Abstract

Matching visual appearances of target sample reservoir over consecutive image frames is the most critical issue in sequence-based object tracking. Recent literatures show the effectiveness of the utilization of local feature points set instead of any global feature vectors of patches. A traditional tracking-by-detection framework without taking advantages of geometric information, however, ignores more or less the potential contributions of feature points. This paper proposes a totally novel tracking-by-correspondences framework, a generative approach via an adaptively-selected robust appearance model, a one-step orient motion model based on points correspondences, an automatic scale determination and a clustered online updating target sample reservoir. Extensive experiments validate the accuracy and robustness of the proposed method, and demonstrate the improved performance has been competitive enough to surpass the state of this art.

Key words*: tracking-by-correspondences, robust feature points, maximal weight clique, common pattern discovery, graph clustering*

摘要

对样本池中的目标物体样本与连续视频帧的视觉特征进行匹配是跟踪领域的关键课题。近几年的文献热衷于用局部特征点代替图像块的全局特征向量进行跟踪，并展现了这种做法的有效性。传统的基于检测的跟踪框架没有利用图像的空间信息，这样或多或少地忽略了特征点的潜在贡献。本文提出了一种全新的基于点对匹配的跟踪框架，这是一种综合了自适应稳定特征筛选模型、一步有向运动模型、自动尺度调整模型、在线样本池聚类更新模型的普适性方法。本文实验验证了该方法的精确性和鲁棒性，并且展现了我们对跟踪性能的改进足以超越世界领先水平。

关键词：*基于点对匹配的跟踪，稳定特征点，最大带权子图，共模搜索，图聚类*

Contents

1. Introduction and related work

Visual object tracking is one of the cardinal problems of visual understanding in general, and is crucial to many computer vision applications ranging from robotics, surveillance, augmented reality to human-computer interaction. The state of this art has advanced significantly in the past 30 years [1–12], from shape-based methods, probabilistic models using mean-shift [4], global template-based trackers [8] or particle filtering [10] to local point based trackers [12]. However, it is still far from achieving results comparable to human performance due to several challenges such like shape deformation, occlusion, heavy appearance changes, heavy illumination changes, motion blur and background clutter.

Current tracking algorithms mentioned above can be categorized into either discriminative or generative approaches. Discriminative methods model tracking as a classification problem which aims to distinguish the target from the backgrounds. It employs the information from both the target as positive samples and the backgrounds as negative ones. Jiang *et al*. [13] proposed metric learning to improve distance measure by minimizing the misclassification rate. Babenko *et al*. [14] adapted Multiple Instance Learning (MIL) by building an evolving boosting classifier that tracks bags of features. Generative methods formulate the tracking problem as searching for the patch most similar to the target. Ross *et al*. [15] utilized low-dimensional subspace representation, learnt incrementally during tracking process, to adapt target appearance changes. Zhou *et al*. [16] tried to integrate multiple similarity measures with a diffusion process on their Tensor Product Graph (TPG), resulting in a beneficial fusion of different similarity metrics that focus on different visual representations.

Recently, tracking approaches based on detection systems [5, 6, 8, 9, 14, 17, 18] have become popular since these systems have fast and robust performance thanks to their simple discrimination between target object and its surrounding backgrounds. Gu *et al*. [19] combined this NN-classifier-based tracking-by-detection framework with Efficient Subwindow Search (ESS) as the motion model. In terms of his appearance model, his feature points were updated and pruned in a bounded hyper ball in object feature space, so as to achieve a

proper balance between plasticity and stability. Although this method works well in some cases, several limitations have been exposed. First, feature points are so local that the similarity function loses global information when merely the number of positive points is contributed to it. It seems that slight drifting will accumulate failure of tracking. Second, all feature points without robustness filtering are classified by a simple classifier, which easily misleads the tracker once several background points resemble the positive ones, especially when they are in the initial frame. Third, while the motion model without advanced filtering techniques speeds up the algorithm, a brute search, anyhow, is more naive and more time-consuming than an oriented search strategy when the geometric information of feature points are involved.

In this paper, we are motivated by Gu *et al*. [19], utilizing robust feature points to represent target and to located it in the subsequent frames. Further, different from methods in literatures, we dispose of above-mentioned limitations by a totally innovative framework. The proposed approach achieves better performance of locating target and has an automatic deterministic scale selection instead of violently sampling different scale factors in scale space. It keeps effective when almost any challenging problems occur. Its robustness and universality are validated by several convincing experiments.

The novel contributions of the proposed method include the following four aspects.

1. It finds maximum weight cliques in a weighted adjacency graph with mutex constraints so as to obtain part of the most robust feature points that occur in all several frames in which target has similar appearances. Such strategy overcomes the confusion from occlusion and background clutter, as well as lowers time complexity.

2. It takes advantages of both the feature information and geometric information of robust feature points to conduct common pattern discovery via spatial coherent correspondences, thus better matching candidates frame by frame without discrimination between positive samples and negative ones.

3. Feature points correspondences between target sample and candidate can directly orient the drift of target in the subsequent frame and determine its scale. No advanced

motion model is introduced here, but both the position and scale are more accurate without any optimization and iteration.

4. Normalized cut is utilized to cluster new tracked target with older target samples in previous frames. The number of clusters guarantees the diversity/plasticity of target with heavy appearance changes while the volume of a cluster decide the robustness/stability of feature points filtered out by the maximum weight cliques. A restriction to the volume limits the space complexity of target sample reservoir and no negative background samples are demanded.

In summary, the proposed method belongs to a generative one with an adaptive robust appearance model, a one-step orient motion model based on point correspondences, and a clustered online updating target sample reservoir. To the best of our knowledge, this is the first time that visual object tracking is formulated as a so-called tracking-by-correspondences framework and reaches competitive results.

2. Problem formulation

We maintain a target sample reservoir $\mathbb{T} = \{T_{ij}\}, i = 1, \ldots, m, j \in \mathbf{N}^+, j \leq p$, in which m clusters contain no more than p target samples with similar appearances. Each sample is a previous target object state, collected during the sequence going. Thus, the reservoir is updated online frame by frame without aforehand training. In terms of a target sample T_{ij}, $\mathbb{P}_{ij} = \{p_{ijk}\}, k \in \mathbf{N}^+$ are its feature points, consisting of two parts, $p_{ijk} = \{p_f, p_c\}$. p_f represent local features. We use SIFT [20] here, although other more recent methods such as SURF [21], Self-Similarity [22] or Critical Nets [23] could be used as well. $p_c = \{x, y\}$ are the coordinates of the feature points.

Based on maximal weight cliques [24], we select part of the most robust feature points $\hat{\mathbb{P}}_{ij} \subseteq \mathbb{P}_{ij}$ from each target sample, aiming at filtering those noise points from occlusion and background clutter, as well as speeding up our following steps.

After locating the target T_{t-1} in the last frame I_{t-1}, we initialize the location of the first candidate C_t^1 in I_t at the same place, $c(C_t^1) = c(T_{t-1})$. Then for robust feature points $\hat{\mathbb{P}}_{ij}$ from each target sample, we discover common visual pattern via spatial coherent correspondences [25] between $\hat{\mathbb{P}}_{ij}$ and $\mathbb{Q}_t^1$, the feature points of C_t^1. We define all possible correspondences set as $\mathbb{COMRR} = \{CORR_n\} = \{(p_n, q_n)\}, n \in \mathbf{N}^+, n \leq |\hat{\mathbb{P}}_{ij}||\mathbb{Q}_t^1|, p_n \in \hat{\mathbb{P}}_{ij}, q_n \in \mathbb{Q}_t^1$. The geometric drift $\triangle c_k$ of each pair of correct correspondence $\mathbb{COMRR}^* = \{CORR_k^*\} = \{(p_k^*, q_k^*)\} \subseteq \mathbb{COMRR}, k \in \mathbf{N}^+, k \leq |\hat{\mathbb{P}}_{ij}|$ contributes to the mean drift $\overline{\triangle c}$ of the candidate according to its normalized similarity score w_k, namely, $\overline{\triangle c} = \sum_k w_k \triangle c_k$. We iteratively conduct such discovery process to obtain another candidate until the mean drift is zero where $c(C_t^{s+1}) = c(T_{ij}) + \overline{\triangle c}$, and at most cases, this process experiences only one iteration because of the slight displacement of target in tracking problem.

Once $T_t^{ij} = C_t^{s+1} = C_t^s$ is obtained, we select $T_t = \arg_{T_t^{ij}}\{\max g(\mathbf{x}^{*ij})\}$ as the final tracking result to add it to the reservoir $\mathbb{T}$, where $g(\cdot)$ is *the average intra-cluster affinity score* and $\mathbf{x}^{ij*}$ is the largest maxima as mentioned in the following detailed subsection. A new affinity graph $\mathbf{G}_\mathbb{T}$ of $\mathbb{T}$ is constructed based on the older one, the vertexes of which are

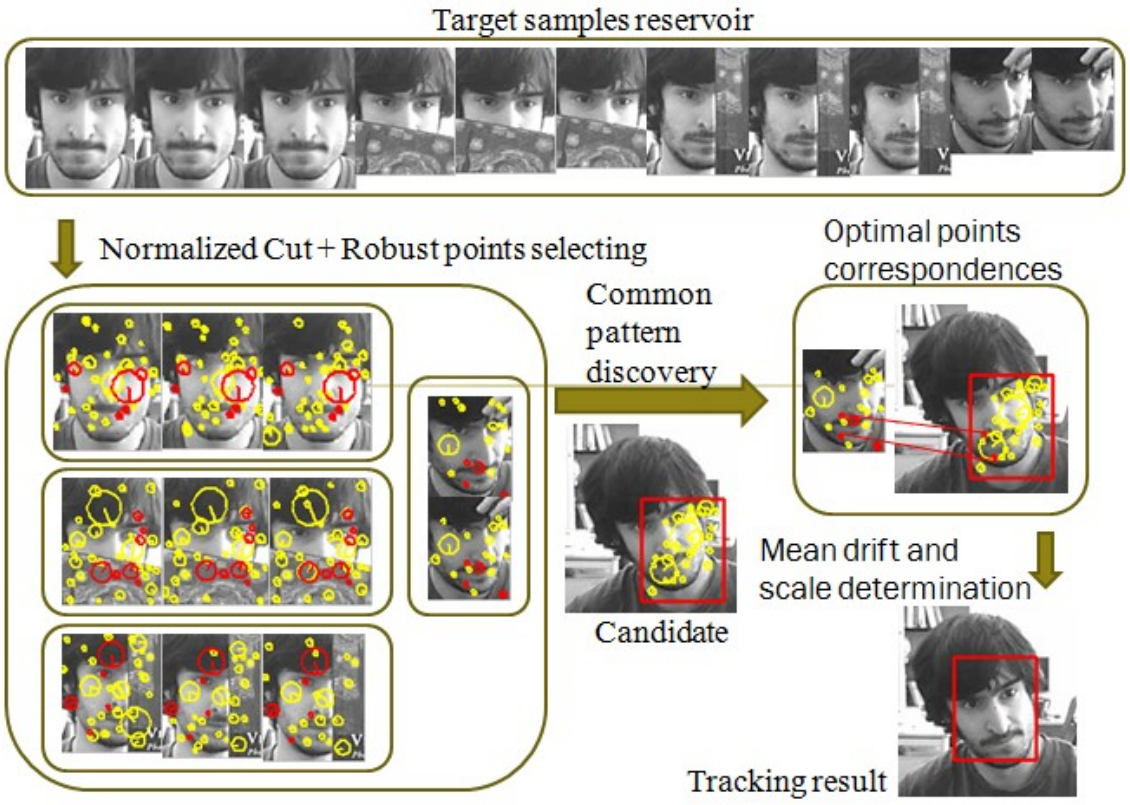

Figure 1. Our tracking-by-correspondences framework.

the target samples and the edge weight of which is defined as the similarity score of every two vertexes. Since the volume of the reservoir should be finite, and both the diversity and robustness of samples should be guaranteed, we cluster such samples into m clusters with Normalized Cut [26], part of samples should be eliminated in order to limit the volume of each cluster to p.

Above sketch our tracking-by-correspondences framework and Figure 1 leads to a better understanding. In the following subsection, we introduce the theoretical details of the proposed approach.

2.1. Robust feature points selection

Given several similar target samples and their feature points from the same cluster $\mathbb{T}_i = \{T_{ij}\} \subset \mathbb{T}$, our goal is to select part of the most robust points so as to avoid occlusion and background clutter on one hand. On the other hand, fewer points lead to fewer

correspondences when discovering common pattern in the motion model. A group of robust points mean they have similar features and exist in similar relative coordinates in different target samples.

Given $\forall i = 1, \ldots, m$, we have $\sum_j |\mathbb{P}_{ij}|$ feature points in total. Our objective is to filter out L robust feature points in each target sample. We construct a weighted graph $\mathbf{G}_i = (V_i, A_i)$, in which each vertex corresponds to one of the $\sum_j |\mathbb{P}_{ij}|$ feature points and A_i is its adjacency matrix. The weight $A_i(u, v)$ between two vertexes u and v represents the similarity of the two points. The similarity can be considered in two aspects. First, we define *the feature similarity* of two points as:

$$S_{if}(u, v) = \exp\left(-\frac{\|p_f(u) - p_f(v)\|_2^2}{\sigma_f^2}\right) \tag{1}$$

Second, we define *the geometric similarity* of two points as:

$$S_{ic}(u, v) = \exp\left(-\frac{\|l_i(u) - l_i(v))\|_2^2}{\sigma_c^2}\right) \tag{2}$$

where

$$l_i(x) = \frac{p_c(x) - c(T_{ij})}{scale(T_{ij})}, j = \arg_j\{x \in \mathbb{P}_{ij}\} \tag{3}$$

Thus, $\forall i = 1, \ldots, m$, $A_i(u, v)$ is defined as:

$$A_i(u, v) = S_{if}(u, v) S_{ic}(u, v) \tag{4}$$

Obviously, A_i is symmetric and nonnegative.

Since we want to gain maximal weight cliques based on formulating and solving an optimization function, a sparser adjacency matrix will ease our solution process. We introduce mutex constraints that specify which points cannot be simultaneously selected as a group of robust feature points. They allow us to eliminate unreasonable configurations which otherwise have large potentials when considering A_i are sometimes unreliable. The proposed mutex constraints are based on the following two insights.

Intra-frame mutex constraint: a robust feature point should always appear in the target. Hence, a group of robust points must be from different target samples of different frames. Within each frame, only one point of a certain group of robust points set should be selected, so as to exclude those points from outliers that resemble the robust points and locate near them.

Inter-frame proximity constraint: two points selected in two neighboring frame should be not spatially far away from each other, since the displacement of the target in proximate frames should be slight.

We encode these two constraints through a binary *mutex matrix* M_i defined over all vertices of graph $\mathbf{G}_i$ as:

$$M_i(u,v) = \begin{cases} 1 & \text{if } (u,v) \in \triangle_1 \cup \triangle_2 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where

$$\triangle_1 = \{(u,v) : u,v \in \mathbb{P}_{ij}, \forall j\} \tag{6}$$

$$\begin{aligned} \triangle_2 = \{(u,v) : \\ \|p_c(u) - p_c(v)\|_2 > \tau, u \in \mathbb{P}_{ij}, v \in \mathbb{P}_{i(j\pm1)}, \forall j\} \end{aligned} \tag{7}$$

and τ reflects the maximum displacement allowed between u and v.

We formulate this robust feature points selection problem as finding constrained maximum weight cliques in graph. The selected points are identified with an indicator vector $\mathbf{x} = (x_1, \ldots, x_n) \in \{0,1\}^n, n = \sum_j |\mathbb{P}_{ij}|$, where a given point v_{ij} is selected if and only if $x_j = 1$.

Referring to [24], we obtain the robust feature points of a given cluster of target samples

by solving the following maximization problem with mutex constraints:

$$\begin{cases} \mathbf{x}^* = \arg_{\mathbf{x}}\{\max_{\mathbf{x}} f(\mathbf{x}) = \mathbf{x}^T A_i \mathbf{x}\} \\ \text{s.t. } \mathbf{x} \in \{0,1\}^n, \mathbf{x}^T M_i \mathbf{x} = 0 \end{cases} \tag{8}$$

[24] converts and relaxes (8) to

$$\begin{cases} \mathbf{x}^* = \arg_{\mathbf{x}}\{\max_{\mathbf{x}} f'(\mathbf{x}) = \mathbf{x}^T A_i \mathbf{x} - \gamma \mathbf{x}^T M_i \mathbf{x}\} \\ \text{s.t. } \mathbf{x} \in [0,1]^n \end{cases} \tag{9}$$

where γ is an arbitrary sufficiently large Lagrange Multiplier.

The algorithm to solve (8) is described in [24]. Multiple initializations are demanded, for we need the algorithm to converge to L local optimums subject to $\bigcap_{k=1}^{L} \mathbf{x}_k^* = \emptyset$. It guarantees any robust point belongs to only one robust point group. Therefore, $\hat{\mathbb{P}}_{ij}, \forall j$ can be selected from all $\mathbf{x}_k^*$. In terms of another given i, the same process is repeated to select another group of $\hat{\mathbb{P}}_{ij}, \forall j$.

The contribution of robust feature points selection is crucial and obvious. Those robust points have very low probability to come from occlusions and background clutters since they are robust in both feature information and geometric information in several previous frames. Figure 2 demonstrates the key effect of this sub-section.

2.2. Robust points correspondences discovery

Given $\hat{\mathbb{P}}_{ij}, \forall i, j$ of any target sample and $\mathbb{Q}_t^s$ of C_t^s, the product space $\mathbb{COMRR} = \hat{\mathbb{P}}_{ij} \times \mathbb{Q}_t^s = \{CORR_n\} = \{(p_n, q_n)\}$ contains all possible correspondences and each correspondence is a pair of feature points from two different images. Exactly the same as (1), we denote *the feature consistency* of a correspondence as:

$$S_f(CORR_i) = \exp\left(-\frac{\|p_f(p_i) - p_f(q_i)\|_2^2}{\sigma_f^2}\right) \tag{10}$$

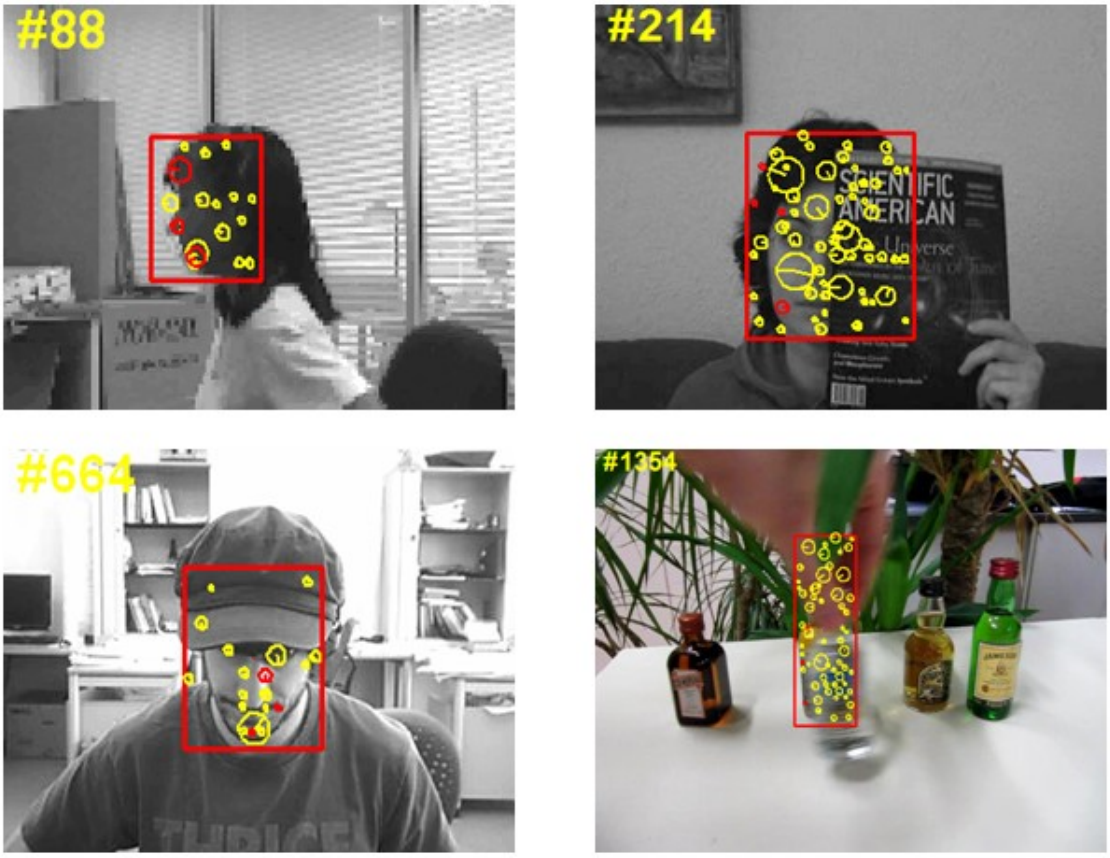

Figure 2. The crucial contribution of robust feature points selection. The above four frames have heavy occlusion or heave appearance change. Even a majority of points coming from occlusion or new appearance, our selection strategy can still select those scattered robust points from target.

Also following (2), for two correspondences $CORR_i = (p_i, q_i)$ and $CORR_j = (p_j, q_j)$, we define their *geometric consistency* as:

$$S_c(CORR_i, CORR_j) = \exp\left(-\frac{\|l(p_i, p_j) - l(q_i, q_j))\|_2^2}{\sigma_c^2}\right) \tag{11}$$

where

$$l(x, y) = \frac{p_c(x) - p_c(y)}{scale(I)}, I = \begin{cases} T_{ij} & x, y \in \hat{\mathbb{P}}_{ij} \\ C_t^s & x, y \in \mathbb{Q}_t^s \end{cases} \tag{12}$$

According to $\mathbb{COPRR}$, we build a weighted graph $\mathbf{G} = (V, A)$ with $N = |\hat{\mathbb{P}}_{ij}||\mathbb{Q}_t^s|$ vertices, each vertex of which represents a correspondence in $\mathbb{COPRR}$. The weighted adjacent

matrix of $\mathbf{G}$ is denoted by A and defined as:

$$A(i,j) = \begin{cases} 0 & i = j \\ S_f(CORR_i)S_f(CORR_j) & \\ \cdot S_c(CORR_i, CORR_j) & i \neq j \end{cases} \tag{13}$$

Obviously, $A(i,j)$ is symmetric and nonnegative.

For a common visual pattern with $k \leq |\hat{\mathbb{P}}_{ij}|$ feature points, $\mathbb{COMM}^* = \{CORR_k^*\}$ responds to a dense subgraph of $\mathbf{G}$ with N vertices, which is a weighted counterpart of maximal clique. Such a dense subgraph has a high *average intra-cluster affinity score*. Thus, according to [25], a connection between the maximal cliques and the local maximizers of a quadratic function is established as:

$$\begin{cases} \mathbf{x}^* = \arg_{\mathbf{x}}\{\max_{\mathbf{x}} g(\mathbf{x}) = \mathbf{x}^T A\mathbf{x}\} \\ \text{s.t. } \mathbf{x} \in \{\mathbf{x} \in \mathbf{R}^n : \mathbf{x} \geq 0, \|\mathbf{x}\|_1 = 1\} \end{cases} \tag{14}$$

The main idea of [25] is illustrated in Figure 3. This maximization problem is similar to (8), but they differ in constraints condition, resulting in two different algorithms. As to this problem, Algorithm 1 in [25] is utilized. Since we do not focus on multiple objects visual tracking, we only care the largest local maxima instead of several of them.

For the optimal maxima $\mathbf{x}^* = \{x_k^*\}$, we need to recover the corresponding common visual pattern. Since x_k^* represents the probability of the common pattern to contain the vertex k, we can recover $\{CORR_k^*\}$ by Algorithm 2 in [25].

2.3. Motion and scale determination

Once $\mathbb{COMM}$ are discovered, the motion of candidate C_t^s based on its correspondences with a given $\hat{\mathbb{P}}_{ij}, \forall i,j$ can be determined by the mean drift of each correspondence, in which the geometric information of robust feature points are made good use of.

We denote the drift of the kth correct correspondence as $\triangle c_k = c(q_k) - c(p_k)$. The weight of such single drift contributing to the mean drift is proportional to its reliability,

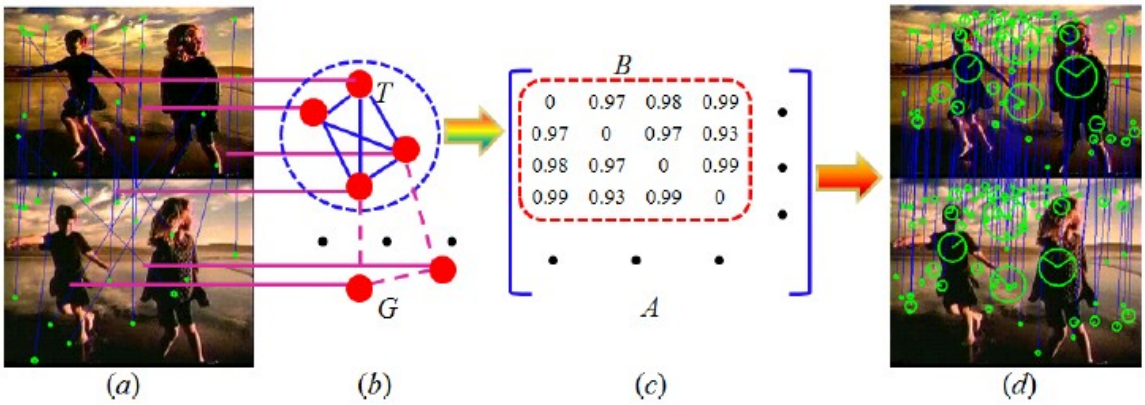

Figure 3. Illustration of the main idea of [25]. Find all candidate correspondences shown in (a) by local features (for clarity, only a small subset of the candidate correspondences are shown), and then form the graph **G** in (b) and weighted adjacent matrix A in (c). The common visual pattern corresponds to the dense subgraph **T** of **G**, and also the dense block B of A after some permutations.

namely the normalized probability of the common pattern. Defining w_k as the normalized x_k^*, the mean drift of candidate is represented as:

$$\overline{\triangle c} = \sum_k w_k \triangle c_k \tag{15}$$

And further, the centroid of the next candidate can be computed by:

$$c(C_t^{s+1}) = c(T_{ij}) + \overline{\triangle c} \tag{16}$$

At the same time, since the scale of a new candidate plays an important role in the adjacency matrixes in subsection 2.1 and subsection 2.2, it should also be adaptively updated thanks to the ratio of mean geometric distances of all pairs of points in one image and those in the other. Namely,

$$\overline{\triangle s} = \frac{1}{C} \sum_i \sum_j \sqrt{w_i w_j} \frac{\|p_c(q_i) - p_c(q_j)\|_2}{\|p_c(p_i) - p_c(p_j)\|_2} \tag{17}$$

$$scale(C_t^{s+1}) = \overline{\triangle s} \cdot scale(T_{ij}) \tag{18}$$

where C is the normalized constant.

Given a $\hat{\mathbb{P}}_{ij}, \forall i, j$, we repeat the process above until $\overline{\triangle c} = 0$ and $\overline{\triangle s} = 1$. Thus, $T_t^{ij} = C_t^{s+1} = C_t^s$. We select $T_t = \arg_{T_t^{ij}}\{\max g(x^{*ij})\}$ as the final tracked target in frame t to add it to the reservoir $\mathbb{T}$.

2.4. Reservoir clustering and updating

A metric learning framework or a tracking-by-detection framework needs large numbers of both positive and negative samples to statistically train a discriminative classifier, while in our tracking-by-correspondences framework, only much fewer positive target samples are required. As a generative approach, we skip training process and a naive and brute exhaustion of comparison of *the average intra-cluster affinity scores* between the candidate and each sample comes feasible. Since a volume limitation of sample reservoir is necessary to ease exhaustion and lower space complexity, the selection of representative samples is crucial. We should give consideration to both the diversity and robustness of reservoir. In terms of diversity, during a sequence the target will probably experience heavy appearance changes, thus several modes of its appearances are quite different but all of the modes should be reserved because they all predict potential appearance of target in the next frame. In terms of robustness, the robust feature points selection process will be more reliable when more similar target samples are involved. Above all, m clusters of no more than p commutatively similar samples make up the reservoir $\mathbb{T} = \{T_{ij}\}, i = 1, \ldots, m, j \in \mathbf{N}^+, j \leq p$. Both m and p are specified parameters according to user's tradeoff between the diversity and robustness of tracking and the computation complexity.

Another tradeoff of reservoir updating strategy is that we sometimes should memorize the original appearance of target when it experiences occlusion on one hand, while on the other hand, we sometimes should else follow the latest appearance of target when its appearance changes largely. Therefore, when a new target sample does not resembling the former ones is added into the reservoir, the tracker will follow such new mode of appearance until one older mode is better corresponded to a new candidate. In this case, the new mode seems an occlusion and after it passes, the tracker still focuses on the original target.

Otherwise, if no older modes are better corresponded, it means the target changes heavily and the new mode deserves the tracker to follow until another new appearance occurs.

In summary, both part of the oldest samples and the latest ones deserves reserved in the reservoir. Once the number of samples in a cluster outnumbers p, we prefer to eliminate the samples of a middle frame index in this cluster. To gain adaptivity, we conduct clustering after tracking each frame, adding the new target sample in the reservoir as well as eliminating such weaker samples mentioned above.

As to clustering, there is no vectorial form of each sample. However, when a new sample is added into the reservoir, it has tried being corresponded to the robust features of all older samples and their own highest *average intra-cluster affinity scores* of each pair have been computed. Also, the highest *average intra-cluster affinity score* of each pair of older samples has been obtained in the previous tracking process. Such scores represent the similarity of two samples which can be seen as edge weights of an affinity graph $\mathbf{G}_{\mathbb{T}}$. It seems that we keep maintaining $\mathbf{G}_{\mathbb{T}}$ of all samples in the reservoir and such graph clustering can be solved through Normalized Cut [26]. There are several versions of codes of Normalized Cut since it was proposed in 2000. The version we use downloads from [27].

3. Algorithm description

At the very beginning, we demand to initialize the target sample reservoir. Since the only meaningful knowledge given to us is the target sample in the first frame, which is insufficient to form an informative reservoir, we have to try a naive algorithm to track the first few frames. Usually, to construct a reservoir with m clusters, we simply track the first m frames and each target sample represents an independent cluster. Based on these samples, we also establish the affinity graph as the original one and then update it once a new sample has been discovered.

Further, by integrating the above-mentioned details in the last section, we propose a visual object tracker whose procedure is shown in the following Algorithm 1.

Algorithm 1 in [24]: Find maximal weight cliques with mutex constraints

Input: Matrix $W = A - \gamma M$, $f(\mathbf{y}_{(0)}) \geq 0$, and $\epsilon > 0$;

repeat

Use

$$(\tilde{\mathbf{x}}_{(k)})_i = \begin{cases} 1 & \text{if } (W\mathbf{y}_{(k)})_i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

to find $\tilde{\mathbf{x}}_{(k)} = \arg\max_{\mathbf{y}\in[0,1]^n} \mathbf{y}W\mathbf{y}_{(k)}$;

if $\tilde{\mathbf{x}}_{(k)} = \mathbf{y}_{(k)}$ **or** $f(\tilde{\mathbf{x}}_{(k)}) > f(\mathbf{y}_{(k)})$ **then**

$\mathbf{y}_{(k+1)} = \tilde{\mathbf{x}}_{(k)}$;

else

$\alpha = -\frac{(\tilde{\mathbf{x}}_{(k)} - \mathbf{y}_{(k)})^T W \mathbf{y}_{(k)}}{(\tilde{\mathbf{x}}_{(k)} - \mathbf{y}_{(k)})^T W (\tilde{\mathbf{x}}_{(k)} - \mathbf{y}_{(k)})}$;

$\mathbf{y}_{(k=1)} = \mathbf{y}_{(k)} + \alpha(\tilde{\mathbf{x}}_{(k)} - \mathbf{y}_{(k)})$;

end if

until $\mathbf{y}_{(k+1)}$ satisfies

$$\begin{cases} \mathbf{x}_i^* = 1 & \text{if } (W\mathbf{x}^*)_i > 0 \\ \mathbf{x}_i^* = 0 & \text{if } (W\mathbf{x}^*)_i < 0 \end{cases} \tag{20}$$

or $f(\mathbf{y}_{(k+1)}) - f(\mathbf{y}_{(k)}) < \epsilon$;

Output: $\mathbf{y}_{(k+1)}$.

Algorithm 1 in [25]: Find the largest local maxima by replicator equation

Input: Weighted adjacent matrix A;
for each vertex v of the graph $\mathbf{G}$ **do**
Build set $\mathbf{T} = \mathbf{N}(v) \cup \{v\}$;
Initialize $x(1)$ in $\mathbf{T}$, that is,

$$x_i(1) = \begin{cases} \frac{1}{|\mathbf{T}|} & \text{if } i \in \mathbf{T} \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

Obtain the corresponding local maximizer $\mathbf{x}^*$ by the replicator equation $x_i(t+1) = x_i(t)\frac{(A\mathbf{x}(t))_i}{\mathbf{x}(t)^T A\mathbf{x}(t)}, i = 1, \ldots, n$;
end
Sort all local maximizers $\{\mathbf{x}^*\}$ according to $f(\mathbf{x}^*)$ in descending order;
Output: The maximizer corresponding to the largest local maixma.

Algorithm 2 in [25]: Recover common visual pattern from local maximizer $\mathbf{x}^*$

Input: Local maximizer $\mathbf{x}^*$;
Sort the components of $\mathbf{x}^*$ in descending order;
Initialize a set $\mathbf{L}$ to be empty;
while *true* **do**
Select the largest component x_i^*;
Test whether all $A(i,j) > \vartheta, j \in \mathbf{L}$, where ϑ is a manually set threshold;
if the test is true **then**
Add i to set $\mathbf{L}$;
Set $\mathbf{x}_i^* = 0$;
else
break;
end if
end
Output: Common visual pattern $\mathbf{L}$.

4. Experimental results

We validate our tracking algorithm on ten challenging sequences from [14] and [18]: *Girl*, *David*, *Faceocc1*, *Faceocc2*, *Dollar*, *Sylverster*, *Board*, *Box*, *Lemming* and *Liquor*, featuring *e.g*. moving cameras, cluttered backgrounds, occlusions, 3D motion or illumination changes. The sequence data, ground truths and results of other methods have also been taken from [14] and [18]. We compare our method with five famous state-of-the-art track-

Algorithm 1: Visual object tracker via robust feature points correspondences

Input: Current frame I_t, reservoir $\mathbb{T}$ and its affinity graph $\mathbf{G}_{\mathbb{T}}$;

Cluster the reservoir into m clusters $\mathbb{T} = \{\mathbb{T}_i\}$ through Algorithm in [26];

for each cluster $\mathbb{T}_i = \{T_{ij}\}$ **do**

while $|\mathbb{T}_i| > p$ **do**

Delete the sample of a middle frame index;

end

Select L groups of robust feature points through Algorithm 1 in [24] and guarantee in each T_{ij} there are L robust feature points $\hat{\mathbb{P}}_{ij} \subseteq \mathbb{P}_{ij}$;

end

for each sample T_{ij} **do**

Initialize candidate C_t where $c(C_t) = c(T_{t-1})$;

while $\overline{\triangle c} \neq 0$ and $\overline{\triangle s} \neq 1$ **do**

Discover common visual patter between $\hat{\mathbb{P}}_{ij}$ and $\mathbb{Q}_t$ through Algorithm 1 and Algorithm 2 in [25];

Denote $S_{ij} = g(\mathbf{x}^*)$ as the highest *average intra-cluster affinity score*;

Normalize x_k^* as w_k;

$\overline{\triangle c} = \sum_k w_k[c(q_k) - c(p_k)]$;

$c(C_t) = c(T_{ij}) + \overline{\triangle c}$;

$\overline{\triangle s} = \frac{1}{C}\sum_i \sum_j \sqrt{w_i w_j}\frac{\|p_c(q_i) - p_c(q_j)\|_2}{\|p_c(p_i) - p_c(p_j)\|_2}$;

$scale(C_t) = \overline{\triangle s} \cdot scale(T_{ij})$;

end

end

Select C_t^* such that S_{ij} are maximized;

Add $T_t = C_t^*$ into $\mathbb{T}$;

Update $\mathbf{G}_{\mathbb{T}}$;

Output: Tracking result T_t, an updated reservoir $\mathbb{T}$ and its new affinity graph $\mathbf{G}_{\mathbb{T}}$.

ing approaches including Online Adaboost tracker (OAB) [17], Online Random Forests tracker (ORF) [28], Fragment tracker (Frag) [7], Multiple Instance Learning tracker (MIL) [14] and Parallel Robust Online Simple Tracker (PROST) [18]. In the comparison, we directly quote the results from [14] and [18] where the best results from each approach were reported. This comparison is summarized in Table 1. In our experiments, we fix our specified parameters once for all: $\sigma_f = 250$, $\sigma_c = 30$, $\tau = 30$, $m = 7$, $p = 4$ and

$$L = \begin{cases} \lceil \sqrt{\sum_j |\mathbb{P}_{ij}| / \max(j)} \rceil & \sum_j |\mathbb{P}_{ij}| / \max(j) > 9 \\ 4 & \text{otherwise} \end{cases} \tag{22}$$

and use the default SIFT parameters in [29]. The quantitative evaluation criterion is the same as what is used in [14] and [18], namely, *Average Center Location Error* (ACLE):

$$e = \frac{1}{M}\sum_{i=1}^{M}\|O_i - O_i^g\|_2 \tag{23}$$

where M is the number of frame and $\|O_i - O_i^g\|_2$ is the Euclidean distance between the tracked target centroid O_i and the ground truth centroid O_i^g.

Sequences	OAB	ORF	Frag	MIL	PROST	**ours**
Girl	43.3	/	26.5	31.6	**19.0**	21.0
David	51.0	/	46.0	15.6	15.3	**9.9**
Faceocc1	49.0	/	**6.5**	18.4	7.0	6.6
Faceocc2	19.6	/	45.1	14.3	17.2	**11.3**
Dollar	24.9	/	55.9	14.7	/	**3.1**
Sylverster	32.9	/	11.2	**9.4**	10.6	22.3
Board	/	154.5	90.1	51.2	37.0	**35.4**
Box	/	145.4	57.4	104.6	**12.1**	29.3
Lemming	/	166.3	82.8	**14.9**	25.4	154.5
Liquor	/	67.3	30.7	165.1	**21.6**	24.6

Table 1. *Average Center Location Error* (ACLE measured in pixels). **Red bold data** indicate the best performance while underlined data indicate the second best one.

Table 1 demonstrates that our tracking-by-correspondences framework outperforms most of traditional online tracking methods, although there exist some exceptions. In details, when **comparing to matching-based methods** like Frag, our approach nearly surpasses it in all sequences and even in *Faceocc1* coming from Frag, our method leads to a comparable result while other trackers are all much weaker than Frag and ours. Frag performs best in *Faceocc1* because it is specifically designed to handle occlusions via a part-based model, however, in a more challenging clip, *Faceocc2*, Frag performs poorly since it cannot tackle appearance changes well. When **comparing to classification-based methods** like OAB and MIL, the former of which trains the classifier for the foreground and background classification via online Adaboost, the latter of which combines multiple instance learning with online Adaboost, and both of which utilize Haar-like features, our method is

always better than both of them even when considering the *Faceocc2* providing by MIL. This highlights our advantages of selecting robust feature points with an adaptive reservoir stategy. When **comparing to optical-flow-based methods** like ORF and PROST, the latter of which combines mean-shift tracker as adaptive element with the former as adaptive appearance-based learner, our algorithm works better in majority of sequences except part of those from PROST in which motion blur occurs. We have to confess the limitation of our approach. For example, in *Lemming*, SIFT descriptor can hardly handle motion blur well because no features are detected in regions of uniform texture. As well, the number of feature points decreases with target patch scaling down, which results in few informative features extracted from the region, like what is in *Sylverster*.

Anyway, the most significant improvements are in the *David*, *Faceocc2* and *Dollar* sequences, which exhibit either heavy appearance changes or large occlusions. Our algorithm successfully keeps the balance of memorizing the original appearance (stability) and following the latest appearance (plasticity) and further extracts the trajectory for all three video sequences. In other sequences, our algorithm either wins or loses by little. Figure 4 shows frame-by-frame *center location error* (CLE) plot for different trackers in different sequences. Figure 5 and Figure 6 show the performance of different trackers on selected frames.

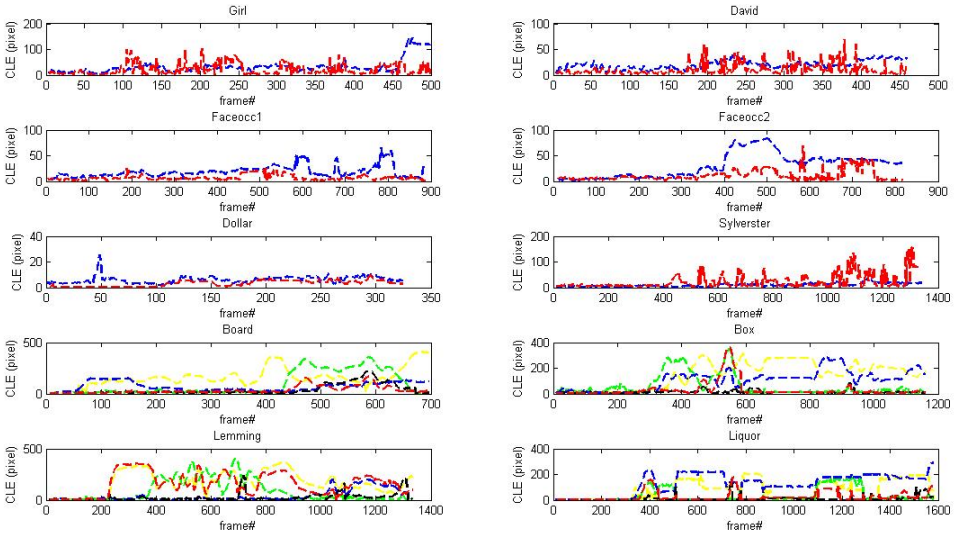

Figure 4. *Center Location Error* (CLE) versus frame number. Each plot shows the performance of ORF (**yellow**), Frag (**green**), MIL (**blue**), PROST (**black**) and ours (**red**).

5. Conclusion

In this paper, we suggest a totally innovative tracking-by-correspondences framework for visual object tracking. We strive for selecting the most robust part of feature points for common visual pattern discovery. Making good use of geometric features of feature points leads to their more informative and global representation of target appearances. Traditional motion model deserves to be replaced since the correct correspondences of robust feature points from two images can determine the location drift and scale change frame by frame in a constant time complexity. In order to keep the balance of robustness and diversity of samples, our generative approach automatically clusters and then updates the positive sample reservoir online. In summary, the proposed method extends the thinking of the state of this art. It is evaluated on several challenging sequences and it is significantly outperforms a large number of state-of-art trackers when coping with shape deformation, occlusion, heavy appearance changes, heavy illumination changes, motion blur, background clutter and such like.

Figure 5. From left to right: *Girl*, *David*, *Faceocc1*, *Faceocc2*, and *Dollar*. *Faceocc1* and *Faceocc2* have significant occlusion. *David* experiences appearance changes. *Girl* has both occlusion and appearance changes. *Dollar* contains the temptation form initial target. For ease of visualization, we show only the comparison between the MIL (**blue**) and ours (**red**). Ours follows the object more closely and handles occlusion better than MIL.

6. Acknowledgement

This work could not be accomplished without the support of many people. Forgive me if I missed some.

I would like to thank my supervisor, Wenyu Liu. Throughout my three-year period in Media and Communication Lab, his rigorous attitude towards research set the best example for me.

Figure 6. From left to right: *Board*, *Box* and *Liquor*. Each column shows the performance of ORF (**yellow**), Frag (**green**), MIL (**blue**), PROST (**black**) and ours (**red**). Ours typically outperforms all other methods in cases with significant occlusion, background clutter, scale and appearance changes.

I would like to thank Yu Zhou, for teaching me about research and fundamental skills and the timely assistance in my study.

I would like to give my sincere thanks to everyone in MC Lab, in particular, to Xiang Bai, Qin Lu and Cong Rao, for their guidance and advices for me to go on the right direction.

I would like to thank all the authors for releasing their sources codes and testing videos, since they made our experimental evaluation possible.

I would like to thank my parents for giving me life and made me the way I am. I love you.

Last but not least, I would give my special thanks to my girlfriend, Yuting Yang, for her love and support. I love you as well.

References

[1] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In IJCAI, 674-679, 1981.

[2] J. Shi and C. Tomasi. Good features to track. In IEEE CVPR, 1994.

[3] M. Isard and A. Blake. A smoothing filter for condensation. In ECCV, 1998.

[4] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In IEEE CVPR, 2000.

[5] C. Tomasi, S. Petrov, and A. Sastry. 3d tracking = classification + interpolation. In ICCV, 2003.

[6] H. Grabner and H. Bischof. On-line boosting and vision. In IEEE CVPR, 2006.

[7] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In IEEE CVPR, 2006.

[8] S. Avidan. Ensemble tracking. IEEE PAMI, 29(2):261-271, 2007.

[9] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In ECCV, 2008.

[10] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade. Tracking in low frame rate video: A cascade particle filter with discriminative observers of different life spans. IEEE PAMI, 30(10):1728-1740, 2008.

[11] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. PROST Parallel Robust Online Simple Tracking. In IEEE CVPR, 2010.

[12] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua. Fast point recognition using random ferns. IEEE Trans. Pattern Anal. Mach. Intell., 32(3):448-461, 2010.

[13] N. Jiang, W. Liu, and Y. Wu. Learning adaptive metric for robust visual tracking. IEEE Transactions on Image Processing, 20(8):2288-2300, 2011.

[14] B. Babenko, M.-H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In CVPR, 2009.

[15] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. IJCV, 77(1):125-141, 2008.

[16] Y. Zhou, X. Bai, W. Liu, L. J. Latecki. Fusion with Diffusion for Robust Visual Tracking. Advances in Neural Information Processing Systems 25. 2012.

[17] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In Proceedings British Machine Vision Conference, 2006.

[18] J. Santner, C. Leistner, A. Saffar, T. Pock, H. Bischof. PROST Parallel Robust Online Simple Tracking. In: IEEE CVPR, 2010.

[19] S. Gu, Y. Zheng, and C. Tomasi. Efficient visual object tracking with online nearest neighbor classifier. In ACCV, 2010.

[20] D. Lowe. Object recognition from local scale-invariant features. In: ICCV, 1999.

[21] H. Bay, T. Tuytelaars, L. Gool. Surf: Speeded up robust features. In: ECCV, 2006.

[22] E. Shechtman, M. Irani. Matching local self-similarities across images and videos. In: IEEE CVPR, 2007.

[23] S. Gu, Y. Zheng, C. Tomasi. Critical nets and beta-stable features for image matching. In: ECCV, 2010.

[24] T. Ma, L. J. Latecki. Maximum Weight Clique with Mutex Constraints for Video Object Segmentation. In: IEEE CVPR, 2012.

[25] H. Liu, S. Yan. Common Visual Pattern Discovery via Spatially Coherent Correspondences. In: IEEE CVPR, 2010.

[26] J. Shi, J. Malik. Normalized Cuts and Image Segmentation. IEEE PAMI, 22(8):888-905, 2000.

[27] `http://www.cis.upenn.edu/~jshi/software`.

[28] L. Breiman. Random Forests. Machine Learning, 45:5-32, 2001,3.

[29] http://www.vlfeat.org/overview/sift.html.