. Regularized multi-task learning. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*.
- Fischler, M. A., and Bolles, R. C. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM (CACM)* 24(6):381–395.
- Grabner, M.; Grabner, H.; and Bischof, H. 2007. Learning features for tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Guo, B., and Liu, J. 2013. Real-time keypoint-based object tracking via online learning. In *Proceedings of the International Conference on Information Science and Technology (ICIST)*.
- Hare, S.; Saffari, A.; and Torr, P. H. S. 2012. Efficient online structured output learning for keypoint-based object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Lepetit, V., and Fua, P. 2006. Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 28(9):1465–1479.
- Leutenegger, S.; Chli, M.; and Siegwart, R. 2011. Brisk: Binary robust invariant scalable keypoints. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.
- Li, Z.; Yang, Y.; Liu, J.; Zhou, X.; and Lu, H. 2012. Unsupervised feature selection using nonnegative spectral analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.
- Li, X.; Hu, W.; Shen, C.; Zhang, Z.; Dick, A.; and Hengel, A. V. D. 2013. A survey of appearance models in visual object tracking. *ACM Transactions on Intelligent Systems and Technology (TIST)* 4(4):58:1–58:48.
- Lowe, D. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)* 60(2):91–110.
- Lucas, B. D., and Kanade, T. 1981. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.
- Maresca, M., and Petrosino, A. 2013. Matrioska: A multi-level approach to fast tracking by learning. In *Proceedings of the International Conference on Image Analysis and Processing (ICIAP)*.
- Mikolajczyk, K., and Schmid, C. 2005. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 27(10):1615–1630.
- Nebehay, G., and Pflugfelder, R. 2014. Consensus-based matching and tracking of keypoints for object tracking. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*.
- Özuysal, M.; Calonder, M.; Lepetit, V.; and Fua, P. 2010. Fast keypoint recognition using random ferns. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 32(3):448–461.
- Park, K.; Shen, C.; Hao, Z.; and Kim, J. 2011. Efficiently learning a distance metric for large margin nearest neighbor classification. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.
- Pernici, F., and Del Bimbo, A. 2013. Object tracking by oversampling local features. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* PP(99):1–1.
- Rosten, E., and Drummond, T. 2006. Machine learning for high-speed corner detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*.
- Rublee, E.; Rabaud, V.; Konolige, K.; and Bradski, G. 2011. Orb: An efficient alternative to sift or surf. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.
- Santner, J.; Leistner, C.; Saffari, A.; Pock, T.; and Bischof, H. 2010. Prost: Parallel robust online simple tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Stückler, J., and Behnke, S. 2012. Model learning and real-time tracking using multi-resolution surfel maps. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.
- Taskar, B.; Guestrin, C.; and Koller, D. 2003. Max-margin markov networks. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*.
- Tsochantaridis, I.; Joachims, T.; Hofmann, T.; and Altun, Y. 2005. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research (JMLR)* 6:1453–1484.
- Weinberger, K. Q., and Saul, L. K. 2009. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research (JMLR)* 10:207–244.
- Zheng, J., and Ni, L. M. 2013. Time-dependent trajectory regression on road networks via multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.