Collaborative part-based tracking using salient local predictors

Wassim Bouachir[a,*], Guillaume-Alexandre Bilodeau[a]

[a] *LITIV lab., Department of Computer and Software Engineering, École Polytechnique de Montréal, P.O. Box 6079, Station Centre-ville, Montréal (Québec), Canada, H3C 3A7*

Abstract

This work proposes a novel part-based method for visual object tracking. In our model, keypoints are considered as elementary predictors localizing the target in a collaborative search strategy. While numerous methods have been proposed in the model-free tracking literature, finding the most relevant features to track remains a challenging problem. To distinguish reliable features from outliers and bad predictors, we evaluate feature saliency comprising three factors: the *persistence*, the *spatial consistency*, and the *predictive power* of a local feature. Saliency information is learned during tracking to be exploited in several algorithm components: local prediction, global localization, model update, and scale change estimation. By encoding the object structure via the spatial layout of the most salient features, the proposed method is able to accomplish successful tracking in difficult real life situations such as long-term occlusion, presence of distractors, and background clutter. The proposed method shows its robustness on challenging public video sequences, outperforming significantly recent state-of-the-art trackers. Our Salient Collaborating Features Tracker (SCFT) also demonstrated a high accuracy even if a few local features are available.

Keywords: Part-based tracking, Feature saliency, keypoint, SIFT, keypoint layout.

---

*Corresponding author

Email addresses: `wassim.bouachir@polymtl.ca` (Wassim Bouachir), `gabilodeau@polymtl.ca` (Guillaume-Alexandre Bilodeau)

1. Introduction

Visual object tracking is a fundamental problem in computer vision with a wide range of applications including automated video monitoring systems [1, 2], traffic monitoring [3, 4], human action recognition [5], robot perception [6], etc. While significant progress has been made in designing sophisticated appearance models and effective target search methods, *model-free* tracking remains a difficult problem receiving a great interest. With *model-free* trackers, the only information available on the target appearance is the bounding box region in the first video frame. Tracking is thus a challenging task due to (1) the insufficient amount of information on object appearance, (2) the inaccuracy in distinguishing the target from the background, and (3) the target appearance change during tracking.

In this paper, we present a novel part-based tracker handling the aforementioned difficulties, including the lack of information on object appearance and features. This work demonstrates that an efficient way to maximize the knowledge on object appearance is to evaluate the tracked features. To achieve robust tracking in unconstrained environments, our Salient Collaborating Features Tracker (**SCFT**) discovers the most salient local features in an online manner. Every tracked local feature is considered as an elementary predictor having an individual reliability in encoding an object structural constraint, and collaborating with other features to predict the target state. To assess the reliability of a given feature, we define feature saliency as comprising three factors: *persistence*, *spatial consistency*, and *predictive power*. Thereby, the global target state prediction arises from the aggregation of all the local predictions considering individual feature saliency properties. Furthermore, the appearance change problem (which is a major issue causing drift [7]) is handled through a dynamic target model that continuously incorporates new structural properties while removing non-persistent features.

Generally, a tracking algorithm includes two main aspects: the target representation including the object characteristics, and the search strategy for object localization. The contributions of our work relate to both aspects. For target representation, our part-based model includes keypoint patches encoding object structural constraints with different levels of reliability. Part-based representations are proven to be robust to local appearance changes and partial occlusions [8, 9, 10]. Moreover, keypoint regions are more salient and stable than other types of patches (*e.g.* regular grid, random patches), increasing the distinctiveness of the appearance model [11, 12]. Regarding the search strategy, the target state estimation is carried out via local features collaboration. Every detected local feature casts a local prediction expressing a constraint on the target structure according to the spatial layout, saliency information, detection scale, and dominant orientation of the feature. In this manner, feature collaboration preserves the object structure while handling pose and scale change without requiring to analyze the relationship between keypoints like in [9], neither calculating homographies such as in most keypoint matching works [13, 14, 15].

More specifically, the main contributions of this paper are:

1. A novel method for evaluating feature saliency to identify the most reliable features based on their *persistence*, *spatial consistency*, and *predictive power*;
2. The explicit exploitation of feature saliency information in several algorithmic steps: (1) local predictions, (2) feature collaboration for global localization, (3) scale change estimation, and (4) for local feature removal from the target model;
3. A dynamic appearance model where persistent local features are stored in a pool, to encode both recent and old structural properties of the target.
4. Extensive experimentation to evaluate the tracker performance against five recent state-of-the-art methods. The experimental work conducted on challenging videos shows the validity of the proposed tracker, outperforming the compared methods significantly.

The rest of this paper is organized as follows. In the next section, we review related part-based tracking works. Algorithm steps are presented in details in section 3. Experimental results are provided and analyzed in section 4, and section 5 concludes the paper.

2. Related works

Among various visual tracking algorithms, part-based trackers have attracted a great interest during the last decade. This is mainly due to the robustness of part-based models in handling partial changes, and to the efficiency of prediction methods in finding the whole target region given a subset of object parts. The fragment-based tracker of Adam *et al.* [16] is one of the pioneering methods in this trend. In their tracker, target parts correspond to arbitrary patches voting for object positions and scales in a competitive manner. The object patches are extracted according to a regular grid, and thus are inappropriate for articulated objects and significant in-plane rotations. Further, Erdem *et al.* demonstrated that the winning patch might not always provide reliable predictions [17]. This issue is addressed in [17] by differentiating the object patches based on their reliability. Therefore, every patch contributes to the target state prediction according to its reliability, allowing to achieve a better accuracy. Many other methods have been proposed for locating the object through parts tracking. The authors in [18] track object parts separately and predict the target state as a combination of multiple measurements. This method identifies inconsistent measurements in order to eliminate the false ones in the integration process. The method in [19] represents the shape of an articulated object with a small number of rectangular regions, while the appearance is represented by the corresponding intensity histograms. Tracking is then performed by matching local intensity histograms and by adjusting the locations of the blocks. Note that these last two trackers present the disadvantage of requiring manual initialization of object parts.

In [10], the appearance model includes a combination between holistic and local representations to increase the model distinctiveness. In this model, the spatial information of the object patches is encoded by a histogram representing the object structure. Similarly, Jia *et al.* sample a set of overlapped patches on the tracked object [8]. Their tracker includes an occlusion handling module allowing to locate the object using only visible patches. Kwon *et al.* [20] also used a set of local patches, updated during tracking, for target representation. The common shortcoming of the last three trackers is the model adaptation mechanism in which the dictionary is updated simply by adding new elements, without adapting existing items. Another approach for creating part-based representations is the superpixel over-segmentation [21, 22]. In [21], Wang *et al.* use a discriminative method evaluating superpixels individually, in order to distinguish the target from the background and detect shape deformation and occlusion. Their tracker is limited to small displacements between consecutive frames, since over-segmentation is performed only for a region surrounding the target location in the last frame. Moreover, this method requires a training phase to learn superpixel features from the object and the background.

One of the major concerns in part-based tracking is to select the most significant and informative components for the appearance model. An interesting approach for defining informative components consists in using keypoint regions. Local keypoint regions (*e.g.* SIFT [23] and BRISK [24]) are more efficient than other types of patches in encoding object structure, as they correspond to salient and stable regions invariably detectable under various perturbation factors [25, 12]. Based on this, Yang *et al.* model the target with a combination of random patches and keypoints [26]. Keypoints layout is used to encode the structure while random patches model other appearance properties via their LBP features and RGB histograms. The target is thus tracked by exploiting multiple object characteristics, but the structural model captures only recent properties, as the keypoint model contains only those detected on the last frame. In a later work, Guo *et al.* [14] used a set of keypoint manifolds organized as a graph to represent the target structure. Every manifold contains a set of synthetic keypoint descriptors simulating possible variations of the original feature under viewpoint and scale change. The target is found by detecting keypoints on the current frame and matching them with those of the manifold model. This tracker achieved stable tracking of dynamic objects, at the cost of calculating homographies with RANSAC, which may be inappropriate for non-planar objects as shown in [9].

Generalized Hough Transform (GHT)-based approaches have been recently presented as an alternative to homography calculation methods. GHT was initially used in context tracking [27], where the target position is predicted by analyzing the whole scene (context) and identifying features (not belonging to the target) that move in a way that is statistically related to the target's motion. In later works, this technique has been applied to object features in order to reflect structural constraints of the target and cope with partial occlusion problems. Nebehay *et al.* [9] propose to combine votes of keypoints to predict the target center. Although every keypoint votes in an individual manner, the geometrical relationship is analyzed between each pair of keypoints in order to rotate and scale votes accordingly. Furthermore, the keypoint model is not adapted to object appearance changes, arising only from the first observation of the target. In [28], the authors used an adaptive feature reservoir updated online to learn keypoint properties during tracking. The tracker achieved robust tracking in situations of occlusion and against illumination and appearance changes. However, this method does not handle scale changes and suffers from sensitivity to large in-plane rotations. In this paper we propose a novel tracking algorithm that exploits the geometric constraints of salient local features in a way to handle perturbation factors related to the target movement (*e.g.* scale change, in-plane and out-of-plane rotations), as well as those originating from its environment (*i.e.* occlusion, background clutter, distractors).

3. Proposed method

3.1. Motivation and overview

In our part-based model, object parts correspond to keypoint patches detected during tracking and stored in a feature pool. The pool is initialized with the features detected on the bounding box region defined in the first video frame, and updated dynamically by including and/or removing features to reflect appearance changes. Instead of detecting local features in a region with a fixed size around the target location (like in [21, 14]), we eliminate the restriction of small displacements by using particle filtering to reduce the search space as proposed in [28]. This allows us to avoid computing local features on the entire image by limiting their extraction to most likely regions based on the target color distribution.

When performing target search on a given frame, features from the pool are matched with those detected on the reduced search space. Following the matching process, the geometrical constraints (of the matched features) are adapted to local scale and pose changes as explained in section 3.3.1. Then all the matched features collaborate in a voting-based method (section 3.3.2), to achieve global localization (section 3.3.3) and estimate the global scale change (section 3.3.4). Thus, the global prediction result corresponds to the aggregation of individual votes (elementary predictions). This method preserves the object structure and handles pose and scale changes, without requiring homography calculations such as in [14], neither analyzing the geometrical relationship between keypoints like in [9]. The figure 1 presents a visual summary of the main algorithm steps.

In order to keep the most relevant elements in the feature pool and exploit appropriately the most reliable predictors, each tracking iteration is followed by a saliency evaluation step. Saliency evaluation is performed to identify reliable features and determine the weights of their predictions accordingly, while eliminating irrelevant features from the appearance model. Our idea is inspired by the democratic integration framework of Triesch and von der Malsburg, where several cues contribute to a joint result with different levels of reliability [29]. In their approach, the elements that are consistent with the global result are considered as reliable and are assigned a higher weight in the future. This strategy has been adopted in other object tracking works

(a) (b) (c)

Figure 1: Visual illustration of the main algorithm steps when tracking a partly occluded face in a moderately crowded scene. **(a)**: the search space is reduced by using a color-based particle filter, and keypoints are detected in the limited region (green dots). **(b)**: matching the detected keypoints with the appearance model allows to identify those belonging to the target. **(c)**: matched features vote for the target center.

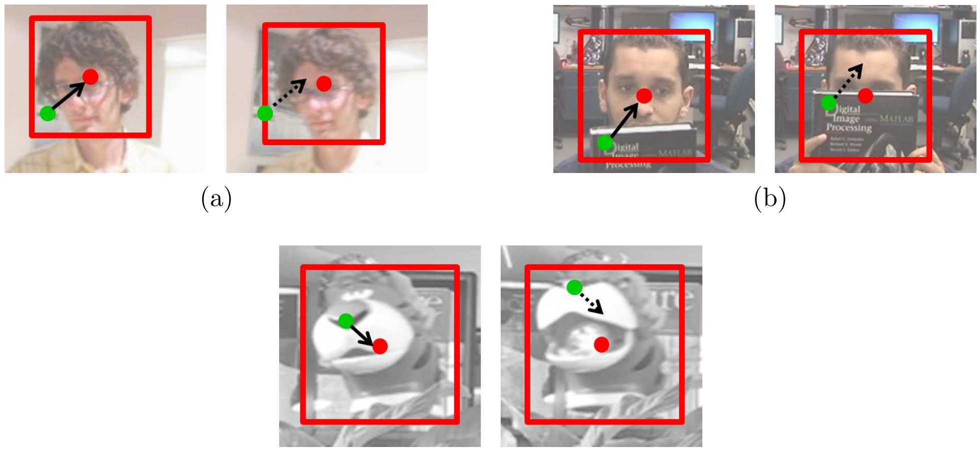
to perform an adaptive integration of cues according to their reliability [17, 30, 31]. In our tracking method, the reliability is defined by the feature saliency including three factors: feature *persistence*, *spatial consistency*, and *predictive power*.

- The *persistence* value ω of a given feature is used to evaluate the degree of co-occurrence between the target and the keypoint, and to determine if the feature should be removed from the pool.

- The *spatial consistency* matrix Σ reflects the motion correlation between the feature and the target center in the local prediction function.

- The *predictive power* ψ indicates the accuracy of the past local predictions by comparison to the past global predictions. This value is used to weight the contribution of a local feature in the global localization function.

Note that both the *spatial consistency* and the *predictive power* are designed to assess the feature quality. On the other hand, the *persistence value* is related to the occurrence level, disregarding the usefulness of the feature. Figure 2 illustrates situations where non-salient features can be identified through saliency evaluation. Non-salient features may correspond to outliers included erroneously to the object model in the initialization step or when updating it. Such a feature may originate from the background as seen



(a) (b) (c)

Figure 2: Typical situations showing that saliency evaluation allows identifying bad predictors. Red and green dots represent, respectively, the target center and the tracked feature. Continuous arrows represent the feature prediction initialization, while dotted arrows show inconsistent votes after a certain number of frames.

in figure 2a or belong to an occluding object (figure 2b) causing incorrect prediction. Once a keypoint is considered as non-salient, the corresponding local prediction (vote) will not be significant in the voting space, and/or its contribution will be reduced in the global localization procedure. Moreover the feature is likely to be removed from the pool as soon as it becomes *non-persistent*.

It should be noted that inconsistent features belonging to the tracked object may remain in the object model if they co-occur frequently with the target. An example is illustrated in figure 2c. However, their local predictions hardly affect the overall localization, since their quality indicators (Σ and ψ) will be reduced. While bad predictors are penalized and/or removed from the model, target global localization is carried out via a collaboration mechanism, exploiting the local predictions of the most salient features. The proposed tracking algorithm is presented in figure 3 and detailed in the next sections.

3.2. Part-based appearance model

In our tracker, the target is represented by a set of keypoint patches stored in a feature pool $\mathcal{P}$. The proposed method could use any type of

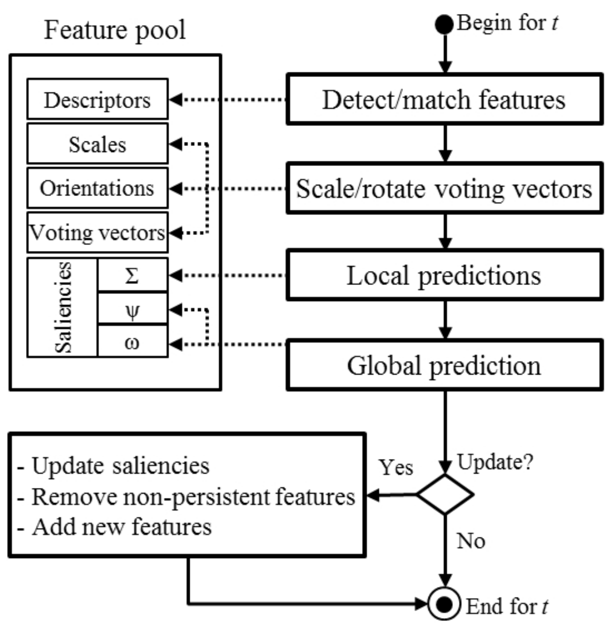


Figure 3: Diagram of the algorithm steps for a given frame at time t. Continuous arrows correspond to transitions between steps while dotted arrows show algorithm steps utilizing components from the appearance model.

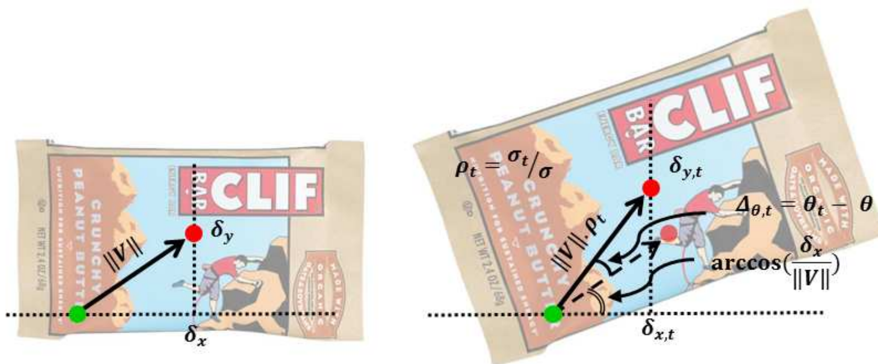


Figure 4: Adapting the voting vector to scale and orientation changes between the first detection frame of the feature (left) and the current frame (right). The red and green dots represent, respectively, the target center and the local feature.

scale/rotation invariant keypoint detector/descriptor. We used SIFT [23] as a keypoint detector/descriptor for its proven robustness [25]. We denote by f a feature from the pool $\mathcal{P}$. All the detected features are then stored under the form

$$f = [d, \theta, \sigma, V, Sal], \tag{1}$$

where:

- d is the SIFT keypoint descriptor comprising 128 elements to describe the gradient information around the keypoint position;
- θ is the detection angle corresponding to the main orientation of the keypoint;
- σ is the detection scale of the keypoint;
- $V = [\delta_x, \delta_y]$ is a voting vector describing the target center location with respect to the keypoint location (see figure 4);
- $Sal = [\omega, \Sigma, \psi]$ is the saliency information including *persistence*, *spatial consistency*, and *predictive power* indicators.

Note that all the detection properties (*i.e.* d, θ, σ, and V) are defined permanently the first time the feature is detected, whereas saliency information (*i.e.* ω, Σ, and ψ) is updated every time features are evaluated.

3.3. Global collaboration of local predictors

In order to limit keypoint detection at time t to the most likely image area, we apply the search space reduction method that we previously proposed in [28]. Detected features from the reduced search space are then matched with those in the target model $\mathcal{P}$ in a nearest neighbor fashion. For matching a pair of features, we require that the ratio of the Euclidian distance from the closest neighbor to the distance of the second closest is less than an upper limit λ. The resulting subset $\mathcal{F}_t \subseteq \mathcal{P}$ contains the matched target features at time t. After the matching process, the voting vectors (of the matched features) are adapted to local scale and pose changes as explained in the following.

3.3.1. Voting vectors adaptation

Each feature $f \in \mathcal{F}_t$ encodes a structural property expressed through its voting vector. Before applying the structural constraint of f, the corresponding voting vector V should be scaled and rotated according to the current detection scale σ_t and dominant orientation θ_t at time t as shown in figure 4. This adaptation process produces the current voting vector $V_t = [\delta_{x,t}, \delta_{y,t}]$, with

$$\delta_{x,t} = \|V\| \rho_t \cos(\Delta_{\theta,t} + \operatorname{sign}(\delta_y) \arccos \frac{\delta_x}{\|V\|}), \tag{2}$$

$$\delta_{y,t} = \|V\| \rho_t \sin(\Delta_{\theta,t} + \operatorname{sign}(\delta_y) \arccos \frac{\delta_x}{\|V\|}), \tag{3}$$

where $\Delta_{\theta,t}$ and ρ_t are respectively the orientation angle difference and the scale ratio between the first and the current detection of f:

$$\Delta_{\theta,t} = \theta_t - \theta, \tag{4}$$

$$\rho_t = {}^{\sigma_t}/_{\sigma}. \tag{5}$$

3.3.2. Local predictions

After adapting the voting vectors to the last local changes, we base local predictions on GHT to build a local likelihood (or prediction) map $\mathcal{M}_l$ for every feature in $\mathcal{F}_t$. For f, the local likelihood map is built in the reduced search space for all the potential object positions $\mathbf{x}$ using their relative positions $\mathbf{x}_f$ with respect to the keypoint location. The local likelihood map is defined using a 2D Gaussian probability density function as

$$\mathcal{M}_l(\mathbf{x}) = \frac{1}{\sqrt{2\pi|\Sigma|}}\ exp\,(-0.5\,(\mathbf{x}_f - V_t)^\top\,\Sigma^{-1}(\mathbf{x}_f - V_t)). \tag{6}$$

3.3.3. Global localization

To achieve global prediction of the target position, features in $\mathcal{F}_t$ collaborate according to their saliency properties (*persistence* and *predictive power*). The global localization map $\mathcal{M}_g$ is thus created at time t to represent the target center likelihood considering all the detected features. Concretely, the global map is computed by aggregating local maps according to the equation

$$\mathcal{M}_{g,t}(\mathbf{x}) = \sum_{f^{(i)} \in \mathcal{F}_t}^{i} \omega_t^{(i)} \psi_t^{(i)} \mathcal{M}_{l,t}^{(i)}(\mathbf{x}). \tag{7}$$

The final target location $\mathbf{x}_t^*$ is then found as

$$\mathbf{x}_t^* = \arg\max_{\mathbf{x}} \mathcal{M}_{g,t}(\mathbf{x}). \tag{8}$$

3.3.4. Estimating the scale

We also exploit saliency information to determine the target size S_t at time t. Scale change estimation is carried out by using the scale ratios of the most persistent keypoints. We denote by $\mathcal{F}_t^* \subset \mathcal{F}_t$ the subset including 50% of the elements in $\mathcal{F}_t$, having the highest value of ω_t. Then we compute

$$S_t = \frac{1}{|\mathcal{F}_t^*|} \sum_{f^{(j)} \in \mathcal{F}_t^*}^{j} \rho_t^{(j)} S^{(j)} \tag{9}$$

to estimate the current target size, taking into account the object size $S^{(j)}$ when the j^{th} feature was detected the first time.

3.4. Model update

The saliency information is updated with the object model when a good tracking is achieved. Our definition of a good tracking at time t is that the matching rate τ_t in the target region exceeds the minimum rate τ_{min}. In this case saliency indicators are adapted and $\mathcal{P}$ is updated by adding/removing features.

3.4.1. Persistence update

If the matching rate τ_t shows a good tracking quality, the *persistence* value $\omega_t^{(i)}$ is updated for the next iteration with

$$\omega_{t+1}^{(i)} = (1-\beta)\omega_t^{(i)} + \beta \mathbb{1}_{\{f^{(i)} \in \mathcal{F}_t\}}, \tag{10}$$

where β is an adaptation factor and $\mathbb{1}_{\{f^{(i)} \in \mathcal{F}_t\}}$ is an indicator function defined on $\mathcal{P}$ to indicate if $f^{(i)}$ belongs to $\mathcal{F}_t$. Following this update, we remove from $\mathcal{P}$ the elements having a *persistence* value lower than ω_{min}. On the other hand, the newly detected features (in the predicted target region) are added to $\mathcal{P}$ with an initial value ω_{init}.

3.4.2. Spatial consistency

The *spatial consistency* Σ is a 2x2 covariance matrix considered as a quality indicator and used in the local prediction function (Eq. 6). Σ is initialized to Σ_{init} for a new feature. It is then updated to determine the spatial consistency between $f^{(i)}$ and the target center by applying

$$\Sigma_{t+1}^{(i)} = (1-\beta)\Sigma_t^{(i)} + \beta\Sigma_{cur}^{(i)}, \tag{11}$$

where the current estimate of Σ is

$$\Sigma_{cur}^{(i)} = (V_{cur}^{(i)} - V_t^{(i)})(V_{cur}^{(i)} - V_t^{(i)})^\top, \tag{12}$$

and $V_{cur}^{(i)}$ is the offset vector measured at time t given the global localization result. As a result, Σ decreases for consistent features, causing the votes to be more concentrated in the local prediction map. By contrast, the more this value increases during tracking (for inconsistent features), the more the votes become scattered.

3.4.3. Predictive power

In this step, we evaluate the predictive power of every keypoint contributing to the current localization, considering the maxima of local prediction maps, and the global maximum corresponding to the final target position. This process, that we call *prediction back-evaluation*, aims to assess how good local predictions are. The local prediction for the i^{th} feature is defined as the position

$$\hat{\mathbf{x}}_t^{(i)} = \arg\max_{\mathbf{x}} \mathcal{M}_{l,t}^{(i)}(\mathbf{x}). \tag{13}$$

The *predictive power* $\psi_{t+1}^{(i)}$ of $f^{(i)}$ at time $t+1$ depends on the distances between its past predictions and the corresponding global predictions. We calculate $\psi_{t+1}^{(i)}$ with the summation of a fuzzy membership function as

$$\psi_{t+1}^{(i)} = \sum_{k=1}^{t} exp(\frac{-(\hat{\mathbf{x}}_k^{(i)} - \mathbf{x}_k^*)^2}{\epsilon S_k^2})\ \mathbb{1}_{\{f^{(i)} \in \mathcal{F}_k\}} \tag{14}$$

Algorithm 1 Tracking algorithm

- initialize $\mathcal{P}$
for all $frames$ **do**
    - Apply feature detector
    - Match features to get $\mathcal{F}_t \subseteq \mathcal{P}$
    for all $matched\_features$ ($f^{(i)} \in F_t$) **do**
        - Scale/rotate $V^{(i)}$: (Eq. 2 & 3)
        - Compute local likelihood map $\mathcal{M}_{l,t}^{(i)}(\mathbf{x})$: (Eq. 6)
        - Find local prediction result $\hat{\mathbf{x}}_t^{(i)}$: (Eq. 13)
    end for
    - Compute global likelihood map $\mathcal{M}_{g,t}(\mathbf{x})$: (Eq. 7)
    - Find global location $\mathbf{x}_t^*$: (Eq. 8) {output for frame t}
    - Estimate target size S_t: (Eq. 9) {output for frame t}
    if ($\tau_t \geq \tau_{min}$) **then**
        - Update ω_{t+1}: (Eq. 10)
        - Remove non-persistent features (*i.e.* $\omega_{t+1} \leq \omega_{min}$)
        for all $matched\_features$ ($f^{(i)} \in F_t$) **do**
            - update $\Sigma_{t+1}^{(i)}$ (Eq. 11) and $\psi_{t+1}^{(i)}$ (Eq. 14)
        end for
        - Add new features to $\mathcal{P}$
        - Initialize V, ω, Σ, and ψ for new features
    end if
end for

where ϵ is a constant set to 0.005. The *predictive power* ψ increases as long as the feature achieves good local predictions. Consequently, the feature is considered as a reliable predictor, and its contribution in the global localization function (Eq. 7) becomes more prominent. We note that both Σ and ψ are designed to evaluate the feature quality. However, the former affects local predictions while the latter weights its contribution in the global localization. The overall tracking algorithm steps are presented in Alg. 1.

4. Experiments

4.1. Experimental setup

4.1.1. The compared trackers

We evaluated our Salient Collaborating Features Tracker (**SCFT**) by a comparison to recent state-of-the-art algorithms. Among the compared trackers, four are part-based methods already discussed in section 2. These trackers are the SuperPixel Tracker (SPT) [21], the Sparsity-based Collaborative Model Tracker (SCMT) [10], the Adaptive Structural Tracker (AST) [8], and the Structure-Aware Tracker (SAT) [28]. The fifth one is the online Multiple Support Instance Tracker (MSIT) [32] using a holistic appearance model. The corresponding source codes are provided by the authors with several parameter combinations. In order to ensure a fair comparison, we tuned the parameters of their methods so that for every video sequence in our dataset, we always use the best parameter combination among the proposed ones.

4.1.2. Dataset

We evaluate the trackers on 20 challenging video sequences. Sixteen of them are from an object tracking benchmark commonly used by the community [33]. The four other sequences *jp1*, *jp2*, *wdesk*, and *wbook* were captured in our laboratory room using a Sony SNC-RZ50N camera. The area was cluttered with desks, chairs, and technical video equipment in the background. The video frames are 320x240 pixels recorded at 15 fps. We manually created the corresponding ground truths for *jp1*, *jp2*, *wdesk*, and *wbook* with 608, 229, 709, and 581 frames respectively [1]. Figure 5 presents the first frame of each of the sequences. In order to better figure out the quantitative results of our tracker, we categorized the video sequences according to the main difficulties that may occur in each sequence. The categorization of the sequences according to seven main properties is presented in table 1. This allows us to construct subsets of videos in order to quantitatively evaluate the trackers in several situations. Note that one video sequence may present more than one difficulty.

[1] Our sequences are available at http://www.polymtl.ca/litiv/en/vid/.

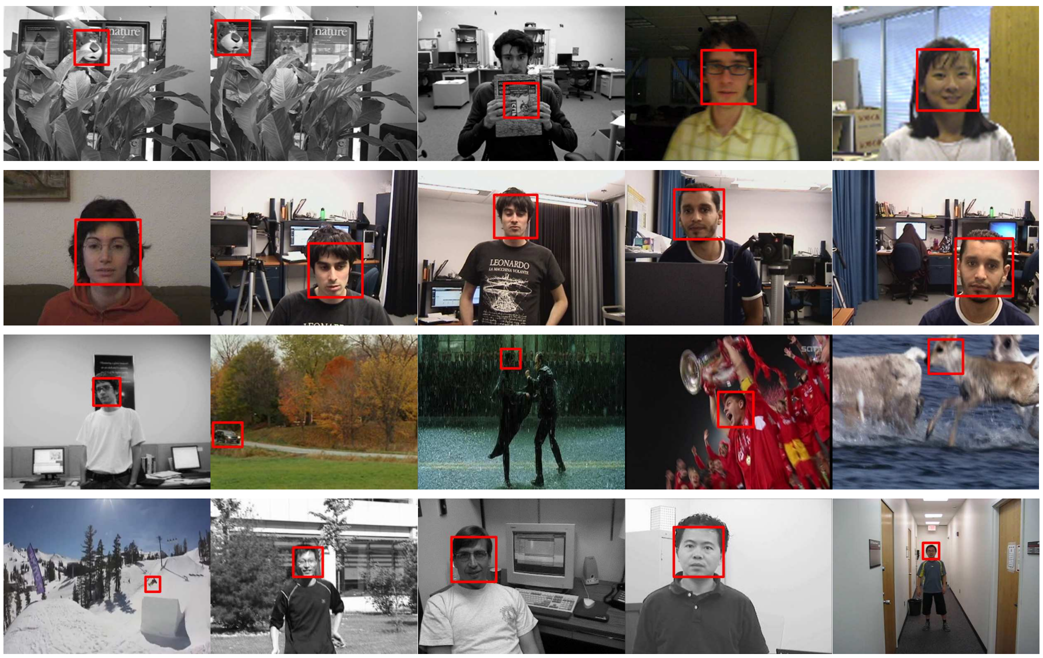


Figure 5: The annotated first frames of the video sequences used for experiments. From left to right, top to bottom: *tiger1*, *tiger2*, *cliffbar*, *David* , *girl*, *faceocc*, *jp1*, *jp2*, *wdesk*, *wbook*, *David2*, *car*, *matrix*, *soccer*, *deer*, *skiing*, *jumping*, *Dudek*, *Mhyang*, *boy*.

video	LTOcc	Distr	BClut	OPR	Illum	CamMo	ArtObj
David					✓	✓	
girl		✓		✓			
faceocc	✓					✓	
tiger1			✓				✓
tiger2			✓				✓
cliffbar			✓				
jp1		✓					
jp2		✓					
wdesk	✓						
wbook	✓						
David2				✓			
car						✓	
matrix	✓		✓		✓		
soccer	✓	✓	✓		✓	✓	
deer		✓				✓	
skiing						✓	✓
jumping						✓	
Dudek				✓		✓	
Mhyang				✓			
boy				✓		✓	

Table 1: Main difficulties characterizing the test sequences. LTocc: Long-Term Occlusion, Distr: presence of Distractors, BClut: Background Clutter, OPR: Out-of-Plane Rotation, Illum: Illumination change, CamMo: Camera Motion, ArtObj: Articulated Object.

4.1.3. Evaluation methodology

Success rate and average location error. In order to summarize a tracker's performance on a video sequence, we use the success rate and the average location error. The success rate is measured by calculating for each frame the Overlap Ratio $OR = \frac{area(P_r \cap G_r)}{area(P_r \cup G_r)}$, where P_r is the predicted target region and G_r is the ground truth target region. For a given frame, tracking is considered as a success if $OR \geq 0.5$. The Center Location Error (CLE) for a given frame consists in the position error between the center of the tracking result and that of the ground truth. The tables 2 and 3 present respectively the success rates and the average center location errors for the compared methods.

Precision plot. While the average location error is known to be useful to summarize performance by calculating the mean error over the whole video sequence, this metric may fail to correctly reflect the tracker behavior. For example, the average location error for a tracker that tracks an object accurately for almost all the sequence before losing it on the last frames could be substantially affected by large CLEs on the last few frames. To address this issue, we adopt the precision plot used in [34] and [35]. This graphic

video	SPT	SCMT	AST	MSIT	SAT	**SCFT**
David	62.37	60.22	37.63	*63.44*	**100**	**100**
girl	84.16	1.98	17.82	0.99	*84.95*	**85.94**
faceocc	5.62	**100**	25.84	80.90	99.55	*99.89*
tiger1	*60.56*	25.35	30.99	2.82	50.99	**80.28**
tiger2	46.27	16.42	31.34	5.97	*70.15*	**75.74**
cliffbar	51.52	24.24	*69.70*	7.58	60.30	**77.27**
jp1	18.09	78.13	84.38	3.78	*89.14*	**99.41**
jp2	39.30	55.02	55.02	16.59	*93.80*	**97.03**
wdesk	13.68	57.26	32.30	10.01	*90.47*	**93.96**
wbook	98.80	**100**	99.83	8.95	99.86	*99.90*
David2	36.44	90.69	38.55	94.23	*98.70*	**100**
car	*99.33*	87.33	92	57.33	*99.33*	**100**
matrix	3	6	1	2	**52**	**52**
soccer	16	31.33	36	37.33	**69.33**	**69.33**
deer	12.68	4.23	18.31	4.23	*95.77*	**100**
skiing	*58.33*	10	15	1.67	58.33	**96.67**
jumping	36.42	84.35	10.22	3.19	*95.53*	**99.04**
Dudek	**100**	**100**	**100**	79	**100**	**100**
Mhyang	85.67	77.67	94.67	**100**	**100**	**100**
boy	*99.33*	*99.33*	97.33	30	92	**99.67**
average	51.38	55.48	49.40	30.50	*85.01*	**91.31**

Table 2: Percentage of correctly tracked frames (success rate) for **SCFT** and the five other trackers. **Bold red** font indicates best results, *blue italics* font indicates second best.

shows the percentage of frames (precision) where the predicted target center is within the given threshold distance from the ground truth center.

Success plot. By analogy to the precision plot that shows percentages of frames corresponding to several threshold distances of the ground truth, the authors in [33] argue that using one success rate value at an overlap ratio of 0.5 may not be representative. As suggested in [33], we use the success plot showing the percentages of successful frames at the ORs varied from 0 to 1.

CLE and OR plots. Two other types of plots are used in our experiments to analyze in depth the compared methods : 1) the center location error versus the frame number presented in figure 6, and 2) the overlap ratio versus the frame number presented in figure 7. These plots are useful for monitoring and comparing the behaviors of several trackers over time for a given video sequence. We finally note that we averaged the results over five runs in all our experiments.

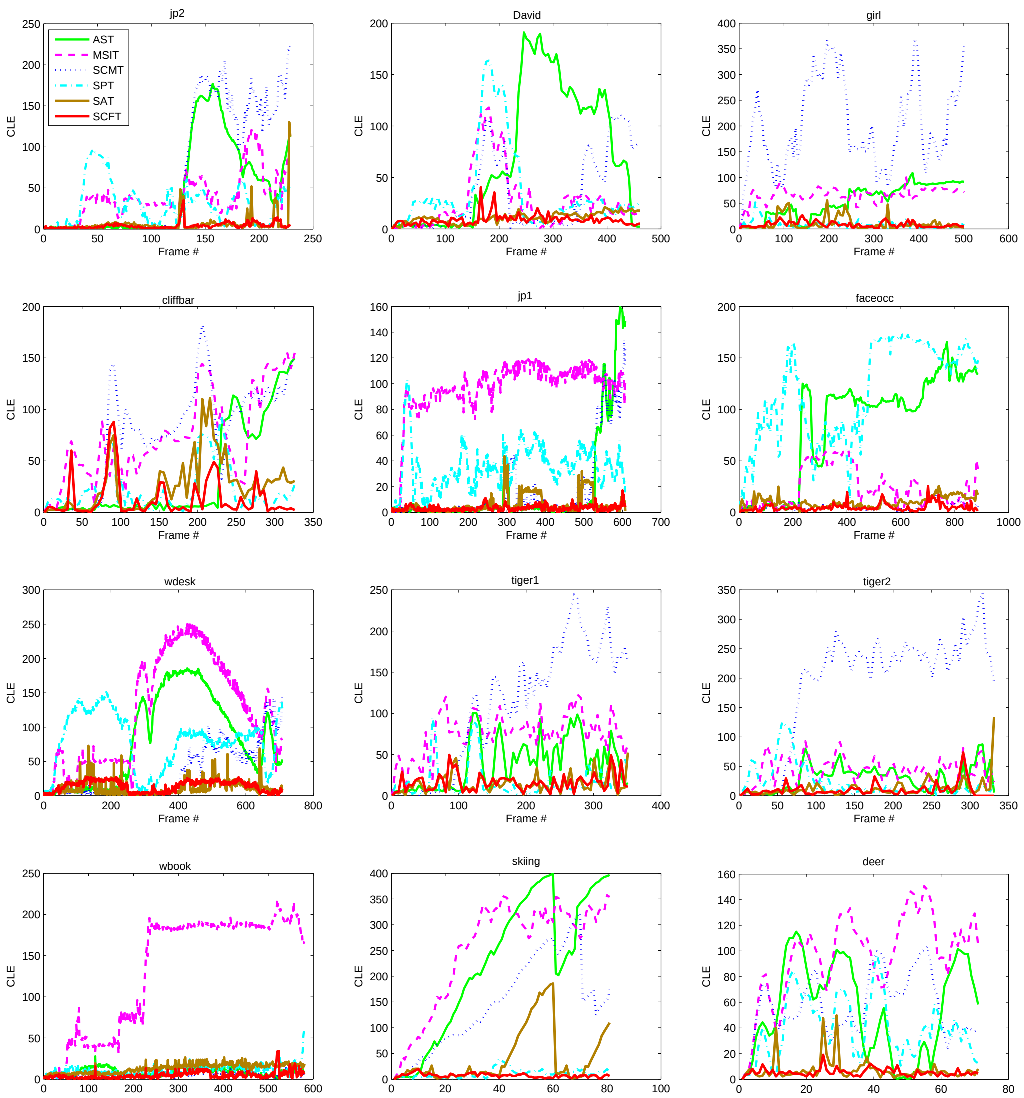


Figure 6: Center location error plots for 12 video sequences.

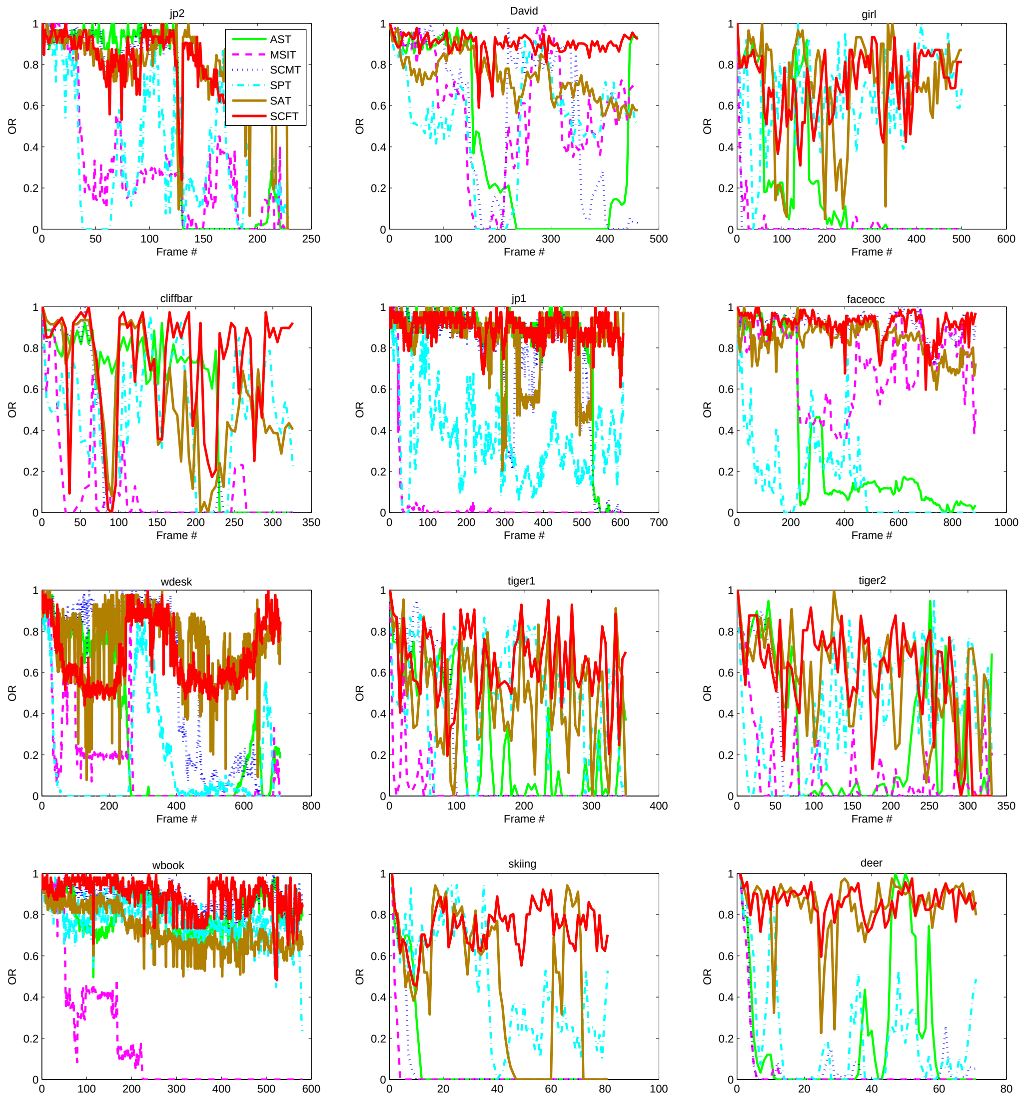


Figure 7: Overlap ratio plots for 12 video sequences.

video	SPT	SCMT	AST	MSIT	SAT	**SCFT**
David	36.09	33.81	68.57	26.71	*10.48*	**9.96**
girl	**8.97**	201.27	53.42	66.15	10.01	*9.29*
faceocc	116.84	**5.07**	85.43	23.36	14.26	*5.58*
tiger1	17.14	107.74	38.06	74.86	**14.91**	*15.65*
tiger2	22.81	189.50	29.15	44.58	*16.13*	**10.25**
cliffbar	*22.11*	77.31	35.35	73.72	25.33	**13.67**
jp1	35.21	17.74	16.66	97.08	*7.03*	**4.75**
jp2	30.58	69.44	45.15	39.47	*7.25*	**4.21**
wdesk	79.92	34.17	80.97	122.62	**11.12**	*14.31*
wbook	11.27	**5.09**	8.68	131.57	11.87	*5.91*
David2	39.74	4.12	9.18	*3.67*	5.68	**3.04**
car	6.65	6.98	*4.92*	34.67	6.16	**4.51**
matrix	43	79.87	57.74	74.82	**26.23**	**26.23**
soccer	35.46	87.91	58.29	32.18	**22.18**	*23.96*
deer	39.66	56.79	54.58	96.52	*7.42*	**5.39**
skiing	*9.83*	122.16	192.04	226.70	44.19	**7.75**
jumping	22.01	**7.41**	90.03	55.75	11.21	*8.15*
dudek	6.11	**4.28**	*4.74*	15.08	9.92	8.14
Mhyang	17.14	20.40	4.52	*2.49*	7.98	**2.31**
boy	*3.42*	**3.09**	3.97	43.65	7.09	7.42
average	30.20	56.71	47.07	64.28	*13.82*	**9.52**

Table 3: Average location errors in pixels for **SCFT** and the five other trackers. **Bold red** font indicates best results, *blue italics* font indicates second best.

4.2. Experimental result

4.2.1. Overall performance

The overall performance for several trackers is summarized by the average values in the tables 2 and 3 (last rows), as well as the average precision and success plots for the whole dataset (figure 8). All the metrics used for overall performance evaluation demonstrate that our proposed method outperforms all the other trackers, achieving an average success rate of 91.31% and an average localization error lower than 10 pixels. A major advantage of using success and precision plots is to allow choosing the appropriate tracker for a specific situation given the application requirements (*e.g.* high, medium, or low accuracy). In our experiments, the success and precision curves show the robustness of **SCFT** for all application requirements. **SCFT** is also the only tracker to reach 80% in precision for an error threshold of 15 pixels, and to produce a success rate exceeding 60% when the required OR is 80%. Except for SAT that realized the second best overall performance, and MSIT that had the last rank, the rankings of the other trackers are different depending on the considered metric. In the following subsections, the experimental results are discussed in details.

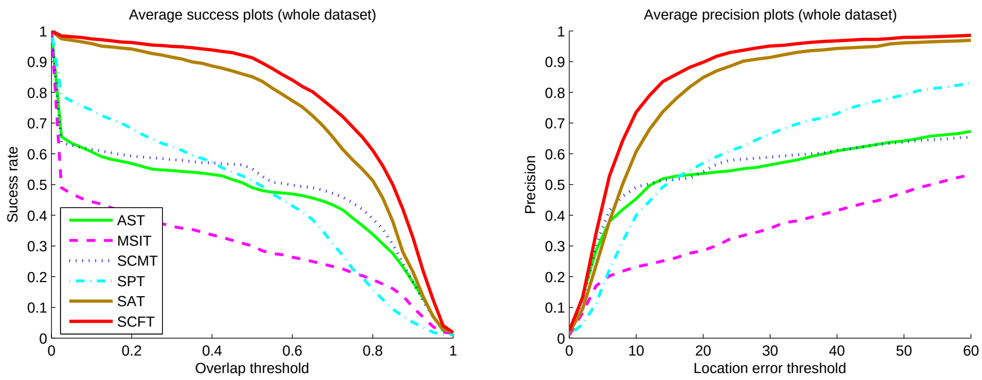


Figure 8: Average success and average precision plots for all the sequences.

4.2.2. Long-term occlusion

We evaluated the six methods in face tracking under long-term partial occlusion (up to 250 consecutive frames). In the *faceocc* and *wbook*, the tracked face remains partially occluded by an object several times for a long period. Some trackers drift away from the target to track the occluding object, which is mainly due to appearance model contamination by features belonging to the occluding object. Our method was able to track the faces successfully in almost all the frames under severe occlusion. The local predictions of a few detected features were sufficient for **SCFT** to achieve an accurate global prediction. Our target model may erroneously include features from the occluding object, but since we evaluate their motion consistency and predictive power, the corresponding local predictions will be scattered in the voting space and have small weights in the global localization function. The error plots for *faceocc* shows that SCMT and SAT also achieved good performances when the target was occluded (*e.g.* between frames 200 and 400). In fact, SCMT and SAT are also designed to handle occlusions, respectively through a scheme considering unoccluded patches, and a voting-based method that predicts the target center.

In the *wdesk* sequence, the tracked face undergoes severe partial occlusions while moving behind a desk. **SCFT**, SAT and SCMT track the target correctly until frame #400 where the person performs large displacements causing SCMT to drift away from the face. Both **SCFT** and SAT continue the tracking successfully while the tracked person hides behind a desk, and our method achieved the best success rate of 93.96%.

The success plots of long-term occlusion videos for **SCFT** and SAT show that both trackers can achieve more than 80% success rate as long as the required overlap ratio is lower than 0.5. Both trackers also had the two best precision curves, but **SCFT** performed significantly better under high requirement in accuracy (*i.e.* location error threshold lower than 15 pixels). As expected, the precision curve of MSIT is located below the others, since the holistic appearance model is not effective for a target undergoing severe occlusions.

4.2.3. Presence of distractors

The third and fourth rows of figure 10 present results of face tracking in moderately crowded scenes (four persons). In this experiments, our goal is to test the distinctiveness of the trackers. The success and precision plots for this category clearly show that **SCFT** and SAT are ranked respectively first and second regardless of the application requirements. This is mainly explained by the use of SIFT features that are proven to be effective in distinguishing a target face among a large number of other faces [36, 37, 38].

In the *jp1* video, we aim to track a face in presence of three other distracting faces, moving around the target and partially occluding it several times. The corresponding OR and CLE plots show that the proposed **SCFT** method produces the most stable tracking at the lowest error during almost all the 608 video frames. Although the success rates of 89.14%, 84.38%, and 78.13% respectively for SAT, AST, and SCMT indicate good performances, the last two trackers drift twice (first at frame#530 and a second time at frame #570) to track distracting faces occluding or neighboring the target. We can also see in the OR and CLE plots that SAT drifts considerably three times, especially between frames #341 and #397 when the tracked face region (person with a black t-shirt in the middle of the scene) is mostly occluded. However, neither the presence of similar objects near the target nor partial occlusion situations affected our **SCFT** tracker. The high performance of the proposed method in these situations is due to the distinctiveness of SIFT keypoints, in addition to the reliance on local predictions of the most salient features, even if outliers (from the background, neighboring or occluding faces) can be present in the feature pool.

In the *jp2* video, we track a walking person in the presence of four other randomly moving persons. The target crosses in front or behind distractors that may occlude it completely for a short period. All the five other methods confused the target with an occluding face, at least for a few frames after full

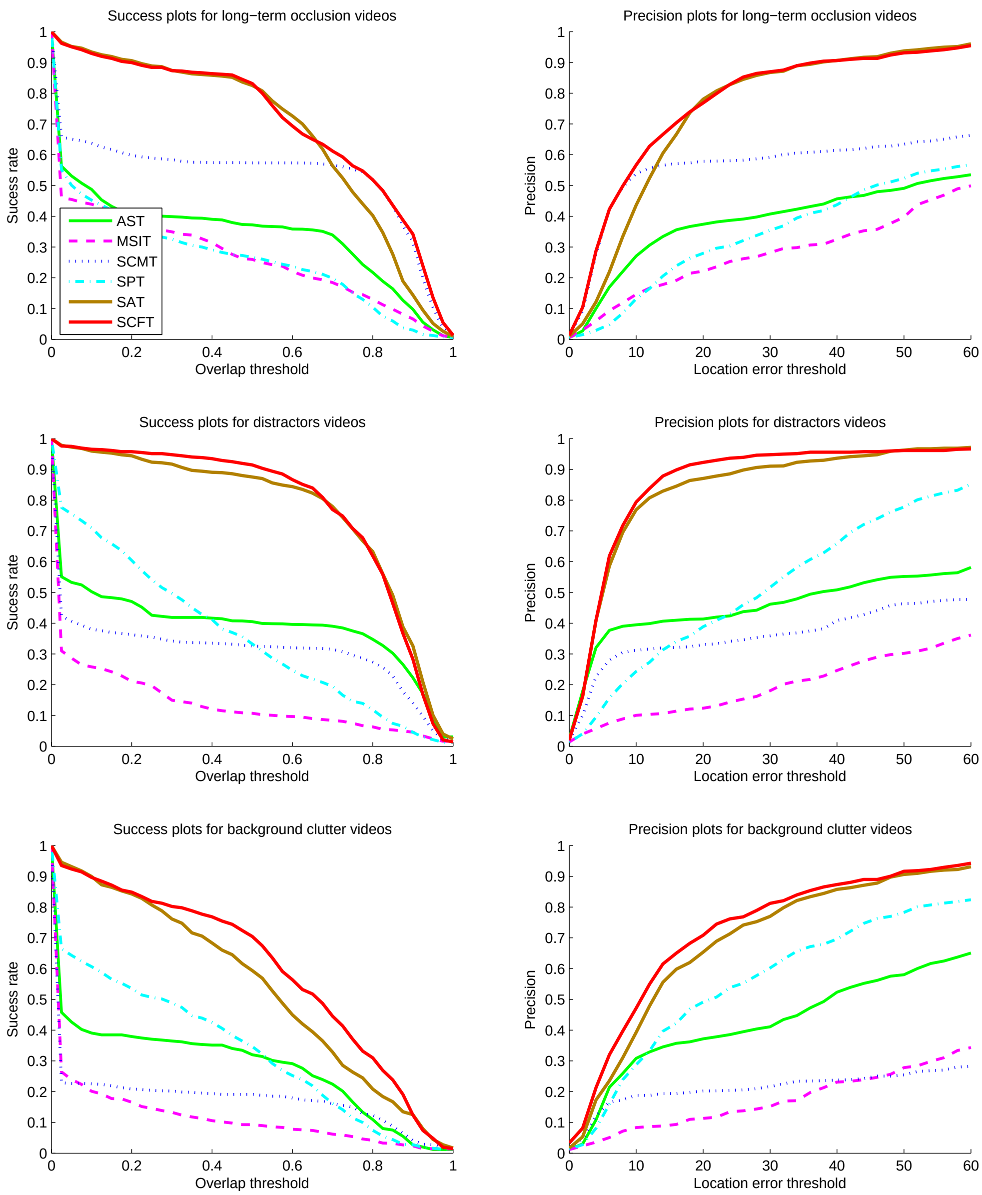


Figure 9: Success and precision plots for long-term occlusion, distractors, and background clutter videos.

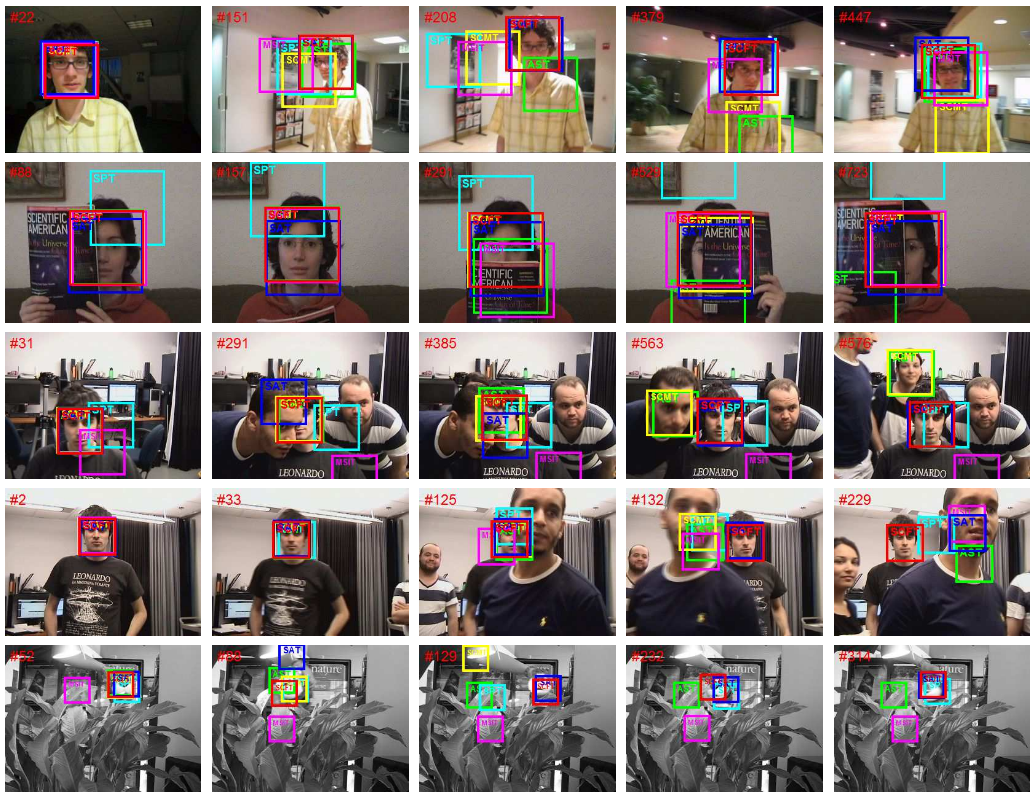


Figure 10: Tracking results for several trackers on the video sequences *David*, *faceocc*, *jp1*, *jp2*, and *tiger1* (from top to bottom).

occlusion. Nevertheless, **SCFT** is able to recover tracking correctly as soon as a small part of the target becomes visible. For both distractors sequences *jp1* and *jp2*, **SCFT** produced simultaneously the highest success rate and the lowest average error.

4.2.4. Illumination change, camera motion

The video sequence *David* is recorded using a moving camera, following a walking person. The scene illumination conditions change gradually as the person moves from a dark room to an illuminated area. The face also undergoes significant pose change during movement. All the trackers, except AST, were able to track the face successfully in more than 60% of the frames. Once again, **SCFT** achieved the best success rate and the lowest average error. This experiment shows the efficiency of our appearance model, allowing the tracker deal robustly with illumination variation. Our method is also not affected by large and continuous camera motion since features are detected wherever the space reduction method shows a significant likelihood of finding the target. On the other hand, in-plane rotations are handled efficiently in the global prediction function since we exploit the information on keypoint local orientation changes.

4.2.5. Out-of-plane rotation

The target person's face in the *girl* video, exhibits pose change and out-of-plane rotations abruptly. SPT, SAT, and **SCFT** were able to track the face correctly in more than 80% of the frames. **SCFT** achieved the best success rate, handling efficiently pose change and partial occlusion. Our tracking was accurate as long as the girl's face was at least partly visible. We lost the target when the face was turned away from the camera, but we were able to recover tracking quickly as soon as it partially reappeared.

4.2.6. Background clutter, articulated object

The main difficulty with the *cliffbar*, *tiger1*, and *tiger2* videos is the cluttered background whose the appearance may disrupt the tracker. For this category, the success and precision curves of **SCFT** are located above the others, showing the advantage of our method for all the tested thresholds of OR and CLE. Always based on the success and precision plots, we can see that SAT and SPT were ranked respectively second and third. It is noteworthy that both methods include discriminative aspects facilitating tracking under such conditions. In fact, SPT uses a discriminative appearance model based on superpixel segmentation while SAT utilizes information on the background color distribution to evaluate the tracking quality.

In the *Cliffbar* sequence, a book is used as a background having a similar texture to that of the target. **SCFT** outperformed significantly all the competing methods in both success rate and average location error. AST, SAT, and SPT also performed relatively well, taking into account the difficulty of the sequence. Indeed, the target undergoes abrupt in-plane rotations and drastic appearance change because of high motion blur. The proposed tracker is hardly affected by these difficulties since it continues adapting the appearance model by including/removing keypoints, and handling pose change through keypoint orientations.

In the *tiger1* and *tiger2* sequences, the target exhibits fast movements in a cluttered background with frequent occlusions. Owing to partial predictions that localize the target center using a few visible keypoints, **SCFT** had the highest percentages of correct tracks for both videos. SAT also overcomes the frequent occlusion problem via its voting mechanism that predicts the target position from available features. The other methods fail to locate the stuffed animal, but SPT had relatively better results due to its discriminative model facilitating the distinction between target superpixels and background superpixels. Note that the tracked object in *tiger1* and *tiger2* is a deformable stuffed animal. The predictions of features located on articulated parts are consequently inconsistent with the overall consensus, but this issue is effeciently handled by the use of *spatial consistency* and *predictive power* that reflect the predictors' reliability. These features may remain in $\mathcal{P}$ and continue predicting the target position without affecting the global result (because of low *predictive power* and *spatial consistency*). Our feature pool may also erroneously include outliers from the background, identified as non-persistent to be removed from the model.

4.2.7. Sensitivity to the number of features

One of the most challenging situations encountered in our dataset is the partial occlusion. The target faces in the *faceocc*, *wdesk*, and *wbook* videos undergo severe long-term occlusions causing the number of detected features to decrease drastically. Since local features detection represents a critical component for part-based trackers, we propose to study the impact of the number of features on SCFT's performance. We considered the video sequences *faceocc*, *wdesk*, and *wbook*, and analyzed the number of detected features on every video frame. We computed the average CLE value for each subset of frames having their numbers of collaborating features within the same interval (spanning 10 values). This allows us to create a scatter plot representing the average CLE versus the number of collaborating features (figure 11). To investigate the relationship between the number of features and the CLE, we model the plot by fitting a fourth degree predictor function and a linear function. The plot shows that the smallest numbers of features produce an average CLE not exceeding nine pixels. After that, the fitted fourth degree function decreases before stabilizing around the mean value of four pixels when more than 30 features are detected. Regarding the linear function ($y = ax + b$), it is obvious to expect that the coefficient a would be negative since the CLE becomes lower when the number of features increases. However, a high absolute value for a would suggest that the algorithm requires a large number of features to achieve accurate tracking. In our case,

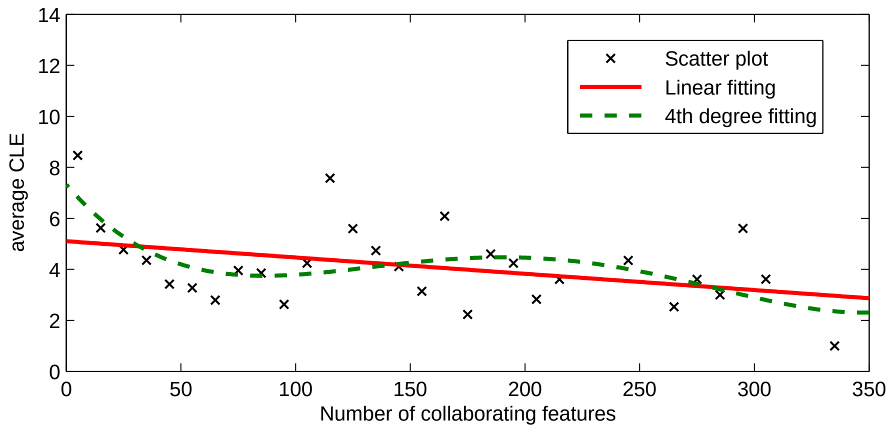


Figure 11: Sensitivity of SCFT's localization error (in pixels) to the number of collaborating features (sequences *faceocc*, *wdesk*, and *wbook*). Data points from the scatter plot correspond to interval centers.

the linear coefficients estimation ($a = -0.0064$; $b = 5.1107$) demonstrate that the error barely increases when the number of collaborating features diminishes from the maximum (*i.e.* 345 features) to one feature. This ascertainment confirms that the collaboration of a few number of unoccluded features is sufficient for our tracker to ensure accurate tracking.

4.2.8. Sensitivity to the saliency factors

In this section, we analyze the effect of the saliency factors separately on the tracking performance. We created three versions of **SCFT**:

- v-ω: the persistence indicator ω is not used in the global prediction function;
- v-ψ: the predictive power ψ is completely removed from the algorithm;
- v-Σ: the spatial consistency matrix is not updated, and is the same for all the features ($\Sigma = \Sigma_{init}$).

The tables 4 and 5 respectively present the percentages of correctly tracked frames and the average location errors for **SCFT** and the three other versions of the tracker on a subset of five video sequences. The selected sequences cover almost all the situations in table 1, and each video includes several

video	v-ω	v-ψ	v-Σ	SCFT
girl	43.56	56.44	*63.55*	**85.94**
tiger1	71.03	*78.87*	74.63	**80.28**
David2	89.20	95.51	*97.53*	**100**
deer	88.18	*92.52*	92.25	**100**
boy	80.22	*91.15*	88.06	**99.67**
average	74.44	82.90	*83.20*	**93.18**

Table 4: Percentage of correctly tracked frames for four versions of the proposed tracker. v-ω: the tracker do not use persistence indicators to weight local predictions, v-ψ: the tracker does not evaluate the predictive power of features, v-Σ: the spatial consistency matrix is the same for all the features. **Bold red** font indicates best results, *blue italics* font indicates second best.

difficulties. The obtained results show that the tracking performance is more affected when the persistence indicator is not considered (version v-ω). In fact,, v-ψ and v-Σ outperformed v-ω for all the five sequences. This result can be explained by the fact that with the removal of one factor among ψ and Σ, the remaining one continues to take into account the precision of the feature's past predictions, since both the spatial consistency and the predictive power are designed to assess the feature quality. However, if the indicator ω is not considered, the prediction step no longer takes into account the occurence level of the keypoint. Furthermore, these experiments demonstrated the complementarity of the three saliency factors, as the best performance is obtained when the three indicators are evaluated and updated during tracking. We finally note that the saliency evaluation method proposed in this work can be adapted or applied directly to a wide range of tracking algorithms that are based on the voting of local features.

4.2.9. Sensitivity to parameters

Most of the parameters of our algorithm were set to default values for all the video sequences. In our experimental work, only three parameters were tuned to optimize the performance of the tracker:

- N^* : the number of particles defining the reduced search space, where keypoints are detected;
- τ_{min} : the minimum matching rate that is required to update the appearance model;

video	v-ω	v-ψ	v-Σ	SCFT
girl	17.98	14.49	*13.24*	**9.29**
tiger1	17.02	*16.89*	16.98	**15.65**
David2	8.06	6.36	*5.11*	**3.04**
deer	10.19	8.13	*7.63*	**5.39**
boy	11.16	7.98	*7.51*	**7.42**
average	12.88	10.77	*10.09*	**8.16**

Table 5: Average location errors in pixels for four versions of the proposed tracker. v-ω: the tracker does not use persistence indicators to weight local predictions, v-ψ: the tracker does not evaluate the predictive power of features, v-Σ: the spatial consistency matrix is the same for all the features. **Bold red** font indicates best results, *blue italics* font indicates second best.

parameters	*girl*	*tiger1*	*David2*	*deer*	*boy*
N^*	30	100	100	20	50
τ_{min}	0.55	0.8	0.3	0.2	0.2
ω_{min}	0.3	0.4	0.1	0.4	0.4

Table 6: Parameter values used in **SCFT** with each video from the subset including *girl*, *tiger1*, *David2*, *deer*, and *boy*.

- ω_{min} : the persistence threshold used to determine if the feature should be removed from the model;

In order to evaluate the sensitivity of **SCFT** to parameters, we considered the same subset of five sequences and ran our tracker multiple times on each video, using the optimized parameters of the other videos. The optimized parameter values for each video are shown in table 6.

The results of these runs are reported in the tables 7 and 8, where the A.D. column shows the Average Difference between the result obtained with the optimized set of parameters and those obtained with the parameter sets of the four other sequences. As we can see, 13.33% is the most significant average decrease in sucess rate (for the *girl* video), while the highest average increase in localization error is that of the *David2* sequence (4.3 pixels). On the other hand, parameter change had a very low impact on the video sequences *deer* (1.41% as average decrease in sucess rate) and *boy* (1.30 pixels as average increase in localization error). In general, **SCFT** was able to achieve a stable tracking for all the runs and the performance of our tracker

	girl	*tiger1*	*David2*	*deer*	*boy*	A.D.
girl	**85.94**	81.19	75.26	72.58	61.41	13.33
tiger1	76.06	**80.28**	70.42	80	80.28	3.59
David2	94.60	88.45	**100**	94.04	95.53	6.84
deer	97.18	100	97.18	**100**	100	1.41
boy	95	93	90.67	98	**99.67**	5.50

Table 7: Percentage of correctly tracked frames obtained by crossing the parameter values between the video sequences. Each row presents the results obtained for a video sequence, by using its optimized set of parameters, as well as the parameter sets of four other sequences. The A.D. column shows the Average Difference (in percentages) between the result obtained with the optimized set of parameters (**bold** font) and those obtained with the parameter sets of the four other sequences.

was not dramatically affected by the change of parameters.

4.2.10. Computational cost

The proposed tracker was implemented using Matlab on a PC with a Core i7-3770 CPU running at a 3.4 GHz. Our algorithm is designed to maintain a reasonable computational complexity. In fact, keypoints are extracted in a limited image region determined by particle filtering to reduce the computational cost of feature detection and local descriptor creation. Moreover, the particle filter generates $N = 400$ particles, among which only a limited subset of N^* particles is used as a reduced search space on the current frame, and for generating the N particles on the subsequent frame. In practice, the computation time of **SCFT** is determined mostly by the number of detected object keypoints voting for the target position, which mainly depends on the object size and texture. As an example, the video sequences *tiger1* and *tiger2*, with a small target size, are processed at approximately 1.3 second per frame. On the other hand, when the object size is larger such as in the *faceocc* sequence, our algorithm requires from 2 to 3 seconds to find the target on a given frame. The table 9 provides a computation time comparison for the six trackers on the *David2* sequence that represents a typical scenario of face tracking. According to the performed measures, our algorithm requires in average 1.2 second to process one frame from the *David2* sequence, which is the second best execution time. AST achieved the shortest time, processing one frame in 0.42 second. Note that all the compared methods are

	girl	*tiger1*	*David2*	*deer*	*boy*	A.D.
girl	**9.29**	11.58	12.66	12.55	13.29	3.23
tiger1	16.18	**15.65**	21.17	18.31	16.27	2.32
David2	8.43	9.27	**3.04**	6.07	5.58	4.30
deer	7.63	7.03	7.63	**5.39**	9.77	2.63
boy	8.98	8.33	8.88	8.67	**7.42**	1.30

Table 8: Average location errors obtained by crossing the parameter values between the video sequences. Each row presents the results obtained for a video sequence, by using its optimized set of parameters, as well as the parameter sets of four other sequences. The A.D. column shows the Average Difference (in pixels) between the result obtained with the optimized set of parameters (**bold** font) and those obtained with the parameter sets of the four other sequences.

	SPT	SCMT	AST	MSIT	SAT	**SCFT**
time/video	1685.74	1738.34	225.95	1179.85	649.68	646.76
time/frame	3.14	3.24	0.42	2.20	1.21	1.20
ranking	5	6	1	4	3	2

Table 9: Processing time comparison for **SCFT** and the five other trackers on the video sequence *David2*. time/video: the total processing time (seconds), time/frame: the average processing time for one frame (seconds).

implemented in Matlab by the authors and run on our described computer.

5. Conclusion

This paper proposes a novel and effective part-based tracking algorithm, based on the collaboration of salient local features. Feature collaboration is carried out through a voting method where keypoint patches impose local geometrical constraints, preserving the target structure while handling pose and scale changes. The proposed algorithm uses saliency evaluation as a key technique for identifying the most reliable and useful features. Our conception of feature saliency includes three elements: *persistence*, *spatial consistency*, and *predictive power*. The *persistence* indicator allows to eliminate outliers (*e.g.* from the background, or an occluding object) and expired features from the target model, while the *spatial consistency* and the *predictive power* indicators penalize predictors that do not agree with past consensus. The experiments on publicly available videos from standard benchmarks show that SCFT outperforms state-of-the-art trackers significantly. Moreover, our tracker is insensitive to the number of tracked features, achieving accurate and robust tracking even if most of the local predictors are undetectable.

Acknowledgements

This work was supported by a scholarship from FRQ-NT and partially supported by NSERC discovery grant No. 311869-2010.

References

[1] O. Javed, M. Shah, Tracking and object classification for automated surveillance, in: European Conference on Computer Vision (ECCV), Springer, 2002, pp. 343–357.

[2] B. Lei, L.-Q. Xu, Real-time outdoor video surveillance with robust foreground extraction and object tracking via multi-state transition management, Pattern Recognition Letters 27 (15) (2006) 1816–1825.

[3] J.-P. Jodoin, G.-A. Bilodeau, N. Saunier, Urban tracker: Multiple object tracking in urban mixed traffic, in: Winter Applications of Computer Vision Conference (WACV), IEEE, 2014.

[4] M. Keck, L. Galup, C. Stauffer, Real-time tracking of low-resolution vehicles for wide-area persistent surveillance, in: Applications of Computer Vision (WACV), 2013 IEEE Workshop on, 2013, pp. 441–448. doi:10.1109/WACV.2013.6475052.

[5] H. Wang, A. Klaser, C. Schmid, C.-L. Liu, Action recognition by dense trajectories, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, 2011, pp. 3169–3176. doi:10.1109/CVPR.2011.5995407.

[6] W. Choi, C. Pantofaru, S. Savarese, Detecting and tracking people using an rgb-d camera via multiple detector fusion, in: Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, 2011, pp. 1076–1083. doi:10.1109/ICCVW.2011.6130370.

[7] L. Matthews, T. Ishikawa, S. Baker, The template update problem, Pattern Analysis and Machine Intelligence, IEEE Transactions on 26 (6) (2004) 810–815.

[8] X. Jia, H. Lu, M.-H. Yang, Visual tracking via adaptive structural local sparse appearance model, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 1822–1829.

[9] G. Nebehay, R. Pflugfelder, Consensus-based matching and tracking of keypoints for object tracking, in: Winter Conference on Applications of Computer Vision, 2014.

[10] W. Zhong, H. Lu, M.-H. Yang, Robust object tracking via sparsity-based collaborative model, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 1838–1845.

[11] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, Pattern Analysis and Machine Intelligence, IEEE Transactions on 27 (10) (2005) 1615–1630.

[12] A. Yilmaz, O. Javed, M. Shah, Object tracking: A survey, Acm Computing Surveys (CSUR) 38 (4) (2006) 13.

[13] M. Grabner, H. Grabner, H. Bischof, Learning features for tracking, in: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, IEEE, 2007, pp. 1–8.

[14] Y. Guo, Y. Chen, F. Tang, A. Li, W. Luo, M. Liu, Object tracking using learned feature manifolds, Computer Vision and Image Understanding 118 (2014) 128–139.

[15] S. Hare, A. Saffari, P. H. Torr, Efficient online structured output learning for keypoint-based object tracking, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 1894–1901.

[16] A. Adam, E. Rivlin, I. Shimshoni, Robust fragments-based tracking using the integral histogram, in: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, Vol. 1, IEEE, 2006, pp. 798–805.

[17] E. Erdem, S. Dubuisson, I. Bloch, Fragments based tracking with adaptive cue integration, Computer Vision and Image Understanding 116 (7) (2012) 827 – 841. doi:http://dx.doi.org/10.1016/j.cviu.2012.03.005.

[18] G. Hua, Y. Wu, Measurement integration under inconsistency for robust tracking, in: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, Vol. 1, 2006, pp. 650–657. doi:10.1109/CVPR.2006.181.

[19] S. Shahed Nejhum, J. Ho, M.-H. Yang, Visual tracking with histograms and articulating blocks, in: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, 2008, pp. 1–8. doi:10.1109/CVPR.2008.4587575.

[20] J. Kwon, K. M. Lee, Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping monte carlo sampling, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, 2009, pp. 1208–1215. doi:10.1109/CVPR.2009.5206502.

[21] S. Wang, H. Lu, F. Yang, M.-H. Yang, Superpixel tracking, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 1323–1330.

[22] W. Wang, R. Nevatia, Robust object tracking using constellation model with superpixel, in: Computer Vision–ACCV 2012, Springer, 2013, pp. 191–204.

[23] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International journal of computer vision 60 (2) (2004) 91–110.

[24] S. Leutenegger, M. Chli, R. Y. Siegwart, Brisk: Binary robust invariant scalable keypoints, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 2548–2555.

[25] J. Heinly, E. Dunn, J.-M. Frahm, Comparative evaluation of binary features, Computer Vision–ECCV 2012 (2012) 759–773.

[26] F. Yang, H. Lu, M.-H. Yang, Learning structured visual dictionary for object tracking, Image and Vision Computing 31 (12) (2013) 992–999.

[27] H. Grabner, J. Matas, L. Van Gool, P. Cattin, Tracking the invisible: Learning where the object might be, in: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 1285–1292.

[28] W. Bouachir, G.-A. Bilodeau, Structure-aware keypoint tracking for partial occlusion handling, IEEE Winter Conference on Applications of Computer Vision (WACV 2014).

[29] J. Triesch, C. Von Der Malsburg, Democratic integration: Self-organized integration of adaptive cues, Neural computation 13 (9) (2001) 2049–2074.

[30] K. Nickel, R. Stiefelhagen, Dynamic integration of generalized cues for person tracking, in: Computer Vision–ECCV 2008, Springer, 2008, pp. 514–526.

[31] P. Brasnett, L. Mihaylova, D. Bull, N. Canagarajah, Sequential monte carlo tracking by fusing multiple cues in video sequences, Image and Vision Computing 25 (8) (2007) 1217–1227.

[32] Q.-H. Zhou, H. Lu, M.-H. Yang, Online multiple support instance tracking, in: Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on, IEEE, 2011, pp. 545–552.

[33] Y. Wu, J. Lim, M.-H. Yang, Online object tracking: A benchmark, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE, 2013, pp. 2411–2418.

[34] B. Babenko, M.-H. Y. S. Belongie, Robust object tracking with online multiple instance learning, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI).

[35] J. F. Henriques, R. Caseiro, P. Martins, J. Batista, Exploiting the circulant structure of tracking-by-detection with kernels, in: Computer Vision–ECCV 2012, Springer, 2012, pp. 702–715.

[36] C. Geng, X. Jiang, Face recognition using sift features, in: Image Processing (ICIP), 2009 16th IEEE International Conference on, 2009, pp. 3313–3316. doi:10.1109/ICIP.2009.5413956.

[37] A. Mian, M. Bennamoun, R. Owens, An efficient multimodal 2d-3d hybrid approach to automatic face recognition, Pattern Analysis and Machine Intelligence, IEEE Transactions on 29 (11) (2007) 1927–1943. doi:10.1109/TPAMI.2007.1105.

[38] A. Mian, M. Bennamoun, R. Owens, Keypoint detection and local feature matching for textured 3d face recognition, International Journal of Computer Vision 79 (1) (2008) 1–12. doi:10.1007/s11263-007-0085-5.