5

Online Learning and Robust Visual Tracking using Local Features and Global Appearances of Video Objects

Irene Y.H. Gu and Zulfiqar H. Khan
Dept. of Signals and Systems, Chalmers Univ. of Technology, Gothenburg, 41296 Sweden

1. Introduction

This chapter describes a novel hybrid visual object tracking scheme that jointly exploits local point features, global appearance and shape of target objects. The hybrid tracker contains two baseline candidate trackers and is formulated under an optical criterion. One baseline tracker, a spatiotemporal SIFT-RANSAC, extracts local feature points separatively for the foreground and background regions. Another baseline tracker, an enhanced anisotropic mean shift, tracks a dynamic object whose global appearance is most similar to the online learned distribution of reference object. An essential building block in the hybrid tracker is the online learning of dynamic object, where we employ a new approach for learning the appearance distribution, and another new approach for updating the two feature point sets. To demonstrate the applications of such online learning approaches to other trackers, we show an example in which online learning is added to an existing JMSPF (joint mean shift and particle filter tracking) tracking scheme, resulting in improved tracking robustness. The proposed hybrid tracker has been tested on numerous videos with a range of complex scenarios where target objects may experience long-term partial occlusions/intersections from other objects, large deformations, abrupt motion changes, dynamic cluttered background/occluding objects having similar color distributions to the target object. Tracking results have shown to be very robust in terms of tracking drift, accuracy and tightness of tracked bounding boxes. The performance of the hybrid tracker is evaluated qualitatively and quantitatively, with comparisons to four existing stat-of-the-art tracking schemes. Limitations of the tracker are also discussed.

2. Related work

2.1 Visual tracking

Visual object tracking has drawn increasing interest in recent years, partly due to its wide variety of applications, e.g., video surveillance in airports, schools, banks, hospitals, traffic, freight, and e-health cares. Tracking is often the first step towards a further analysis about the activities, behaviors, interactions and relationships between objects of interest. Many object tracking methods have been proposed and developed, e.g., state-space based tracking using Kalman filters and particle filters (Welch & Bishop,97; Rosales &

Sclaroff,99; Gordon et al.,01; Gordon,00; Wang et al.,08; Vermaak et al.,03; Okuma et al.,04), joint state-space representation and association (Bar-Shalom & Fortmann,98), multiple hypothesis tracking (MHT) (Reid,79), anisotropic mean shift tracking (Comaniciu et al.,03; Khan & Gu,10), optical flow-based tracking (Shi & Tomasi,94), and point feature-based tracking (Strandmark & Gu,09; Haner & Gu,10), among many others. An overview on visual tracking methods can be found in (Yilmaz et al.,06; Sankaranarayanan et al.,08).

In the state space-based tracking approach using Kalman Filters (KFs), the assumptions of Gaussian noise and linear models of state vector are made. A state vector typically includes different attributes of object, e.g. object appearance and shape, and/or other object features. G.Welch (Welch & Bishop,97) applies KFs to track user's poses in man-computer interactive graphics. R.Roosales (Rosales & Sclaroff,99) uses Extended Kalman Filters (EKFs) to estimate the 3D object trajectory from 2D motions. (Gordon et al.,01) uses Unscented Kalman Filters (UKFs) that enforces Gaussian distributions while keeps nonlinearity by using discrete samples to estimate the mean and covariance in posterior densities. Under Multiple Hypothesis Tests (MHTs), (Reid,79) uses an iterative process to track multiple objects and finds the best matching between the real object descriptors. Gordon et al. (Gordon,00) uses particle filers (PFs) to track 1D (one dimensional) signals. Tracking is achieved by estimating the probability density of state vector from synthetic nonlinear and non-Gaussian distributed 1D signals, and is formulated under the Bayesian framework by estimating the posterior probability using the rule of propagation of state density over time. Extension of PFs to visual tracking is not straight forwards, since the size of state vector for tracking a visual object is significantly larger than that of a 1D target. This requires a large number of particles and consequently a heavy computation, which often hampers the practical use of PFs. To overcome this, (Wang et al.,08) proposes to use Rao-Blackwellized PFs that marginalizes out the linear part of the state vector (the appearance), while the nonlinear shape and pose parts are then estimated by PFs, while (Khan et al.,09; Deguchi et al.,04) propose to embed the object appearance in the likelihood of PFs so that the size of the state vector can be kept small.

Visual tracking from mean shift has drawn much interest lately, partly due to its computational efficiency and relatively robust performance. Different from the conventional mean shift for nonlinear image smoothing or segmentation that seeks the local modes in the kernel estimate of pdf, mean shift tracking is an efficient and fast implementation of the similarity metric, the Bhattacharyya coefficient, that maximizes the similarity between the reference and a candidate object regions. It is worth mentioning that other similarity metrics, e.g., Kullback-Leibler divergence (Khalid et al.,05), or SSD measure (Hager et al.,04), can also be used as well. The main drawback of mean shift is that tracking may drift away or fail especially when the background clutter and the object of interest have similar color distributions, or when long term partial occlusions of objects, pose changes of large objects, and fast change of object motion occur. Following the pioneering work of mean shift tracking by (Comaniciu et al.,03), various attempts are made to address these issues. (Collins,03) extends the mean shift by introducing a normalizing factor to the bandwidth matrix to capture target variations in scales (Bretzner & Lindeberg,98). It performs extensive search within a range of ellipses and is computationally expensive. (Yilmaz,07) proposes tracking by using a level-set asymmetric kernel. It is performed in image coordinates by including the scale and orientation as additional dimensions and simultaneously estimating all unknowns. (Sumin & Xianwu,08) proposes to simultaneously track the position, scale and orientation of bounding box by using anisotropic mean shift, where the bandwidth matrix is used to compute the scale and orientation. (Zivkovic & Krose,04) proposes

an EM-like algorithm that tracks a deformable object whose bounding box contains five degrees of freedom. It simultaneously estimates the center and the bandwidth matrix of kernel. (Maggio & Cavallaro,05; Xu et al.,05; Parameswaran et al.,07; Khan et al.,09) include the spatial information in the color histogram by dividing an ellipse shape bounding box into multiple parts to make the tracker more robust. (Maggio & Cavallaro,05; Khan et al.,09) further integrate the multi-part mean shift into the particle filter framework using overlapped, or non-overlapped regions where improved results are reported. The tracking performance is rather robust, however, tracking drift or tracking failure may still occur in some occasions, especially when a cluttered background or an intersecting object has similar color distributions to the target object.

While global appearance distributions are widely used in visual object tracking, local point features of object are often used as an alternative. One of the main advantages of using point features is their resilience to partial object occlusions. When one part of an object is occluded, point features from the non-occluded part can still be used for the tracking. Local appearance-based tracking usually involves detecting and characterizing the appearance of object by local features from points, lines or curves, establishing correspondences between detected feature points (lines, or curves) across frames and estimating the parameters of the associated transformation between two feature point (line, or curve) sets. Several strategies are used to select feature points that are invariant to affine or projective transformations. (Harris & Stephens,88) proposes to extract rotational and translational-invariant features by combining corner and edge detectors based on local autocorrelation functions. (Shi & Tomasi,94) proposes to threshold the minimum eigen values of image gradient matrices at candidate feature points and use them as the appropriate feature points for tracking. These methods generate rotational and translational invariant point features however are variant to affine or projective transformations. (Lowe,04) proposes a Scale-Invariant Feature Transform (SIFT) that is invariant to rotations, translations, scalings, affine transformations, and partially invariant to illuminations. Each point feature is described by a feature descriptor or a vector, formed from the gradient directions and the magnitudes around the point. (Bay et al.,06) proposes to use Speeded Up Robust Features (SURF), having similar performance as SIFT (Bauer et al.,07) but faster. Due to the robustness of SIFT features, various attempts (Skrypnyk & Lowe,04; Mondragon wt al.,07; Li et al,06; Xu et al,08; Battiato et al.,07) have been made to integrate SIFT in the tracking. (Skrypnyk & Lowe,04) proposes to use SIFT features to track camera poses and to register virtual objects in online videos. Since feature points are often sensitive to noise, the consensus of a group of points can be exploited. (Mondragon wt al.,07) uses SIFT and RANSAC to detect points of interest and reject outliers when estimating projective transformations, where videos are captured by an online UAV camera. (Li et al,06) handles object occlusions by matching local invariant features learned online rather than predicted from motions, since the local point features from a non-occluded object part can still be used for the tracking. (Xu et al,08) proposes vehicle tracking by using SIFT features extracted from detected moving object bounding boxes, followed by frame-by-frame matching. (Battiato et al.,07) proposes a video stabilizer by inferring the inter-frame motion through SIFT feature tracking in consecutive frames. These methods are efficient and invariant to scaling, rotation and moderate lighting changes, however require the appearance of object containing sufficient textures. Several attempts are made to solve this problem by combining SIFT features with other tracking methods. To extend the visual tracking from 2 images to a video sequence, efficient methods for establishing spatiotemporal local feature point correspondences through video frames

are required. (Strandmark & Gu,09) proposes multiple motion models and feature point maintenance by employing an online updating process that may add new feature points, prune the existing points, or temporally freeze the updating, and (Haner & Gu,10) further improves the method by introducing two local feature point sets, one for the foreground and another for the background, where the background point set is used to provide priors on possible occlusions.

While both the global and local object models offer some attractive properties for visual tracking, hybrid models that combines these two types of models may offer better results as they can complement each other. (Zhou et al.,08) proposes an expectation-maximization algorithm that integrates SIFT features along with the color-based appearance in the mean shift, resulting in a better tracking performance even if one of the two methods becomes unreliable. (Wu et al.,08) enhances the performance of particle filters in cluttered background by taking into account the SIFT features in particle weights along with the color similarity measure. (Zhao et al.,08) uses feature point analysis to recover affine parameters, from which relative scales between two frames are estimated. It reconstructs target positions and relative scales using the affine parameters estimated from the SIFT feature correspondences. (Chen et al.,08) uses a similar method to handle the occlusion and scaling under the mean shift framework. While local feature points are promising for tracking, several problems remain, e.g. lacking of SIFT point features in smooth objects; lacking of sufficient feature point correspondences through video frames especially when the object contains pose changes, intersections and large deformations. (Khan & Gu,10) proposes to combine an enhanced anisotropic mean shift and a spatiotemporal SIFT-RANSAC procedure into a unified framework by using an optimal criterion, where the mean shift seeks global object appearance similarity and the spatiotemporal SIFT-RANSAC finds local feature points in the foreground/background. To further enhance the robustness against the tracking drift, the scheme also includes online learning of global appearances and local features.

2.2 Online learning

Online learning is another key issue that significantly impacts the performance of visual tracking. Most tracking methods require some kind of reference object models. Offline training videos from the same target object usually are not available, since video scenes from specific objects (e.g. suspicious actions of a particular person) captured by a surveillance camera are rarely repeatable. For tracking dynamic target objects, online learning of reference object appearances and/or shape is thus crucial. This is not trivial as the change of object can be caused by the object itself (e.g. change in colors, poses or shape), but also by partial/full occlusions from an occluding object or cluttered background, in addition to other changes such as lighting and illuminations. Online learning of dynamic objects requires that only the change associated with the target object itself is learned/updated into the model, while the remaining change from the background or other objects does not trigger the learning process. This is challenging since we have neither priors on the background/occluding objects, nor the information on when and where an occlusion may occur. Techniques for online learning vary depending on the attributes of object (e.g. global/local appearance, shape, motion) to be learned. Further, it depends on the technique used in the visual tracking as tracking and online learning are usually incorporated under a same framework. Many online learning techniques have been proposed. For example, (Lim et al.,04; Yang et al.,04; Wang et al.,07) perform incremental learning of 1D/2D PCA that describes the object appearance in a visual tracker. (Wang et al.,08) proposes an online Grassmann manifold learning scheme where

dynamic object appearances are constrained on a smooth curved surface rather than a linear subspace. For online learning of pdf associated with each individual pixels, (Li et al.,08) suggests a color co-occurrence based method to learn the time-varying principal pdf of individual pixels that contain motions. For online learning of object appearance pdf used in the mean shift, (Khan & Gu,10) propose a robust online learning method in the regular frame intervals based on the criterion that determines whether the change is likely caused by the target object. For online learning of local features, (Strandmark & Gu,09) proposes multiple motion models and dynamic maintenance of the feature point set by using SIFT and RANSAC allowing online adding and pruning the feature points, or freezing the updating. (Haner & Gu,10; Khan & Gu,10) further improve the method by applying dynamic maintenance of separate foreground and background feature point sets under different criteria, where the background set is used to provide priors on occlusions to the foreground object.

2.3 Chapter outline

This chapter is focused on describing a hybrid visual tracking scheme, where both the local features and the global dynamic object appearance and shape are exploited. A key component of the tracker, the online learning, is employed to two baseline trackers: one is used to maintain the dynamic local feature point sets, and the other is used to learn the global appearance of dynamic object. The hybrid tracking scheme combines local point features and global appearances. It includes: (a) A local point feature-based candidate tracking method by employing consensus point feature correspondences separately in the foreground set and in the surrounding background set through using a spatiotemporal SIFT-RANSAC procedure. They are accompanied with an online maintenance process that can add, prune, freeze and re-initialize feature points in the sets; (b) A global appearance similarity-based candidate tracking method by using an enhanced anisotropic mean shift whose initial kernel is partially guided by the local point features, and is equipped with the online learning of reference object distribution; (c) The final hybrid tracker by combining the candidate trackers in (a) and (b) through using an optimal criterion.

We then show that the online learning strategy for the candidate tracker in (b) can be directly applied to the online learning of another state-of-the-art visual tracking method, the joint anisotropic mean shift and particle filter (JMSPF) tracker, which may result in further tracking robustness in terms of tracking drift, tightness of tracked bounding boxes, and tracking failures in complex scenarios.

Experimental results on visual tracking of video objects with a range of difficult scenarios are included. Several distance metrics are used to quantitatively evaluate the robustness of the tracker, to evaluate the performance of the tracker with and without the online learning. Further performance evaluations are made qualitatively with three existing trackers, and quantitatively with two existing trackers. The computations are also compared for the hybrid tracker and three existing trackers.

The remainder of the chapter is organized as follows. The general structure and overall description of the hybrid tracker are given in Section 3. In Section 4, we describe two baseline trackers, one is based on using local feature point correspondences extracted from the spatiotemporal SIFT-RANSAC, and the other is based on the global object appearance similarities. In particular, Section 5 emphasizes two online learning techniques employed to these two baseline trackers. In Section 6, the hybrid tracker is formulated from the two baseline trackers under an optimal criterion. Section 7 describes a direct application of the

online learning method to an existing joint anisotropic mean shift and particle filter (JMSPF) tracker. Section 8 is contributed to the experimental results and performance evaluations aimed at demonstrating the feasibility and robustness of the hybrid tracker. The advantages and limitations are also discussed. Finally, conclusions are given in Section 9.

3. A Robust hybrid visual tracking scheme: The big picture

This section describes the general structure and gives the big picture of the hybrid tracking scheme, where multiple issues are treated in a unified tracking framework. The tracking scheme, as shown in the block diagram of Fig.1, can be split into several basic modules. This is briefly summarized as follows:

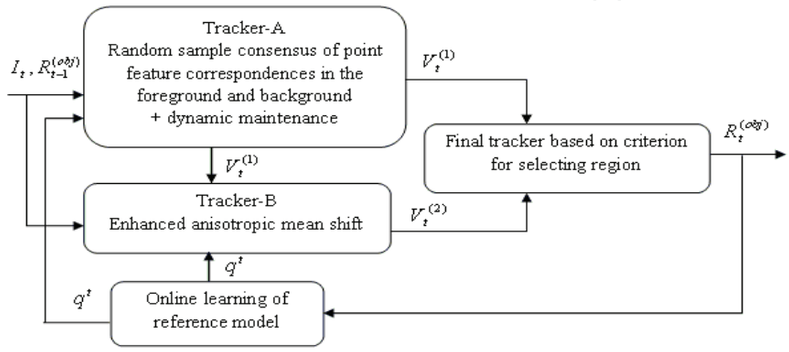


Fig. 1. Block diagram of the hybrid tracking scheme, where $V_t^{(i)}$, $i = 1, 2$, is the parameter vector for the tracked candidate region $R_t^{(i)}$ from the baseline tracker-A and tracker-B at time t, I_t is the tth video frame, $R_{t-1}^{(obj)}$ is the finally tracked object region from the hybrid tracker at $t-1$, and q^t is the estimated appearance pdf for the reference object at t.

(a) *Two online learning methods.* Two novel methods for online learning of dynamic object are described: one is used for online maintenance of local point feature sets, and the other is for dynamically updating the object appearance distribution. This is aimed at keeping a time-varying reference object description thereby the tracking drift can be reduced. The method is based on seeking the best frame, indicated by reliable tracking without occlusions, in each fixed-size frame interval. This is achieved by using a criterion function in the interval.

(b) *Baseline tracker-A: exploit local point features for object tracking.* The baseline tracker-A in the hybrid tracker is realized by exploiting the local point features. Local point features are useful when partial occlusions occur: point features in non-occluded part remain unchanged, despite the global object appearance may experience significant changes. To make the point feature-based tracking robust, two sets of point features are utilized: one for the foreground region and the other for the background region (see Fig.1). The idea of using the background point features is to provide priors to the foreground on possible occlusions. The local point features are collected by introducing a novel spatiotemporal SIFT-RANSAC procedure followed by an online maintenance process that may add, prune and update feature points in both sets. To prevent drift and error propagation, a re-initialization process is also used.

(c) *Baseline tracker-B: exploit global object appearance for object tracking.* The baseline tracker-B in the hybrid tracker is realized by exploiting global object appearance distributions. To find the most similar appearance object compared with the reference (e.g., previously tracked object, or a reference object), anisotropic mean shift with a 5-degree parametric bounding box is used. An enhancement is added to the conventional mean shift by allowing its kernel partially guided by the local point features. This may reduce the tracking drift as the mean shift is sensitive to cluttered background/occluding objects having the similar color distribution to the foreground object. Online learning of object appearance distributions and re-initialization introduced for achieving more tracking robustness and preventing propagation of tracking drift across frames.

(d) *The hybrid tracking scheme: formulated from an optimal criterion.* The above local point feature-based tracker and global object appearance-based tracker in (b) and (c) are exploited jointly to form the hybrid tracker. It is formulated by using an optimal criterion that parallel employs the two baseline trackers and one from their weighted combination.

4. Baseline tracking methods using local features and global appearances

This section describes two baseline tracking methods, one is based on using local point features of object (Tracker-A in Fig.1), and the other is based on using global object appearance (Tracker-B in Fig.1).

4.1 Local feature point-based visual tracking

This section describes one baseline tracking method, tracker-A, in the hybrid tracker shown in Fig.1. It is a local feature point-based tracking method realized by a *spatiotemporal SIFT-RANSAC* procedure. It generates separate local feature point sets in the foreground and the surrounding background respectively.

The use of local point features is motivated by problems encountered in tracking partially occluded objects, or objects having similar color distributions to the cluttered background. In these scenarios, local salient point features from non-occluded object parts, or salient point features of object may be exploited for tracking. For matching local point features, two well known computer vision techniques are employed: SIFT (scale-invariant feature transform) (Lowe,04) and RANSAC (random sample consensus) (Fischler & Bolles,81) are used. The former is used to match scale-invariant feature points, while the latter is used to remove outliers. A brief review of SIFT and RANSAC is given in Section 4.1.1. For introducing more robustness to partial occlusions/itersections, two sets of feature points, one to the foreground area and another to the background area surrounding the candidate object, are employed. The background set is used to provide priors on occlusions. This is described in Section 4.1.2.

It is worth emphasizing that one of the key steps to realize the spatiotemporal SIFT-RANSAC is the online maintenance of point feature sets (in Section 5.1), while the conventional SIFT and RANSAC usually cannot be applied successfully to a long video sequence (e.g. of a few hundreds of frames).

4.1.1 Review: SIFT and RANSAC for feature point correspondences

Two standard computer vision techniques, SIFT (Lowe,04) and RANSAC (Fischler & Bolles,81), are briefly reviewed in this subsection.

In SIFT, each point feature is described by a feature vector

$$\mathbf{f}_i = \{p_i, \Phi_i\} = \{p_i, \sigma_i, \varphi_i, \mathbf{g}_{h_i}\} \tag{1}$$

where $p_i = (x_i, y_i)$ is the 2D position of SIFT keypoint, $\Phi_i = \{\sigma_i, \varphi_i, \mathbf{g}_{h_i}\}$ is the parameter vector associated with the point p_i, including the scale σ_i, the main gradient orientation within the region φ_i and the gradient orientation histogram $\mathbf{g}_{h_i}$ (128 bins). First, the original image I is convolved with a bandpass filter h to obtain a filtered image I_o, $I_o(x, y, \sigma) = I(x, y) * h(x, y, k\sigma)$, where $h(\cdot)$ is formulated from the difference of two scale Gaussian shape kernels $h(x, y, \sigma) = g(x, y, k\sigma) - g(x, y, \sigma)$. The location of each feature point (SIFT keypoint) $p_i, i = 1, 2, \cdots$, at scale σ_i corresponds to the thresholded local extrema of $I_o(x, y, \sigma)$. For each SIFT keypoint, one or more principal orientations θ_i are assigned by computing the gradient magnitudes and orientations in a region surrounding the point and finding 80% peaks in the orientation histogram. The orientation histogram is computed from a 16×16 region centered at p_i, partitioned into 4×4 blocks each consisting of 8 bins. This results in a total of $8 \times 16 = 128$ bins. The value of orientation histogram $\mathbf{h}_i$ is obtained by summing up the gradient magnitudes in each bin.

Matching SIFT keypoints in two image frames is obtained by searching the Euclidian distance from the nearest neighboring keypoints with the minimum errors. Under a pre-defined motion model, corresponding SIFT keypoints across two image frames are related. For example, if two images are related to an affine transform $T(\beta, \theta, (d_x, d_y))$, then each pair of SIFT keypoints is related by $\begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} = \beta \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix}$. Equivalently, given two sets of keypoints $\{(x_1, y_1), \cdots, (x_n, y_n)\}$ and $\{(\tilde{x}_1, \tilde{y}_1), \cdots, (\tilde{x}_n, \tilde{y}_n)\}$ from two images (e.g. at $(t-1)$ and t), these pairs of keypoints are related by,

$$\begin{bmatrix} \tilde{x}_1 \\ \tilde{y}_1 \\ \vdots \\ \tilde{x}_n \\ \tilde{y}_n \end{bmatrix} = \begin{bmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ \vdots & & \vdots & \\ x_n & -y_n & 1 & 0 \\ y_n & x_n & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta\cos\theta \\ \beta\sin\theta \\ d_x \\ d_y \end{bmatrix} \tag{2}$$

The LS (Least Squares) estimation, $argmin_{(\beta,\theta,d_x,d_y,\text{ index order of } p_j)} ||\tilde{p}_i - T(p_j)||_2$, can be applied to find the best matching and to estimate the transform parameters β, θ, (d_x, d_y) at t. To make the matching robust against the false positive, the best matching is selected if the ratio of distances between the first and the second nearest neighbors is less than some empirically determined threshold (Lowe,04).

Since individual SIFT keypoints are prune to noise, RANSAC is often followed to remove outliers through finding the maximum number of consensus correspondences and estimating the associated motion parameters. RANSAC contains a two-step iterative process: estimate the parameters of the transform $T_i^{(t)}$ and find a subset of inlier points from the SIFT keypoints that yield the maximum consensus under $T_{i^*}^{(t)}$. In the first step, it differentiates outliers from inliers by selecting a minimum number of points needed to estimate the transform from the SIFT keypoint set at random. Then, the parameters of the transform are estimated. Using the estimated parameters, more points are picked up if they fit to this specific transform. This is done by calculating the error for each pair of keypoints and comparing with a small error threshold T_e. The iteration is repeated until the error is smaller than T_e, or the total number of iterations exceeds a pre-specified maximum iteration number T_{iter}. In the second step, it fits the transform to the inliers while ignoring the outliers. The transform parameters are updated using all collected inlier points. A tracked candidate object region is then obtained by drawing a tight rectangle surrounding the consensus points.

4.1.2 Using separate feature point sets for the foreground and the background

In the local point feature-based baseline tracker, two separate feature point sets, $\mathcal{P}^F$ and $\mathcal{P}^B$, are formed. $\mathcal{P}^F = \{p_i^F \mid p_i^F : \Phi_i^F\}$ is for the foreground, and $\mathcal{P}^B = \{p_i^B \mid p_i^B : \Phi_i^B\}$ is for the background surrounding the candidate object. In each set, the parameter vector Φ_i^F (or, Φ_i^B) is defined according to (1). The basic idea of using background points is to extract priors on possible object occlusions or intersections. As shown in Fig.2, a *searching area* (the black rectangle) is defined to be larger than that of an object region (the red rectangle). The region between the searching area and the candidate object region (between the black and red rectangles) is defined as the background region.

Fig. 2. Foreground and background regions. Red rectangle: a foreground object region; Black rectangle: the searching area. The area between the black and red rectangles is the background region. Red ellipse: maybe used for some objects (e.g humans) to minimize the possible inclusion of background points around the four corner areas.

Both sets of feature points are extracted by combining SIFT and RANSAC as described above, however, in different regions. Finally, a tight outer rectangular boundary surrounding the selected foreground feature points in $\mathcal{P}^F$ is drawn as the tracked candidate object region $R^{(1)}$ for the baseline tracker-A. The region is specified by the parameter vector $V^{(1)} = [p_c = (y_{1,c}^{(1)}, y_{2,c}^{(1)}), w^{(1)}, h^{(1)}, \theta^{(1)}, \mathcal{P}^F]^T$ including the 2D center position, the width, height and orientation of the region, and the foreground sets. To reduce the possibility of including background points in the foreground set, the shape of object type may be considered. For example, for human objects, foreground feature points may be constrained within an ellipse that is tightly made within the rectangular region $R^{(1)}$.

It is worth mentioning that the resulting region $R^{(1)}$ is also provided to the baseline tracker-B (in Section 4.2) to enhance the conventional mean shift which is prune to the background or occluding objects with similar color distributions.

4.1.3 Re-initialization

A re-initialization process may be applied to some frames to prevent tracking drift or tracking error propagation across frames. The idea can be analogue to using I (intra-coding) frames in video compression. A re-initialization process for the baseline tracker-A is used to avoid severe errors, e.g., when the number of corresponding points is very small, unreliable tracking or accumulated tracking drift may occur. In the former case, a small number of feature points may lead to an ill-posed RANSAC estimation, or a unreliably tracked region. In the latter case, accumulated drift may eventually lead to tracking failure. A tracker thus needs to be re-initialized to avoid the propagation of errors across frames.

Based on the observation that a bounding box does not change significantly in consecutive frames, and that a very low similarity value between the tracked region and the reference object indicates a possible tracking drift or unstable tracking, the frames for re-initialization

can be selected. Two conditions, the distance of consecutive box shape and the Bhattacharyya coefficient between the tracked and the reference object regions, are used to determine whether a re-initialization is applied. That is, if one of the following two conditions is satisfied,

$$dist_t^{(1)} = \sum_{i=1}^{4} \|\mathbf{x}_{t,i}^{(1)} - \mathbf{x}_{t-1,i}^{(1)}\|^2 > T_1^{(1)}, \text{ or } \rho_t^{(1)} < T_2^{(1)} \tag{3}$$

then the baseline tracker-A is re-initialized at t by:

$$R_t^{(1)} \leftarrow R_{t-1}^{(obj)}, \; V_t^{(1)} \leftarrow V_{t-1}^{(obj)}, \text{ and } \tilde{\rho}_t^{(1)} \leftarrow 0 \tag{4}$$

where $R_{t-1}^{(obj)}$ is the tracked bounding box from the final hybrid tracker at $(t-1)$, and $V_{t-1}^{(obj)}$ is the corresponding parameter vector, $\tilde{\rho}_t^{(1)}$ is the normalized Bhattacharyya coefficient defined in (21), $\mathbf{x}_{t,i}^{(1)}$ and $\mathbf{x}_{t-1,i}^{(1)}$ are the four corners of the object bounding box at t and $(t-1)$ from the baseline tracker-A, $\rho_t^{(1)}$ is the Bhattacharyya coefficient for the baseline tracker-A, and $T_1^{(1)}$ and $T_2^{(1)}$ are the empirically determined thresholds.

4.2 Global appearance-based visual tracking

This section describes another baseline tracking method, tracker-B, in the hybrid tracker shown in Fig.1. It is a global object appearance-based tracking method realized by an enhanced anisotropic mean shift. Based on the observation that mean shift tracking yields reasonably good tracking results, however, tracking drift may occur when nearby objects or background clutter have similar color distributions. To tackle the problem, two strategies are introduced here to the mean shift: The first one is to employ an *enhanced* anisotropic mean shift, where the result from the point feature-based tracking (the baseline tracker-A) is used to partially guild the location of mean shift kernel. The second strategy is to add online learning of reference object appearance (in Section 5.2), as well as a re-initialization process to prevent the propagation of tracking drift. This baseline tracker-B results in a tracked candidate object region $R^{(2)}$ specified by a parameter vector $V^{(2)} = [y^{(2)} = (y_1^{(2)}, y_2^{(2)}), w^{(2)}, h^{(2)}, \theta^{(2)}, \mathbf{h}_{rgb}^{(2)}]^T$, where $y^{(2)} = (y_1^{(2)}, y_2^{(2)})$, $w^{(2)}$, $h^{(2)}$ and $\theta^{(2)}$ are the 2D center, width, height and orientation of $R^{(2)}$, and $\mathbf{h}_{rgb}^{(2)}$ is the color histogram.

4.2.1 Anisotropic mean shift

This subsection briefly reviews the anisotropic mean shift for visual tracking. More details of mean shift can be found in (Sumin & Xianwu,08; Comaniciu et al.,03).

Let the pdf estimate $p(y, \Sigma) = \{p_u(y, \Sigma), u = 1, \cdots, m\}$ for a candidate object be the spatial kernel-weighted color histogram within the bounding box in the image $I(y)$, and the spatial kernel-weighted color histogram $q(x_c, \Sigma_c) = \{q_u(x_c, \Sigma_c), u = 1, \cdots, m\}$ for the reference object within the bounding box in $I_0(x)$. The corresponding histogram bins for the candidate and reference objects are described respectively as follows:

$$\begin{aligned} p_u(y, \Sigma) &= \frac{c}{|\Sigma|^{\frac{1}{2}}} \sum_{j=1}^{n} k(\tilde{y}_j{}^T \Sigma^{-1} \tilde{y}_j) \delta[b_u(I(y_j)) - u] \\ q_u(x_c, \Sigma_c) &= \frac{c_0}{|\Sigma_c|^{\frac{1}{2}}} \sum_{j=1}^{n} k(\tilde{x}_j{}^T \Sigma_c^{-1} \tilde{x}_j) \delta[b_u(I_0(x_j)) - u] \end{aligned} \tag{5}$$

where $\tilde{y}_j = (y_j - y)$, $\tilde{x}_j = (x_j - x_c)$, Σ (or, Σ_c) is a kernel bandwidth matrix, $b_u(I(y_j))$ (or, $b_u(I_0(x_j))$) is the index of color histogram bin at the location y_j (or, x_j) associated with the

candidate (or, reference) object region, y_j (or, x_j) is summed over all pixels within the bounding box, c (or, c_0) is a constant used for the normalization, m is the total number of bins, $k(\cdot)$ is the spatial kernel profile, and y (or, x_c) is the center of the kernel (or, bounding box).
To measure the similarity between a candidate and the reference object region, the Bhattacharyya coefficient ρ defined below, is used:

$$\rho(p,q) = \sum_{u=1}^{m} \sqrt{p_u(y,\Sigma) q_u} \tag{6}$$

Applying the first-order Taylor series expansion to (6) around (y_0, Σ_0), (where y_0 and Σ_0 are the kernel center and bandwidth in the previous frame) yields $\rho \approx \sum_u \frac{1}{2}\sqrt{q_u p_u(y_0,\Sigma_0)} + \frac{c}{2|\Sigma|^{\frac{1}{2}}} \sum_j \omega_j k(\tilde{y}_j^T \Sigma^{-1} \tilde{y}_j)$, where $\omega_j = \sum_u \sqrt{\frac{q_u}{p_u(y_0,\Sigma_0)}} \delta[b_u(I(y_j)) - u]$. The kernel center (or, bounding box center) can be estimated by setting $\nabla_y \rho(p,q) = 0$. This leads to:

$$\hat{y} = \frac{\sum_{j=1}^{n} g(\tilde{y}_j^T \Sigma^{-1} \tilde{y}_j) \omega_j x_j}{\sum_{j=1}^{n} g(\tilde{y}_j^T \Sigma^{-1} \tilde{y}_j) \omega_j} \tag{7}$$

where $g(\cdot) = -k'(\cdot)$ is the shadow of the kernel. To estimate $\hat{\Sigma}$, a γ-normalized kernel bandwidth Σ (in (Bretzner & Lindeberg,98)) is applied to ρ, where y in $\tilde{y}_j$ is substituted by $\hat{y}$ that is obtained from (7). The kernel bandwidth matrix is estimated by setting $\nabla_\Sigma(|\Sigma|^{\gamma/2}\rho(p,q)) = 0$, yielding,

$$\hat{\Sigma} = \frac{2}{1-\gamma} \frac{\sum_{j=1}^{n} \omega_j \, g(\tilde{y}_j^T \Sigma^{-1} \tilde{y}_j) \tilde{y}_j^T \tilde{y}_j}{\sum_{j=1}^{n} \omega_j \, k(\tilde{y}_j^T \Sigma^{-1} \tilde{y}_j)} \tag{8}$$

where γ is empirically determined, and $\tilde{y}_j = (y_j - \hat{y})$. The estimation of (7) and (8) are done alternatively in each iteration. The iteration repeats until the estimated parameters converge, or a pre-specified maximum number of iterations is reached.

4.2.2 Estimating shape parameters of bounding box

For estimating the parameters of bounding box in the baseline tracker-B, a simple approach different from (Sumin & Xianwu,08) is employed. The anisotropic mean shift used in the baseline tracker-B contains a fully tunable affine box with five degrees of freedom, i.e., the 2D central position, width, height and orientation of the box.
Let the orientation of the box be defined as the angle between the long axis of bandwidth matrix and the horizontal-axis of the coordinate system. Let the height h and width w of the bounding box be defined as the radii along the long and short axes of an ellipse, as depicted in Fig.3. Since h, w, and θ are related to the kernel bandwidth matrix Σ by,

$$\Sigma = \mathbf{R}^T(\theta) \begin{bmatrix} (h/2)^2 & 0 \\ 0 & (w/2)^2 \end{bmatrix} \mathbf{R}(\theta) \tag{9}$$

where $\mathbf{R} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$, computing these parameters can be efficiently done by applying eigenvalue decomposition to Σ,

$$\Sigma = V \Lambda V^{-1} \tag{10}$$

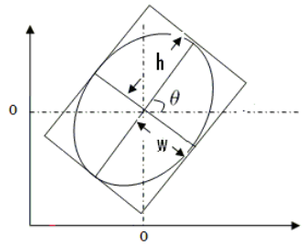


Fig. 3. Definition of the width, height and orientation of a bounding box.

where $V = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix}$, $\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$, and by relating these parameters with eigenvectors and eigenvalues using (9) and (10) as follows,

$$\hat{\theta} = \tan^{-1}(v_{2,1}/v_{1,1}), \;\; \hat{h} = 2\sqrt{\lambda_1}, \;\; \hat{w} = 2\sqrt{\lambda_2} \tag{11}$$

where v_{11} and v_{21} are the two components from the largest eigenvector. The first five parameters in $V^{(2)}$ are obtained from the estimates in (7), (8) and (11).

4.2.3 Enhancing the anisotropic mean shift

The basic idea of enhanced mean shift is to partially guild the location of mean shift kernel from the result of local feature point-based tracker (or, baseline tracker-A). This is designed for correcting possible tracking drift due to, e.g. similar color distributed object / background clutter, or partial object occlusions/intersections. Enhancement is done by assigning the mean shift tracker to an area that is also agreeable with that from the local feature points of target.

Areas used for the mean shift and for the candidate object: To limit the number of background pixels entering the foreground region, one may use a slightly smaller ellipse area inside the rectangular box (e.g., scaled by K, K=0.9 in our tests) of candidate object region $R^{(2)}$. This is based on the observation that an ellipse box can be tighter for some object types and would therefore exclude the background pixels around the four corners of the rectangular box.

The result from the local feature-based tracker (i.e. baseline tracker-A) is employed to guide the kernel position of mean shift, if the result from the baseline tracker-A is shown to be reliable. This is done by examining the Bhattacharyya coefficient and the number of consensus feature points in the tracked object region $R_t^{(1)}$. If they are both high (indicating a reliable result), then the initial parameter vector in $R_t^{(2)}$ for the baseline tracker-B is assigned by that in the tracker-A, otherwise by the parameter vector from the tracker-B at $t-1$, i.e.

$$V_t^{(2)} = \begin{cases} V_t^{(1)} & \text{if } |\mathcal{P}^F| > T_1^{(2)} \text{ and } \rho_t^{(1)} > T_2^{(2)} \\ V_{t-1}^{(2)} & \text{otherwise} \end{cases} \tag{12}$$

where $T_1^{(2)}$ and $T_2^{(2)}$ are thresholds determined empirically.

4.2.4 Re-initializing the region

A re-initialization process is added to tackle the issue of tracking drift or tracking error propagation across frames. The idea can be analogue to applying intra video coding in each fixed frame interval. The following criterion, in a similar spirit to that in the baseline tracker-A,

is used to determine whether or not the re-initialization is applied to the tracked region $\hat{R}_t^{(2)}$ for the baseline tracker-B. That is, if one of the following two conditions is satisfied:

$$dist_t^{(2)} = \sum_{i=1}^{4} \|\mathbf{x}_{t,i}^{(2)} - \mathbf{x}_{t-1,i}^{(2)}\|^2 > T_3^{(2)}, \text{ or } \rho_t^{(2)} < T_4^{(2)} \tag{13}$$

then, the baseline tracker-B is re-initialized at t by:

$$R_t^{(2)} \leftarrow R_{t-1}^{(obj)}, \; V_t^{(2)} \leftarrow V_{t-1}^{(obj)} \text{ and } \tilde{\rho}_t^{(2)} \leftarrow 0 \tag{14}$$

where $R_{t-1}^{(obj)}$ is the previous tracked bounding box from the final hybrid tracker at $(t-1)$ and $V_{t-1}^{(obj)}$ is the corresponding parameter vector, $\tilde{\rho}_t^{(2)}$ is the normalized Bhattacharyya coefficient defined in (21), $\mathbf{x}_{t,i}^{(2)}$ and $\mathbf{x}_{t-1,i}^{(2)}$ are the four corners of tracked object regions $R_t^{(2)}$ and $R_{t-1}^{(2)}$ from the baseline tracker-B, $T_3^{(2)}$ and $T_4^{(2)}$ are two empirically determined thresholds.

5. Online learning in the spatiotemporal SIFT-RANSAC and the enhanced anisotropic mean shift

Online learning is an important step that may significant impact the tracking performance. In this section, we describe two online learning techniques, one is utilized to maintain two feature point sets from the spatiotemporal SIFT-RANSAC in the baseline tracker-A, another is applied to update the reference object distribution from the enhanced anisotropic mean shift in the baseline tracker-B.

5.1 Online learning of local point feature sets

This subsection describes a key step in the spatiotemporal SIFT-RANSAC: online learning of two feature point sets across video frames. The use of spatiotemporal SIFT-RANSAC is motivated by the problems encountered in the conventional SIFT-RANSAC for tracking objects through long video sequences. Often, the number of corresponding feature points reduce significantly through video frames due to object pose changes, partial occlusions or intersections. In these cases, some feature points on the object may disappear, consequently only a subset of feature points finds their correspondences across two image frames. This phenomenon may propagate through video frames, and may eventually lead to a dramatically reduced number of corresponding points. When the number of points is very small, the region surrounding these points may become very small and unreliable. Further, motion parameter estimation can become ill-posed if the number of equations is less than the unknown parameters.

The online learning procedure in the spatiotemporal SIFT-RANSAC is designed to dynamically maintain the corresponding point sets through video frames. This includes the possibility of adding new points, pruning weak points with low weights, and freezing the adaptation when partial occlusions are suspected. This online learning is applied separately to the foreground candidate object region and the surrounding background region. The method for online maintenance of spatiotemporal point correspondences is similar to the work in (Haner & Gu,10; Khan & Gu,10), which contains the following main steps:

- Assign weights to all corresponding feature points;
- Add new candidate feature points;

- Prune feature points with low weights;
- Freeze the updating when a partial occlusion of object is likely to occur;
- Separate maintenance of the foreground feature point set $\mathcal{P}^F$ and the background feature point set $\mathcal{P}^B$.

5.1.1 Maintenance of foreground feature point set $\mathcal{P}^F$

For this feature point set, online learning of the dynamic feature points contains the following main steps:

Assigning weights: First, a set of feature points $\mathcal{P}^F = \{p_i = (x_i, y_i),\ i = 1, 2, \cdots\}$ at the frame t is selected from the SIFT, within the transformed bounding box area obtained by using the estimated affine transform parameters from the RANSAC to the tracked object box at (t-1). Feature points from RANSAC that are outside the transformed bounding box (i.e. belong to the background) are then removed. After that, a candidate consensus point set $\mathcal{P}^F$ is created. $\mathcal{P}^F$ consists of three subsets of feature points according to how consensus correspondences in the foreground set are established,

$$\mathcal{P}^F = \{\mathcal{P}_a^F \cup \mathcal{P}_b^F \cup \mathcal{P}_c^F\} \tag{15}$$

In (15), $\mathcal{P}_a^F$ contains matched consensus points, i.e. the feature points selected by the RANSAC, $\mathcal{P}_b^F$ contains the outliers that fail to agree with the best estimated transform parameters in the RANSAC (which could be the result from noise influence or object dynamics), and $\mathcal{P}_c^F$ is the set of newly added feature points from the SIFT that are initiated within the candidate object region at t and do not correspond to any background feature points. Each feature point p_i in the candidate set $\mathcal{P}^F$ is assigned to a weight according to:

$$W_t^i = \begin{cases} W_{t-1}^i + 2 & p_i \in \mathcal{P}_a^F \\ W_{t-1}^i - 1 & p_i \in \mathcal{P}_b^F \\ W_0^i & p_i \in \mathcal{P}_c^F \end{cases} \tag{16}$$

where the initial weight for a new feature point W_0^i is set to be the median weight value of all points in the subsets $\mathcal{P}_a^F$ and $\mathcal{P}_b^F$, i.e. $W_0^j = \text{median}(W_t^j | p_j \in \mathcal{P}_a^F \cup \mathcal{P}_b^F)$, and W_{t-1}^i for $\mathcal{P}_a^F$ and $\mathcal{P}_b^F$ is initialized to zero in the first frame. Once the maximum consensus points are selected at t, their weights in (16) are increased. For those corresponding points that do not fit to the estimated transform parameters in the RANSAC (i.e. matched outliers), their weights in (16) are reduced.

Adding or pruning consensus points: After updating the weights in (16), feature points in $\mathcal{P}^F$ are then updated. This is done by first sorting out the feature points in $\mathcal{P}^F$ according to their weights. New feature points in $\mathcal{P}_c^F$ are added with the median weight values, so that they may remain in the set after the subsequent pruning process. The pruning process is then applied to keep a reasonable size of $\mathcal{P}^F$ by removing low weight feature points in the set. This is done by keeping the L_F (L_F empirically determined, L_F=1000 in our tests) highest weight points in the set and removing the remaining ones.

Freezing the updating when a partial occlusion is highly probable: If an object is occluded by cluttered background or intersected by other objects, object appearance within the bounding box may change significantly. The Bhattacharyya coefficient value may indicate the existence of such scenarios, as the images between the tracked object and the reference object become less similar. When such a scenario occurs, the online maintenance process should be temporally frozen in order to not include the feature points from the background clutter or

the occluding object. The Bhattacharyya coefficient is computed by using the histograms from the reference object and from the tracked object region $R^{(1)}$ at t obtained in RANSAC as $\rho_t^{(1)} = \sum_{u=1}^{m} \sqrt{p_u^{t,(1)}(y,\Sigma)\, q_u^t}$, where $p_u^{t,(1)}$ and q_u^t are the uth bin of spatial kernel-weighted color histogram from $R^{(1)}$ of tracker-A, and of the reference object region, respectively. The kernel center $y = (y_1, y_2)$ and the anisotropic kernel bandwidth matrix Σ can be computed using the method described in Section 4.2. The Bhattacharyya coefficient $\rho_t^{(1)}$ is used to indicate whether or not the region $R^{(1)}$ is likely to contain occluding objects/background area, e.g. from partial occlusions or object intersections. If $\rho_t^{(1)}$ is small, indicating that the global appearance of object is significantly different from that of the reference object, then the dynamic maintenance is temporally frozen, so that no new feature points would be wrongly added. This is done as follows: If $\rho_t^{(1)} \leq T_F$ (T_F is a threshold determined empirically), then the maintenance process freezes, otherwise the maintenance process proceeds.

5.1.2 Maintenance of background feature point set $\mathcal{P}^B$

Online learning is also performed to the background set to maintain the dynamic background feature points. The background set contains feature points in between the large searching box and the candidate object region (see the area between the black and red rectangles in Fig.2). Comparing with the foreground case, the following differences exist for maintaining the background set:

The searching area and its re-initialization: The search area at t (see the black rectangle in Fig.2) is a rectangular area, that is larger than the tracked object region $R_{t-1}^{(1)}$. This is done by extending both the left and right side of $R_{t-1}^{(1)}$ by $k_x w^{(1)}$ ($w^{(1)}$ is the width of $R_{t-1}^{(1)}$, and k_x=0.5 in our tests), and extending both the top and bottom side of $R_{t-1}^{(1)}$ by $k_y h^{(1)}$ ($h^{(1)}$ is the height of $R_{t-1}^{(1)}$, and $k_y \in [0.1, 0.5]$ in our tests). This results in a searching area of $(2k_x + 1)w^{(1)}$ in width, and $(2k_y + 1)h^{(1)}$ in height. Correspondingly, the search area is re-initialized immediately after the re-initialization of the foreground object region $R_t^{(1)}$.

Shift foreground points to the background set: Those feature points in the foreground set that find their correspondences in the background set are re-assigned to the background set.

Adding and pruning new background points: New feature points that are found in the background region at the current frame t are added into this set. A maximum number L_B is then assigned to the background point set (L_B=1500 in our tests, empirically determined). Feature points in this set are sorted out according to their aging: If the total number of feature points exceeds L_B, then only the newest L_B feature points are kept while the remaining old aging points are removed.

5.2 Online learning of dynamic reference object appearance

Since the appearance of a target object may change in time, using a time-independent appearance distribution $q^t = q$ for the reference object may lead to tracking drift or tracking failures in the mean shift especially when the pdf of a visual object changes (e.g. significant changes in the object color distribution). Despite much research work in the mean shift-based object tracking, online learning of reference object pdf q remains an open issue. This is mainly due to the ambiguity on the change which could be caused either by object itself or by some background interferences (e.g. occlusions/intersections or background clutter) and the lack of

occlusion priors. An efficient method for online learning of reference object pdf for the mean shift has so far been reported in (Khan & Gu,10).
The basic idea behind this online learning method is to *only* update the reference appearance distribution at those frames where reliable tracking without occlusion is indicated. Further, the updating frequency does not need to be very high, since object changes are usually gradual due to the mechanical movement. The online learning is done by using a criterion function and seeking the local maximum point that corresponds to good tracking performance in each individual frame interval of fixed length. Let $\rho_t = \sum_u \sqrt{q_u^{j-1} p_u^t}$ be the Bhattacharyya coefficient between the current tracked object from the final tracker and the reference object in the previous $(j-1)$th interval, and $\mathbf{x}_{t,i}$ be the four corners of the tracked region $R_t^{(obj)}$ from the final tracker. Noting that q_u^{j-1} implies that q_u^t is in the $(j-1)$th interval, $t \in [(j-2)S+1, (j-1)S]$, where S is the total frames in the interval (S is empirically determined depending on the motion speed and video frame rate, S=25 frames in our tests). If the following conditions are both satisfied:

$$\text{dist}_t = \sum_{i=1}^{4} \|\mathbf{x}_{t,i} - \mathbf{x}_{t-1,i}\|^2 < T_1, \quad \text{and} \quad \rho_t > T_2 \tag{17}$$

then the reference object appearance distribution q^j in the jth interval is updated

$$q^j = \kappa p^{t^*} + (1-\kappa) q^{j-1} \tag{18}$$

where $j = 1, 2, \cdots$, and t^* is the highest performance frame chosen from

$$t^* = \text{argmax}_{t \in [(j-1)S+1, jS]} \rho_t \tag{19}$$

and q^j is the updated reference object pdf in the jth interval that is related to the time interval $t \in [(j-1)S+1, jS]$, κ is the constant controlling the learning rate ($\kappa = 0.1$ in our tests), p^{t^*} is the appearance distribution of the candidate object where t^* is chosen from (19). If (17) is not satisfied, then the reference object distribution remains unchanged, i.e., $q^j \leftarrow q^{j-1}$. Key steps for updating q^j in the jth interval can be summarized as:

1. Check whether conditions in (17) are satisfied in $t \in [(j-1)S+1, jS]$;
2. If satisfied, updating q^j using (18) where the frame t^* is selected from (19);
3. Otherwise, freezing the update by assigning $q^j \leftarrow q^{j-1}$.

As an example, Fig.4 shows the two performance curves in (17) for the video "stair walking", where the thresholds T_1 and T_2 (blue dash line) and the updated frames (red dots) are marked. To demonstrate the effect of online learning, Fig.5 shows the results from the hybrid tracker with and without adding online learning to the reference object appearance. It shows that the improvement of tracking performance is most visible with the increase of video frame number. Since object appearance changes gradually in time, online learning of dynamic reference object distribution has indeed yielded visible improvement in tracking.

6. Hybrid tracker formulated from a criterion function

This section describes the formulation of hybrid tracker through combining the two baseline trackers under a given criterion function.
For the baseline tracker-A in Section 4.1, feature point correspondences are estimated by using spatiotemporal SIFT-RANSAC in the foreground and background regions. A tight rectangle

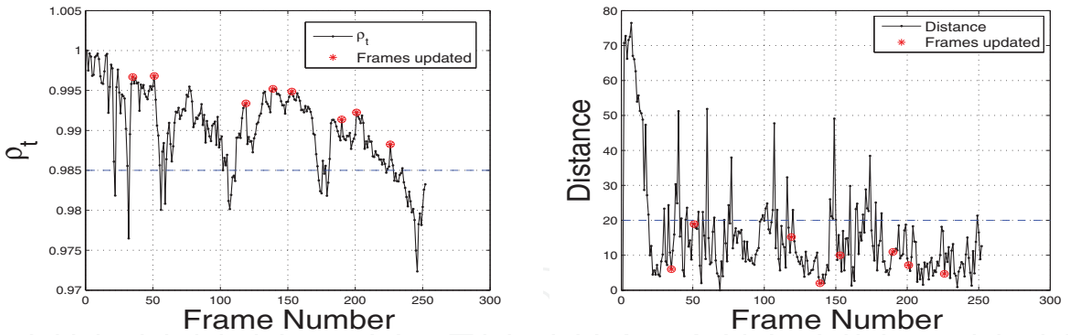


Fig. 4. Online learning of q^t for video "stair walking". Left: the curve of Bhattacharyya coefficient ρ_t in (17) vs. frame number, where the blue dash line is the threshold T_2, and the red dots are the frames updated. Right: the curve of distance of four corners $dist_t$ in (17) vs. frame number, where the blue dash line is T_1 and the corresponding updated frames are in red dots.

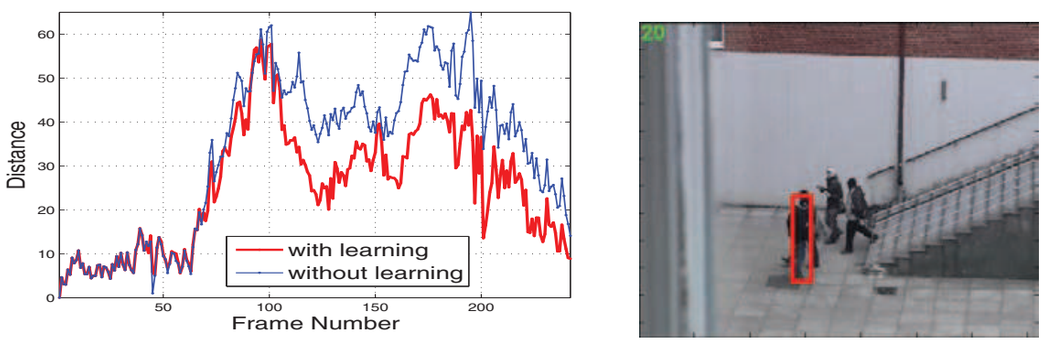


Fig. 5. Left: Tracking errors for the hybrid tracker as a function of video frame number. Distance d_1 in the curve is defined between the tracked and the ground-truth regions, according to (25). Red: with online learning; Blue: without online learning. Both curves are for the "stair walking" video; Right: an example frame of "stair walking".

surrounding the foreground points is then drawn as the candidate object region $R_t^{(1)}$ that is described by a parameter vector,

$$R_t^{(1)} : V_t^{(1)} = [y_c = (y_{1,c}^{(1)}, y_{2,c}^{(1)}), w^{(1)}, h^{(1)}, \theta^{(1)}, \mathcal{P}^F]_t^T$$

containing the 2D center position, width, height and orientation of the bounding box, and the foreground point set. For the baseline tracker-B in Section 4.2, an image region $R_t^{(2)}$ whose image content is most similar to the reference object appearance is sought by using the enhanced anisotropic mean shift with its kernel partially guided by the local feature points. This enhanced mean shift tracker generates a parameterized candidate region

$$R_t^{(2)} : V_t^{(2)} = [y = (y_1^{(2)}, y_2^{(2)}), w^{(2)}, h^{(2)}, \theta^{(2)}, \mathbf{h}_{rgb}^{(2)}]_t^T$$

A third candidate object region $R_t^{(3)}$ is then formed whose parameter vector is a weighted combination of the parameter vectors of the above two baseline trackers, i.e.,

$$R_t^{(3)} : \quad V_t^{(3)} = \sum_{i=1}^{2} \tilde{\rho}_t^{(i)} V_t^{(i)} \tag{20}$$

where $\rho_t^{(i)}$ is the Bhattacharyya coefficient (defined in (23)), and $\tilde{\rho}_t^{(i)}$ is the normalized Bhattacharyya coefficients for the two baseline trackers,

$$\tilde{\rho}_t^{(i)} = \frac{\rho_t^{(i)}}{\rho_t^{(1)} + \rho_t^{(2)}} \tag{21}$$

For the final hybrid tracker, the parameter vector associated with the optimal target object region $R_t^{(obj)}$ is selected by maximizing the following criterion,

$$V_t^{(obj)} = \arg\max_{i:V_t^{(i)}} \left\{ \rho_t^{(i)}, i = 1, \cdots, 3 \right\} \tag{22}$$

where $\rho_t^{(i)}$, i=1,2,3, is the Bhattacharyya coefficient measuring the similarity between the reference object and the candidate object from the tracked candidate region $R_t^{(i)}$ at time t,

$$\rho_t^{(i)} = \sum_{u=1}^{m} \sqrt{q_u^t \; p_u^{t,(i)}} \tag{23}$$

$p_u^{t,(i)}$ is the uth bin of candidate object pdf estimate either from the baseline tracker-A (i=1) or from the baseline tracker-B (i=2), q_u^t is the uth bin of reference object pdf estimate. Noting that the superscript t in q_u^t indicates that the reference object pdf is dynamic. Table 1 summarizes the algorithm of the entire hybrid tracking scheme.

Initialization::
Frame $t = 0$: mark a bounding box for the object and compute q_0;
For frame $t = 1, 2, \cdots$, do:
1. *Baseline tracker-A: Local feature correspondences by the spatiotemporal SIFT-RANSAC.*
1.1 Compute correspondence points by SIFT in the searching area;
1.2 Find consensus points, estimate the transform, compute scores by RANSAC;
1.3 Perform dynamic point maintenance to $\mathcal{P}_F$ and $\mathcal{P}_B$;
1.4 Compute $V_t^{(1)}$, $R_t^{(1)}$, and $\rho_t^{(1)}$;
1.5 If (3) is satisfied, re-initialize $V_t^{(1)}$, $R_t^{(1)}$, and $\tilde{\rho}_t^{(1)}$;
2. *Baseline tracker-B: Enhanced anisotropic mean shift:*
2.1 Using (12) to determine the initial $V^{(2)}$;
2.2 Compute $\hat{y}^{(t)}$ using (8), and $\hat{\Sigma}^{(t)}$ using (9);
2.3 Repeat Step 2.2 until convergence;
2.4 Compute $\hat{w}_t^{(2)}$, $\hat{h}_t^{(2)}$, $\hat{\theta}_t^{(2)}$ from (11), and form $V_t^{(2)}$;
2.5 Compute $\rho_t^{(2)}$ and form $R_t^{(2)}$;
2.6 If (13) is satisfied, re-initialize $V_t^{(2)}$, $R_t^{(2)}$ and $\tilde{\rho}_t^{(2)}$;
3. Compute the combined region parameters $V^{(3)}$ using (20);
4. Determine $R_t^{(obj)}$ for the hybrid tracker according to (22);
5. Online learning of object appearance pdf:
If $mod(t, S) = 0$ (i.e., boundary of an interval), then online learning of q^j using (18), if conditions in (17) are satisfied;
END (For)

Table 1. The algorithm for the hybrid tracking scheme

7. Application: Employ online learning in the joint mean shift and particle filter-based tracker

In this section, we show a new application example where the online learning approach in Section 5.2 is directly added to an existing joint anisotropic mean shift and particle filter-based (JMSPF) tracking scheme (Khan et al.,09), in order to further improve the tracking performance.

In the JMSPF tracking scheme, a particle filter is used to track the parameters of object shape (or, the bounding box of object), while the multi-mode anisotropic mean shift is embedded in the the particle filter through shifting its kernel location to the most similar object area and subsequently forming up the conditional probability based on the appearance distance metric. In such a way, PF weights are updated by using the likelihood obtained from the mean-shift, that re-distribute particles according to the distance metric through exploiting the most similar object appearance from the mean shift, rather than using the random sampling. This leads to more efficient utilizing of particles, hence a significantly reduction of the required number of particles: from $N_P = 800$ particles when the state vector contains both the shape and the appearance of object (Wang et al.,08), to N_P=15 particles in this JMSPF tracking scheme. Details of the JMSPF tracking scheme is referred to (Khan et al.,09).

Due to the lack of effective online learning methods, the JMSPF tracker in (Khan et al.,09) uses a time-invariant reference object appearance. Adding online learning of object appearance to the JMSPF tracker can be done by applying (18) in fixed-length frame intervals, with a small modification to include the superscript i for particles,

$$q^{j,i} = \kappa p^{t^*,i} + (1-\kappa) q^{j-1,i}, \quad i = 1, \cdots, N_P \tag{24}$$

if the both conditions in (17) are satisfied. Further, the re-initialization process in Section 4.2.4 can also be applied. A JMSPF tracking scheme after adding the online learning and re-initialization process can be shown schematically in Fig.6.

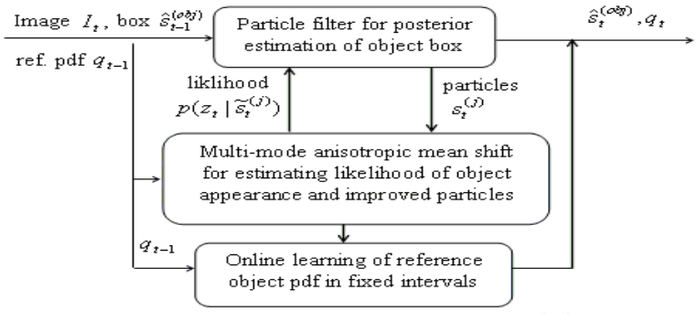


Fig. 6. Block diagram of the joint multi-mode anisotropic mean shift and particle filter-based tracking scheme (JMSPF) with online learning. Notations used in the block diagram: I_t: image frame at t; $\hat{s}_{t-1}^{(obj)}$ and $\hat{s}_t^{(obj)}$: tracked box parameters at (t-1) and t; $s_t^{(j)}$: jth particle at t; q_{t-1} and q_t: the estimated reference object pdf at (t-1) and t.

Fig.7 shows the tracking errors on the two videos "ThreePastShop2Cor" (CAVIAR Dataset) and "Pets2006_S07_C4" (PETS2006) with and without online learning. One can see that adding online learning to the scheme is able to further improve the tracking robustness and reduce the tracking drift in these cases.

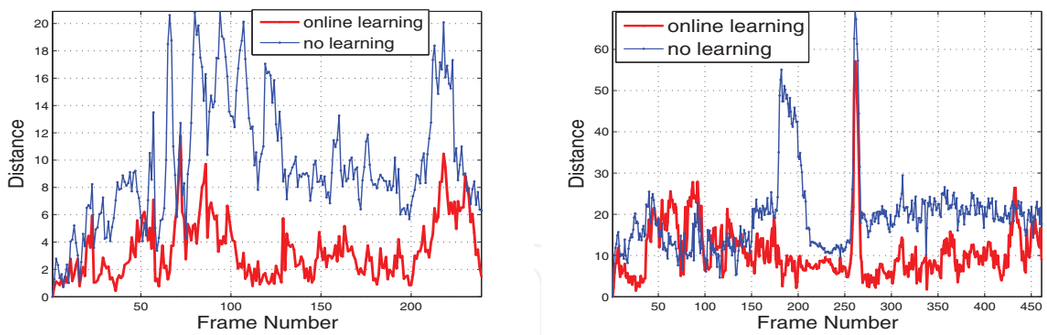


Fig. 7. Tracking errors for the JMSPF tracker: the distance d_1 in the curve is defined between the tracked and ground-truth regions according to (25). Red: with online learning; Blue: without online learning. Left: from the video "ThreePastShop2Cor"; Right: from the video "Pets2006_S07_C4". The ground truth boxes are manually marked. For "ThreePastShop2Cor", only the first 250 frames of ground truth boxes are marked and compared.

8. Experimental results and performance evaluation

The hybrid tracking scheme (summarized in Section 6) has been tested on numerous videos containing a range of complex scenarios. Our test results have shown that the hybrid tracking scheme is very robust and has yielded a marked improvement in terms of tracking drift, tightness and accuracy of tracked bounding boxes. This is especially obvious when complex scenarios in videos contain long-term partial occlusions, object intersections, severe object deformation, or cluttered background / background objects with similar color distributions to the foreground object.

8.1 Experimental setup

For testing the effectiveness of the hybrid tracking scheme, test videos that contain difficult scenarios in a range of complexities (e.g. long-term partial occlusion, object intersection, deformation or, pose changes) are selected. These videos are either captured from a dynamic or a static camera. In the tests, the initial bounding box is manually marked. Methods for automatic initial bounding box is beyond the scope of this chapter, readers can exploit other techniques, e.g. multiple hypothesis tests (Reid,79), active shape models or polygon vertices (Cootes et al.,01). For the mean shift tracking, a $32 \times 32 \times 32$ bin histogram is used for the RGB color images. The maximum number of iterations is 10 for the enhanced mean shift for all videos and is determined empirically. Table 2 summarizes the thresholds used for re-initialization thresholds as well as the γ values for normalizing the kernel bandwidth matrix of the mean shift in the hybrid tracker. $(T_1^{(2)}, T_2^{(2)})$ in (12) are set to (10, 0.95) in all cases. Further, Table 3 summarizes the online learning thresholds used for the hybrid tracker and the improved JMSPF tracker.

8.2 Qualitative evaluation and comparison of tracking results

The hybrid tracking scheme has been tested on numerous videos that contain a variety of difficult tracking scenarios. Fig.8 shows the tracking results (key video frames) from the hybrid scheme (marked by red solid line rectangles) on 5 videos. In all cases, online learning is included in the hybrid tracker.

Video	in baseline tracker-A		in baseline tracker-B		
	$T_1^{(1)}$	$T_2^{(1)}$	$T_3^{(2)}$	$T_4^{(2)}$	γ
walking lady	50	0.73	30	0.950	0.33
OneShopOneWait2Cor	300	0.65	15	0.975	0.33
ThreePastShop2Cor	60	0.98	15	0.975	0.33
Pets2006_S7_C4	150	0.87	12	0.975	0.31
Pets2007_DS5_C1	100	0.88	15	0.975	0.23

Table 2. Parameters in the hybrid tracker: re-initializing thresholds in (3) and (13), and γ-normalization in (8).

Video	Hybrid tracker		JMSPF tracker	
	T_1	T_2	T_1	T_2
walking lady	10	0.95	50	0.90
OneShopOneWait2Cor	30	0.95	20	0.90
ThreePastShop2Cor	30	0.96	15	0.96
Pets2006_S7_C4	30	0.95	20	0.95
Pets2007_DS5_C1	30	0.95	20	0.90

Table 3. Online learning thresholds in (17) for the hybrid tracker and the JMSPF tracker.

The video "walking lady" captured from a moving camera contains several long-term partial occlusions when the lady walks behind cars. Further, colors from a part of the object sometimes appear to be similar to the occluding car.
The video "OneShopOneWait2Cor" is downloaded from the CAVIAR dataset (CAVIAR Dataset). The selected target object is a walking man with dark colored clothes. During the course of walking, there is intersection where another man partially occludes the target man, also there are pose changes while the man is waiting, and scale changes during the course of walking.
The video "ThreePastShop2Cor" is also from the CAVIAR dataset (CAVIAR Dataset). In the video, the selected target (a man wearing a red coat with a backpack) walks in parallel with two other persons before intersecting with one of them by suddenly changing his walking direction. The man continues his walking and passes another nearby person with a red coat coming from the opposite direction. After a while, several other intersections appear when the man walks continuously away from the camera (depth changes).
The video "Pets2006_S7_C4" is from the Pets 2006 dataset (PETS2006), named "Dataset S7 (Take 6-B)" by the camera 4. The selected target object is a walking man with dark clothes. During the course of walking, there are several intersections with partial occlusions, pose changes. The man also passes over other walking persons with similar color clothes.
The video "Pets2007_S05_C1" is from the Pets 2007 dataset (PETS2007), named "Dataset S5" from the camera one. The video ia probably captured around a check-in desk in an airport, where there are many walking persons. The selected target object is a man with white shirt carrying a backpack. Tracking this single target through the crowds (containing around 20 persons where some are moving, some stand still) is a rather challenging task, as there are many and frequent partial occlusions, intersections and pose changes.
The aim of these tests is to qualitatively evaluate the robustness of the hybrid tracking scheme, especially in video scenes containing long term partial occlusions, object intersections, deformations and fast motion, cluttered background or background object.
Comparisons: Comparisons are made with three state-of-the-art methods that are closely-related to the hybrid tracker described in this chapter. They are:

Fig. 8. Comparing tracking results from 3 trackers: the hybrid tracker (red solid line box), *Tracker-1* (green short dash line); *Tracker-2* (blue long dash line). From rows 1-6: results for videos (key frames) "walking lady", "OneShopOneWait2Cor", "ThreePastShop2Cor", "Pets2006_S5_C4" and "Pets2007_DS5_C1".

- *Tracker-1:* an anisotropic mean shift tracker in (Sumin & Xianwu,08) that is formed entirely based on using global object modeling.
- *Tracker-2:* a spatiotemporal SIFT-RANSAC tracker that is entirely based on using local feature points.
- *Tracker-3:* a fragments-based tracker that uses integral histograms (Adam et al.,06).

Since we do not have the program code of *Tracker-3*, comparison is made by using the same videos shown in the *Tracker-3* (Adam et al.,06). Three videos (rows 1-3 in Fig.8) in (Adam et al.,06) are found by the Internet search, and are used for comparisons with *Tracker-3*. Fig.8 shows the tracking results from the three tracking methods: the hybrid tracker (marked in red solid line boxes), *Tracker-1* (marked in green short dash line boxes), and *Tracker-2* (marked in blue long dash line boxes). Observing the tracked results in Fig.8, the hybrid tracker is shown to be very robust with tightly tracked bounding boxes and without tracking drift. Comparing the results from the hybrid tracker and the two existing *Tracker-1* and *Tracker-2*, the hybrid tracking scheme is shown to be much more robust with marked improvement, especially in difficult video scenarios that contain long partial occlusions, object intersects, fast object motions, nearby persons/cluttered background with similar color distributions and shape. For the video "Pets2007_S05_C1", the hybrid tracker has eventually failed after 490 frames as the scenes contain too many persons (around 15) with high frequency of partial occlusions. Comparing with *Tracker-1* and *Tracker-2*, the two trackers have failed in

about 400 and 240 frames, respectively, while the hybrid tracker has managed to track the object somewhat longer.
Fig.9 shows the tracking results in (Adam et al.,06) (referred to as: *Tracker-3*). Comparing the tracking results (marked in red box) shown in Fig.9 and the tracking results in Fig.8 (rows 1-3, marked in red box), one can see that the two trackers have somewhat similar tracking performance in these 3 videos, both have tracked the target object rather well. Comparisons using more complex videos, e.g., "Pets2007_DS5_C1" would probably be able to distinguish the performance differences of these 2 trackers, however, no tests are made as this would require to run the program of (Adam et al.,06).

Fig. 9. Results from *Tracker-3* (courtesy from (Adam et al.,06)): results from *Tracker-3* (Red); manually selected target (Pink). Top to bottom: frames from the videos "walking lady", "OneShopOneWait2Cor" and "ThreePastShop2Cor".

8.3 Quantitative evaluation and comparisons of performance

To quantitatively evaluate and compare the performance of the hybrid tracker and the two existing trackers (*Tracker-1* and *Tracker-2*), three distance metrics are used.

8.3.1 Distance metrics

The Euclidian distance: is defined between the four corners of the tracked object bounding box and the manually marked ground truth box as follows,

$$d_1 = \frac{1}{4}\sum_{i=1}^{4}\sqrt{(x_{i,1}-x_{i,1}^{GT})^2+(x_{i,2}-x_{i,2}^{GT})^2} \tag{25}$$

where $(x_{i,1}, x_{i,2})$ and $(x_{i,}^{GT}, x_{i,2}^{GT})$, $i = 1, \cdots, 4$, are the corners of rectangular box from the final hybrid tracker and the manually marked Ground Truth (GT), respectively.

The MSE (Mean Square Error): is defined between the 5 parameters (2D center, width, height and orientation) of tracked object box and the manually marked Ground Truth (GT) object

bounding box over all frames in each video,

$$MSE = \frac{1}{N}\sum_{t=1}^{N}\sqrt{\left(v_i^t - v_i^{t,GT}\right)^2} \tag{26}$$

where v_i^t is the ith parameter of a tracked box at t, $v_i^{t,GT}$ is the ith ground truth parameter at t, N is the total number of frames in the video.

The Bhattacharyya distance: is defined between the tracked object box and the reference object box as follows:

$$d_2 = \sqrt{1 - \sum_u \rho(p_u, q_u)} \tag{27}$$

where u is the index of histogram bin. Under this criterion, good performance is indicated by small d_2 values. The average Bhattacharyya distance $\bar{d_2}$ is computed by averaging the Bhattacharyya distances over all frames in each video.

In the first row of Fig.10, we compare the tracking errors for the 3 trackers (the hybrid tracker, *Tracker-1* and *Tracker-2*), in terms of the Euclidian distance d_1 (in (25)) between the tracked box and the ground truth box as a function of image frames, on the video "face" and "walking lady". Comparing the results from the two videos, the hybrid tracker has clearly shown better performance than those from the *Tracker-1* and *Tracker-2* in these cases. In the 2nd row of Fig.10, we compare the hybrid tracker and the JMSPF tracker with online learning. Comparing the results from the two videos, the JMSPF tracker seems better in "ThreePastShop2Cor" and slightly worse in "walking lady" to that obtained from the hybrid tracker. The performance of these two methods varies depending on the test videos.
Table 4 shows the tracking errors (the MSEs defined in (26)) for the four trackers: the hybrid tracker, the JMSPF tracker, *Tracker-1*, and *Tracker-2*. Comparing the results in the table, the hybrid tracker and JMSPF tracker have shown clearly better performance than those from the two existing *Tracker-1* and *Tracker-2*. Further, the JMSPF tracker is shown to be much better than that of the hybrid tracker on the video "ThreePastShop2Corthe" and slightly worse on the video "walking lady".

Video	Box Parameters	Hybrid tracker	JMSPF tracker	*Tracker-1*	*Tracker-2*
walking lady	x-position	1.6851	2.9454	51.575	19.854
	y-position	1.6020	4.7661	66.222	10.357
	width w	0.8935	1.0221	4.7926	3.3857
	height h	1.1682	2.6063	2.5973	55.973
	θ (in radius)	0.0011	0.0047	0.0627	0.0123
ThreePastShop2Cor	x-position	2.8815	0.9940	23.835	4.8997
	y-position	3.3784	2.7138	37.878	7.6360
	width w	1.9675	1.0327	5.8228	2.4200
	height h	16.836	2.1415	9.8112	4.3763
	θ (in radius)	0.0067	0.0001	0.0023	0.0012

Table 4. Tracking errors, the MSE defined in (26), for 4 different trackers.

Table 5 shows the tracking errors, the average Bhattacharyya distances d_2 in (27), for the four trackers: the hybrid tracker, the JMSPF tracker, *Tracker-1* and *Tracker-2*.

Video	Hybrid Tracker	JMSPF Tracker	*Tracker-1*	*Tracker-2*
walking lady	0.2076	0.2652	0.3123	0.3654
OneShopOneWait2Cor	0.1466	0.2037	0.4549	0.5240
ThreePastShop2Cor	0.1580	0.1444	0.3241	0.2861
Pets2006_S7_C4	0.1007	0.2267	0.2473	0.2032
Pets2007_DS5_C1	0.1455	0.2133	0.2840	0.2370

Table 5. Tracking errors, the average Bhattacharyya distances $\bar{d}_2$ in (27), for 4 different trackers. The smaller the d_2, the better the performance.

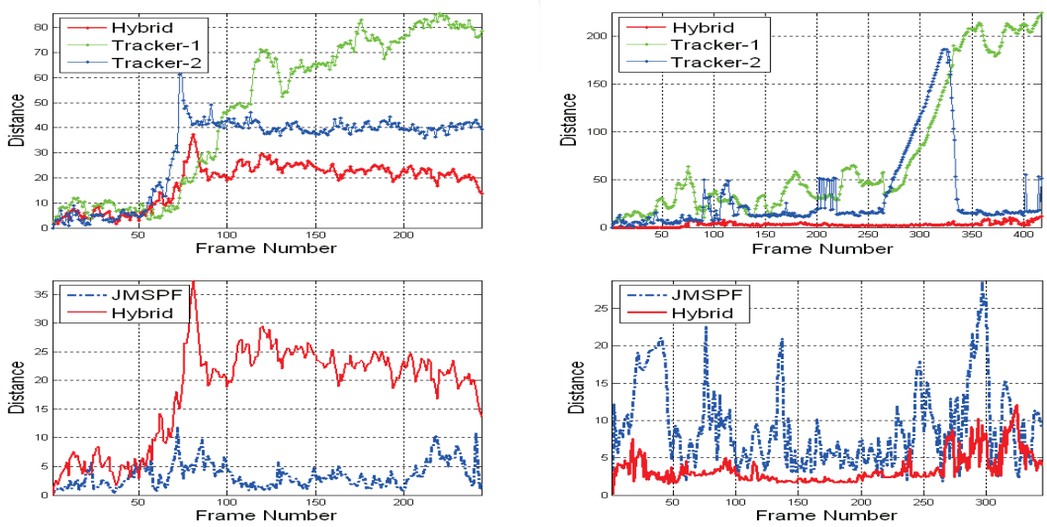


Fig. 10. Comparison of tracking errors (the Euclidian distance d_1 in (25)) on the videos "ThreePastShop2Cor" (column 1) and "walking lady" (column 2). 1st row: comparison among 3 trackers: hybrid tracker (red), *Tracker-1* (blu) and *Tracker-2* (green); 2nd row: comparison between hybrid tracker (red solid) and JMSPF tracker (blue dash). Noting the scale difference in vertical axis for "walking lady" in the 2nd column.

8.4 Computational cost

To give an indication on the computational cost, the execution times are recorded for four tracking methods: the hybrid tracker (summarized in Section 6), *Tracker-1*, *Tracker-2*, and JMSPF tracker (in Section 7). Table 6 shows the average time (in Second) required for tracking one object in one video frame, where the average is done over all frames in each video. Noting that tracking time varies dependent on the complexity of video scenes. All these tracking schemes are implemented by Matlab programs, and run on a PC with a Intel Pentium Dual 2.00 GHz processor. Observing Table 6 one can see that the hybrid tracker requires

Video	Hybrid tracker (in sec)	*Tracker-1* (in sec)	*Tracker-2* (in sec)	JMSPF tracker (in sec)
OneShopOneWait2Cor	0.1498	0.0934	0.049	1.4356
ThreePastShop2Cor	0.1501	0.0954	0.053	1.3945
Pets2006_S7_C4	0.1352	0.0935	0.041	1.5423
Pets2007_DS5_C1	0.1489	0.0845	0.050	0.9870

Table 6. Average required time to track a visual object in one video frame, for 4 different visual trackers. All programs are implemented in Matlab without optimizing the program codes.

more computations as comparing with *Tracker-1* or *Tracker-2*, as the result of combining baseline trackers, adding online learning and computing the criterion in order to make the final tracking. Despite this, the hybrid tracker achieves an average tracking speed of 10 frames/second using the Matlab program. One may also observe that the JMSPF tracker requires rather heavy computations, approximately 10 times of that required by the hybrid tracker.

8.5 Comparison between the hybrid tracker and the JMSPF tracker

Both tracking schemes, the hybrid tracker (summarized in Section 6) and the JMSPF tracker (in Section 7) are shown to be very robust in tracking visual objects. Fig.11 shows some results of tracked video frames (key frames are selected) on 5 videos from these two methods. Qualitatively evaluation of these tracking results through visual comparisons, both the hybrid tracker and the JMSPF tracker are shown to be very robust. From the general visual impression, the JMSPF tracker has a slightly better performance in terms of the tightness and the accuracy of the bounding box in some frames. For the video "Pets2007_S05_C1", similar to that in the hybrid tracker, the JMSPF tracker has eventually failed after about 490 frames as the scenes contain too many persons with high frequency of partial occlusions. Quantitatively evaluations of the performance by comparing d_2 values (defined in (27)) in Table 5 (columns 1 and 2) and d_1 values (defined in (25)) in Fig.10 (the right sub-figure), and comparing the computational speed in Table 6 (columns 1 and 4), show that the hybrid tracker has slightly smaller (average) d_2 values and a much fast computational speed (about 10 times faster) on the tested videos. While d_1 values in the two trackers vary depending on the videos. Overall, the hybrid tracker seems a more attractive choice, as the tradeoff between the average performance, tracking robustness and computational speed.

8.6 Limitations

Despite very robust tracking performance from the hybrid tracking scheme, several weak points are observed from the experiments. (a) If a target object in the video experiences a long-duration partial occlusions over a large percentage of area (e.g.$>$60%), then the tracking performance can be degraded, especially if the visible part is rather smooth and lacks of local feature points. (b) For images contain a relatively large object, e.g., a face, large pose changes could potentially cause tracking degradation. This is probably due to the complexity of face movement (many possible local motions) and the use of pdf as the face appearance model (that may not be the best choice). Improvement through using object-type-specific appearance models and feature point correspondences under multiple local motion models could be considered. (c) When full object occlusion occurs. Although our tests occasionally contain a few frames of full occlusion, it causes the tracker temporally frozen or tracking failure, however, the tracking is able to immediately recover or resume tracking soon after the partial appearance of the object. In principle, full occlusions with a long duration is beyond the limit of this scheme. The problem may be better tackled by trackers using videos from multiple cameras.

9. Conclusion

A novel hybrid visual tracking scheme is presented in this chapter, which jointly exploits local features and global appearances and shape of dynamic objects. The hybrid tracker is formulated using a criterion function that optimally combines the results from two baseline trackers: the spatiotemporal SIFT-RANSAC and the enhanced anisotropic mean shift.

Fig. 11. Comparing tracking results from 2 trackers: the hybrid tracker (red solid line box) and the JMSPF tracker with online learning (green dash line box). From rows 1-5: result frames (key frames) from videos "walking lady", "OneShopOneWait2Cor", "ThreePastShop2Cor", "Pets2006_S5_C4" and "Pets2007_DS5_C1".

Online learning of dynamic object is introduced to the global object appearance and local object feature points separately: For object appearances, online learning of the appearance distribution of reference object is performed in each fixed-length frame interval where the ambiguity between the object change and the change due to partial occlusions is addressed. For object feature points, online maintenance of two feature point sets (foreground and background) is performed in each video frame, where the background set is used as priors on the occlusion. It is worth noting that the online maintenance of feature point sets is a key step for the realization of spatiotemporal SIFT-RANSAC. It is also worth mentioning that the enhanced mean shift, by allowing the kernel position partially guided by local feature points, significantly reduces the mean shift sensitivity to similar color distributed background/other objects.

Experimental results on numerous videos with a range of complexities have shown that the hybrid tracker has yielded very robust tracking performance. This is especially evident when tracking objects through complex scenarios, for example, video scenes where the target object experiences long-term partial occlusions or intersections from other objects, large object deformations, abrupt motion changes of object, dynamic cluttered background/occluding objects having similar color distributions to the target object. Results of quantitative and qualitative evaluations and comparisons of the hybrid tracker and the two existing tracking methods, (*Tracker-1* and *Tracker-2*), have shown a marked tracking improvement from the hybrid tracker, in terms of reduced tracking drift and improved tightness of tracked object bounding box. Comparisons by visual inspecting the tracking results of 3 videos from the hybrid tracker and from (Adam et al.,06) have shown that both trackers perform rather well

in these cases. Further comparisons on complex video scenarios requires to run the program from (Adam et al.,06) and hence not performed. Comparisons of the hybrid tracker and the JMSPF tracker with online learning (in Section 7) have shown that latter has rather similar however occasionally slightly better performance but at the cost of significantly increase in computations (approximately 10 times). Comparisons of hybrid tracker with and without online learning have shown that adding online learning has significantly reduced the tracking drift especially for long video sequences. Overall, the hybrid tracking scheme is shown to be very robust and yielded marked improvements over the existing trackers (*Tracker-1* and *Tracker-2*). Comparing with the JMSPF tracker, the hybrid tracker provides a better tradeoff between the tracking robustness and tracking speed ($\approx$ 10 frames/second in our Matlab program).

10. References

[Adam et al.,06] Adam A, Rivlin E, Shimshoni I (2006), "Robust Fragments-based Tracking using the Integral Histogram", vol.1, pp.798-805, in Proc.*IEEE int'l conf CVPR*.

[Bay et al.,06] Bay H., Tuytelaars T. & Gool L.V.(2006), "SURF: Speeded Up Robust Features", in proc. *European conf. ECCV*, pp.404-417.

[Bauer et al.,07] Bauer J, Sunderhauf N & Protzel P (2007), "Comparing Several Implementations of Two Recently Published Feature Detectors", in proc. *Int'l Conf. Intelligent and Autonomous Systems, Toulouse, France*.

[Bar-Shalom & Fortmann,98] Bar-Shalom Y. and Fortmann T. (1998), Tracking and Data Association. New York: Academic.

[Battiato et al.,07] Battiato S., Gallo G., Puglisi G. & Scellato S. (2007), "SIFT Features Tracking for Video Stabilization", in *Proc of Int'l Conf Image Analysis and Processing*, pp. 825-830.

[Bretzner & Lindeberg,98] Bretzner L. & Lindeberg T. (1998), "Feature Tracking with Automatic Selection of Spatial Scales", in proc. *Comp. Vision and Image Understanding*, vol. 71, pp. 385-392.

[CAVIAR Dataset] http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/

[Chen et al.,08] Chen A, Zhu M, Wang Y & Xue C. (2008), "Mean shift tracking combining SIFT", in proc. *Int'l conf. Audio, Language and Image Proc.*, pp.1532-1535.

[Cootes et al.,01] Cootes TF, Edwards GJ, Taylor CJ (2001), "Active appearance models", *IEEE trans. TPAMI*, vol.23, no.6, pp.681Ű685.

[Collins,03] Collins R.T.(2003), "Mean-shift blob tracking through scale space", in proc. *IEEE Int'l conf. CVPR'03*, vol. 2, pp. 234-240.

[Comaniciu et al.,03] Comaniciu D., Ramesh V. & Meer P. (2003), "Kernel-based object tracking", *IEEE Trans Pattern Analysis and Machine Intelligence*, Vol.5, pp.564-577.

[Deguchi et al.,04] Deguchi K., Kawanaka O. & Okatani T. (2004), "Object tracking by the mean-shift of regional color distribution combined with the particle-filter algorithm", in proc. *Int'l conf. ICPR*, vol. 3, pp. 506-509.

[Fischler & Bolles,81] Fischler MA & Bolles RC (1981), "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartographys", in *Communications of the ACM*, vol. 24, pp. 381-395.

[Gordon et al.,01] Gordon N.J., Doucet A. and Freitas N.D. (2001), "Sequential Monte Carlo Methods in Practice", *New York: Springer*.

[Gordon,00] Gordon N.J., Doucet A. and de Freitas N. (2000), "On sequential monte carlo sampling methods for Bayesian filtering", *Statistics and Computing*, vol. 10, pp. 197-208.

[Haner & Gu,10] Haner S and Gu IYH (2010), "Combining Foreground / Background Feature Points and Anisotropic Mean Shift For Enhanced Visual Object Tracking", in proc. *IEEE Int'l conf. ICPR*, 23-26 August, Istanbul, Turkey.

[Harris & Stephens,88] Harris C. & Stephens M.(1988), "A Combined Corner and Edge Detector", in *Proc. 4th Alvey Vision Conf., Manchester*, pp.147-151.

[Hager et al.,04] Hager G.D., Dewan M., Stewart C.V. (2004), ŤMultiple Kernel Tracking with SSD,Ť vol. 1, pp.790-797, in proc. *IEEE Int'l conf. CVPR'04*.

[Khan & Gu,10] Khan, Z.H.; Gu, I.Y.H.(2010), "Joint Feature Correspondences and Appearance Similarity for Robust Visual Object Tracking", *IEEE Trans. Information Forensics and Security*, Vol.5, No. 3, pp. 591-606.

[Khan et al.,09] Khan ZH, Gu IYH, Backhouse AG (2010), "Robust Visual Object Tracking using Multi-Mode Anisotropic Mean Shift and Particle Filters", to appear, IEEE trans. Circuits and Systems for Video Technology.

[Khalid et al.,05] Khalid M.S., Ilyas M.U., Mahmoo K., Sarfaraz M.S., Malik M.B.(2005), "Kullback-Leiber divergence measure in correlation of gray-scale objects", in Proc. *2nd IntŠl conf. on innovations in Information Technology (IIT05)*.

[Li et al,06] Li. Y., Yang J., Wu R. & Gong F. (2006), "Efficient Object Tracking Based on Local Invariant Features", in *Proc. of Int. Symposium on Comm. and Information Technologies (ISCIT)*, pp. 697-700.

[Li et al.,08] Li L, Huang W, Gu IYH, Luo R & Tian Q (2008), "An efficient sequential approach to tracking multiple objects through crowds for real-time intelligent CCTV systems", *IEEE trans. Systems, Man and Cybernetics*, part B, vol.38, No.5, pp.1254-1269.

[Lim et al.,04] Lim J, Ross D,Lin R-S, Yang M-H (2004), "Incremental learning for visual tracking", in Proc. *Int'l conf NIPS*.

[Lowe,04] Lowe D.G (2004), "Distinctive Image Features from Scale-Invariant Keypoints", *Int. Journal of Computer Vision*, vol.60, pp. 91-110.

[Maggio & Cavallaro,05] Maggio E. & Cavallaro A. (2005), "Multi-part target representation for colortracking", in proc. *IEEE Int'l conf. ICIP*, pp.729-732.

[Mondragon wt al.,07] Mondragon I.F., Campoy P., Correa J. F. , & Mejias L.(2007), "Visual Model Feature Tracking For UAV Control", in *Proc. Int'l Symposium Intelligent Signal Processing*, pp.1-6.

[Okuma et al.,04] Okuma K., Taleghani A., Freitas N., Little J.J. & Lowe D.G. (2004), "A boosted particle filter: multitarget detection and tracking", in Proc. *Int'l conf. ECCV*, pp.28-39.

[PETS2006] http://www.cvg.rdg.ac.uk/PETS2006/data.html

[PETS2007] http://pets2007.net/

[Parameswaran et al.,07] Parameswaran V, Ramesh V & Zoghlami I (2006), "Tunable kernels for racking", in proc. *IEEE Int'l Conf. CVPR*, pp.2179-2186.

[Reid,79] Reid D.B. (1979), "An algorithm for tracking multiple targets", *IEEE Trans. Autom. Control*, vol. 24, no. 2, pp. 843-854.

[Rosales & Sclaroff,99] Rosales R. and Sclaroff S. (1999), "3D trajectory recovery for tracking multiple objects and trajectory guided recognition of actions", in *proc. IEEE Int'l Conf. CVPR*, pp. 117-123.

[Sankaranarayanan et al.,08] Sankaranarayanan A.C., Veeraraghavan A., Chellappa R. (2008), "Object Detection, Tracking and Recognition for Multiple Smart Cameras", *Proc. of the IEEE*, Vol.96, No.10, pp. 1606-1624.

[Shi & Tomasi,94] Shi J. & Tomasi C. (2008), "Good features to track", in proc. *IEEE Int'l conf. CVPR*, pp. 593-600.

[Strandmark & Gu,09] Strandmark P. & Gu I.Y.H. (2009), "Joint Random Sample Consensus and Multiple Motion Models for Robust Video Tracking", in Springer *LNCS* Vol. 5575, pp. 450-459.

[Skrypnyk & Lowe,04] Skrypnyk I.& Lowe D.G.(2004), "Scene modelling, recognition and tracking with invariant image features", in *Proc. Int. Symposium Mixed and Augmented Reality (ISMAR)*, pp. 110-119, 2004.

[Sumin & Xianwu,08] Sumin Q. & Xianwu H. (2008), "Hand tracking and gesture gecogniton by anisotropic kernel mean shift", in proc. *IEEE Int'l. conf. NNSP*, vol. 25, pp. 581-585.

[Vermaak et al.,03] Vermaak J., Doucet A., Perez P. (2003), "Maintaining multimodality through mixture tracking", in Proc. *IEEE Int'l conf. ICCV*, pp.1110-1116.

[Wang et al.,07] Wang T, Gu IYH, Shi P (2007), "Object tracking using incremental 2D-PCA learning and ML estimation", in proc. *IEEE int'l conf. ICASSP*.

[Wang et al.,08] Wang T., Backhouse A.G., & Gu I.Y.H. (2008), "Online subspace learning on Grassmann manifold for moving object tracking in video", in proc.*IEEE int'l conf. ICASSP*.

[Wang et al.,08] Wang T., Gu I.Y.H., Backhouse A.G. and Shi P. (2008), "Face Tracking Using Rao-Blackwellized Particle Filter and Pose-Dependent Probabilistic PCA", in proc. *IEEE int'l conf ICIP*, San Diego, USA, Oct. 12-15.

[Welch & Bishop,97] Welch G. and Bishop G. (1997), "Scaat: incremental tracking with incomplete information", in proc. *24th Annual Conf. Comp. Graphics & Interactive Techniques*.

[Wu et al.,08] Wu P, Kong L, Zhao F & Li X(2008), "Particle filter tracking based on color and SIFT features", in Proc. *IEEE int'l conf Audio, Language and Image Processing*, pp. 932-937.

[Xu et al,08] Tu Q., Xu Y., & Zhou M. (2008), "Robust vehicle tracking based on Scale Invariant Feature Transform", in proc. *Int. Conf. Information and Automation (ICIA)*, pp. 86-90.

[Xu et al.,05] Xu D, Wang Y & An J (2005), "Applying a new spatial color histogram in mean-shift based tracking algorithm", in proc. *Int'l conf. Image and Vision Comp.* New Zealand.

[Yang et al.,04] Yang J, Zhang D, Frangi AF, Yang J-Y (2004), "Two-dimensional PCA: a new approach to appearance-based face representation and recognition", *IEEE Trans. PAMI*, vol.26, no.1, pp.131-137.

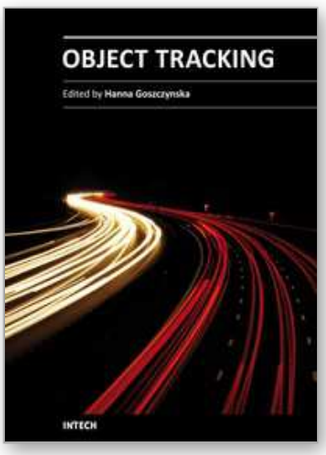
[Yilmaz,07] Yilmaz A., "Object tracking by asymmetric kernel mean shift with automatic scale and orientation selection", in Proc. *IEEE conf. CVPR'07*.

[Yilmaz et al.,06] Yilmaz A., Javed O. and Shah M.(2006), "Object tracking: A survey", *ACM Computing Surveys*, vol. 38, no. 4.

[Zhao et al.,08] Zhao C, Knight A & Reid I (2008), "Target tracking using mean-shift and affine structure", in Proc. *IEEE Int'l conf. ICPR*, pp.1-5.

[Zhou et al.,08] Zhou H., Yuan Y., Shi C.(2008), "Kernel-Based method for tracking objects with rotation and translation", *Int. Journal Computer Vision*.

[Zivkovic & Krose,04] Zivkovic Z. & Krose B. (2004), "An EM-like algorithm for color-histogram-based object tracking", in proc. *IEEE Int'l conf. CVPR*, vol.1, pp. I-798-808.



Object Tracking

Edited by Dr. Hanna Goszczynska

ISBN 978-953-307-360-6
Hard cover, 284 pages
Publisher InTech
Published online 28, February, 2011
Published in print edition February, 2011

Object tracking consists in estimation of trajectory of moving objects in the sequence of images. Automation of the computer object tracking is a difficult task. Dynamics of multiple parameters changes representing features and motion of the objects, and temporary partial or full occlusion of the tracked objects have to be considered. This monograph presents the development of object tracking algorithms, methods and systems. Both, state of the art of object tracking methods and also the new trends in research are described in this book. Fourteen chapters are split into two sections. Section 1 presents new theoretical ideas whereas Section 2 presents real-life applications. Despite the variety of topics contained in this monograph it constitutes a consisted knowledge in the field of computer object tracking. The intention of editor was to follow up the very quick progress in the developing of methods as well as extension of the application.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Irene Y.H. Gu and Zulfiqar H. Khan (2011). Online Learning and Robust Visual Tracking using Local Features and Global Appearances of Video Objects, Object Tracking, Dr. Hanna Goszczynska (Ed.), ISBN: 978-953-307-360-6, InTech, Available from: http://www.intechopen.com/books/object-tracking/online-learning-and-robust-visual-tracking-using-local-features-and-global-appearances-of-video-obje

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen