Exploiting structural constraints for visual object tracking

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 paper presents a novel structure-aware method for visual tracking. The proposed tracker relies on keypoint regions as salient and stable elements that encode the object structure efficiently. In addition to the object structural properties, the appearance model also includes global color features that we first use in a probabilistic approach to reduce the search space. The second step of our tracking procedure is based on keypoint matching to provide a preliminary prediction of the target state. Final prediction is then achieved by exploiting object structural constraints, where target keypoints vote for the corrected object location. Once the object location is obtained, we update the appearance model and structural properties, allowing to track targets with changing appearance and non-rigid structures. Extensive experiments demonstrate that the proposed Structure-Aware Tracker (SAT) outperforms recent state-of-the-art trackers in challenging scenarios, especially when the

*Corresponding author

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

target is partly occluded and in moderately crowded scenes.

Keywords: Object tracking, Structure-aware tracker, keypoint, SIFT, keypoint layout.

---

1. Introduction

Model-free visual tracking is one of the most active research areas in computer vision [1, 2, 3]. With a *model-free* tracker, the only available input is the target state annotated in the first video frame. Tracking an object is thus a challenging task due to (1) the lack of sufficient information on object appearance, (2) the inaccuracy in distinguishing the target from the background (which is generally done using a geometric shape), and (3) the object appearance change caused by various perturbation factors (*e.g.* noise, occlusion, motion, illumination, etc.).

This work aims to develop a novel visual tracking method to handle real life difficulties, particularly when tracking an object in a moderately crowded scene in the presence of distracting objects similar to the target, and in the case of severe partial occlusion. The robustness of a tracking algorithm in handling these situations is determined by two major aspects: the target representation and the search strategy. The target representation refers to the appearance model that represents the object characteristics while the search strategy deals with how the search of the target is performed on every processed frame. The main contributions and differences of our work from previous works are on both aspects. In the proposed tracker, the target representation includes color features for coarse localization of the target, and keypoints for encoding the object structure while adding distinctiveness and robustness to occlusions. In our search strategy, probabilistic tracking and deterministic keypoint matching are used sequentially to provide a preliminary estimate of the target state. Object internal structural constraints are then applied in a correction step to find an accurate prediction. Our approach for representing the object structure is related to previous works on *context tracking* [4, 5, 6, 7, 8]. The main idea of *context tracking* is to consider the spatial context of the target including neighboring elements whose motion is correlated with the target. While the proposed approach is inspired by the idea of *context tracking*, in our work we exploit the spatial layout of keypoints to encode the internal structure of the target. More specifically, our contributions are:

1. A novel target representation model where local features are stored in a reservoir encoding recent and old structural properties of the target;
2. A new threefold search strategy that reduces the search space, tracks keypoints, and corrects prediction sequentially;
3. A discriminative approach that evaluates tracking quality online to determine if potential new target properties should be learned.

Extensive experiments on challenging video sequences show the validity of the proposed Structure-Aware Tracker (SAT) and its competitiveness with state-of-the-art trackers. A previous version of this work was presented at a conference [9]. This paper extends this previous work with a more complete review of related works, more details and depth in the explanation of the method, and additional experiments analyzing the tracker behavior in several situations.

This paper is organized as follows. In the next section, we review recent works on keypoint tracking and context tracking which are related to our algorithm. The proposed SAT algorithm is presented in section 3. Experimental results are given and discussed in section 4. Section 5 concludes the paper.

2. Related works

2.1. Keypoint tracking: from object context to object structure

Many tracking algorithms achieved good performances at a low complexity by using a geometric shape to contain the target, and global features for modeling [10, 11, 12]. Nevertheless, this approach is not designed to handle occlusions, unless representing the target by multiple fragments to be matched. Keypoint methods can handle the occlusion problem by establishing partial correspondences that allow locating the occluded target. Unlike fragment-based methods (where the target image region is divided randomly or according to a regular grid), keypoint locations correspond to salient and stable patches that can be invariantly detected under various perturbation factors. Moreover, their spatial layout naturally encodes structural properties that can enhance the target model.

Due to these characteristics, keypoint-based methods have attracted much attention during the last decade. In this approach, objects are modeled as a set of keypoints detected by an external mechanism (i.e. a keypoint detector) [13, 14, 15]. After computing their descriptors, the object localization can be achieved according to two possible approaches: matching in the case of a generative approach, and classification in the case of a discriminative approach. Generative trackers use a database where keypoint descriptors are stored. The descriptors are designed to be stable and invariant, and can be matched in a nearest-neighbor fashion. Discriminative approaches consider matching as a binary classification problem. Every feature is thus classified as belonging to the background, or to the tracked object. The classifier is built either via online learning, or offline, considering the background and the target observed under various transformations.

Some recent works on object tracking rely on target context to predict its state, which is often referred as *context tracking* [4, 5, 16, 7, 17]. According to this approach, it is necessary to consider target context to ensure the tracker robustness in most real life video surveillance applications. Following this principle, the authors in [4] use a *compagnion* to improve object tracking. This corresponds to image regions around the tracked object with the same movements as those of the target. In [5] the spatial context that can help the tracker includes multiple *auxiliary objects*. These objects have consistent motion correlation with the tracked target and thus help to avoid the drifting problem. In [16], Gu and Tomasi consider the spatial relationship between the target and similar objects and track all of them simultaneously to eliminate target confusion. In a more general approach, Grabner et al. introduced the notion of *supporters* defined as *"useful features for predicting the target object position"* [7]. These features do not belong to the target, but they move in a way that is statistically related to the motion of the target. They developed a method for discovering these local image features around the target, and demonstrated that motion coupling of *supporters* may allow locating the target even if it is completely occluded. In a later work, Dinh et al. [17] used *supporters* for context tracking, and added the concept of *distracters* which are regions co-occuring with the target while having a similar appearance. Their tracker explicitly handles situations where several objects similar to the target are present.

Context tracking methods expanded the target model by exploiting the motion correlation information in the scene. However, finding motion correlation between objects is a costly task that often requires detecting and analyzing features on the whole image, as in [18] where the authors detect and analyze all local features in the scene, to keep only features which move along with the target object. Furthermore, most of the proposed trackers were tested only on specific scenarios and in constrained environments, where almost all the experiments were limited to proofs of concept. Our idea of using structural constraints in the target appearance model is inspired by *context tracking* methods. However, our motivations differ in an important aspect since our model incorporates the internal structural information of the target, and not the structural layout of different scene elements. In our work, we show that the structural information of the target, encoded by the keypoint spatial layout, allows achieving accurate tracking and handling partial occlusion by inferring the position of the target using the unoccluded features.

2.2. Tracking objects by structure

The idea of exploiting object structure for tracking was present, more or less explicitly, in recent works. This is the so called *part-based tracking* that relies on local components for target representation. The most common way to encode object structure is the sparse representation such as in [19] and [20]. In [19], the authors propose to use a histogram-based model that encodes the spatial information of the object patches. In a similar manner, Jia et al. sample a set of overlapped patches on the tracked object [20]. Their strategy includes an occlusion handling module allowing target localization by using only visible image patches.

Another approach for encoding structure consists in using keypoints, since they are more significant than random overlapped patches. In this direction, the authors in [21] model the target by a set of keypoint *manifolds* organized as a graph to explicitly represent the target structure. Each feature *manifolds* includes, in addition to the keypoint descriptor, a set of synthetic descriptors simulating possible variations of the original feature (under viewpoint and scale change). The target location is found by detecting keypoints on the current frame, matching them with those of the target model, and computing a homography for the correspondences. In [22], the authors include both random patches and keypoints in the target model. The random patches are described by their RGB color histograms and LBP (Local Binary Patterns) descriptors to form an appearance model. Keypoints are characterized by their spatial histograms to be considered as a structural model. Tracking then implies matching detected keypoints in the current frame with those of the object in the previous frame. Matched keypoints are utilized to construct a spatial histogram, which is used jointly with LBP and RGB histograms to locate the target. This approach exploits multiple object characteristics (LBP, color, Keypoints), but the object structural model captures only recent structural properties, as the spatial histogram considers only the keypoints that are matched with those of the target in the last frame.

In our work, we argue and demonstrate through our experiments that keypoint regions are more efficient than random patches in encoding the structure, as they correspond to salient and stable patches invariably detectable under several perturbation factors. Unlike in [22] where random regions are analyzed to extract local features, and [21] where keypoints are extracted from a region with a fixed size (with the assumption of small displacements), we use a probabilistic method to reduce the search space to the most likely image regions, based on the target's global color features. Concerning the target structure, our structural model is not limited, like in [22] to recent properties, which would make it strongly related to the last prediction (and thus may be completely contaminated if the tracker drifts from the target). Instead, our representation includes both recent and old structural constraints in a reservoir of features. The local features and their structural constraints are learned online during tracking. The deletion of a given feature is related to its persistence (not to its moment of occurrence), while the impact of its constraint depends on the persistence as well as the consistence of the feature. Every local feature expresses its structural constraint individually by voting to possible target locations. Thus, our voting-based method preserves the object structure without requiring building and updating complex keypoint graphs, neither calculating homographies such as in [21]. Our method takes into consideration the temporal information of all the target's model components. The target model is thus updated to reflect the object appearance changes including structure changes, which allows tracking objects with non-rigid structures.

(a) (b) (c) (d)

Figure 1: Illustration of the SAT algorithm steps when tracking a partly occluded face. **(a)**: Reducing the search space with a probabilistic method, based on color. Local features (red dots) are computed only on the obtained areas. **(b)**: Predicting a preliminary target state based on feature matching. **(c)**: Visible features vote for a new position (yellow star) by applying their structural constraints. **(d)**: The target state is corrected based on the new location

3. Proposed algorithm

3.1. Motivation and overview

The proposed method is illustrated in figure 1 where we aim to track a partly occluded face. First, we apply a color-based particle filtering. This allows to reduce the search space and provides a coarse estimation by considering only the best particles. Keypoints are then detected by analyzing the reduced search space as shown in figure 1a. The detected keypoints are matched with those of the target model, which leads to a preliminary estimate of the target location (see figure 1b).

Note that the preliminary prediction considers only the matching scores of the particles and thus does not guarantee an accurate localization. This is illustrated in figure 1b, where the circular shape representing the best particle includes pixels from the background and from the occluding object. Knowing the internal structure of the target, our idea is to perform a correction step by applying internal structural constraints to improve target prediction. In practice, this is carried out by a voting mechanism where available features (unoccluded) determine the exact position of the target (figure 1c and 1d). Once the target is predicted, the appearance model including keypoints and their structural constraints is updated according to an evaluation criterion (that we define in section 3.5). The newly detected keypoints are added to the model while existing keypoints are re-evaluated based on two properties. First, we consider the individual keypoint persistence represented by its weight value. The second property is the spatial consistency of the keypoint that depends on the motion correlation with the target center. If a keypoint of the background is erroneously included in the target model, these two voting parameters will reduce the effect of its vote until its removal from the model when its persistence decreases significantly. Our algorithm steps are explained in details in the following.

3.2. Appearance Model

Our appearance model describes the image region delimited by the circle that circumscribes the target. This is a multi-features model including (1) the color probability distribution represented by a weighted histogram, (2) a set of local descriptors computed for the detected keypoints within the target region, and (3) the target structural properties encoded by the voting parameters of keypoints. By constructing a m-bin histogram $\hat{\mathbf{q}} = \{\hat{q}\}_{u=1\ldots m}$, with $\sum_{u=1}^{m} \hat{q}_u = 1$, some parts of the background may lie inside the circular kernel. As discussed in [23], these pixels will affect the color distribution and may cause tracking drift. To reduce the effect of these pixels, we use a kernel function $k(x)$ that assigns smaller weights to pixels farther from the center. The color histogram is thus computed for the h pixels inside the target region according to the equation:

$$\hat{q}_u = \frac{1}{\sum_{i=1}^{h} k(d_i)} \sum_{i=1}^{h} k(d_i)\delta[c_i - u] \tag{1}$$

where $d_i \in [0, 1]$ is the normalized distance from the pixel x_i to the kernel center, c_i is the bin index for x_i in the quantized space, δ is the Kronecker delta function, and $k(d_i)$ is the tricube kernel profile defined by:

$$k(d_i) = \frac{70}{81}(1 - d_i^3)^3. \tag{2}$$

Note that the tricube function was selected among various kernel functions, as it allows the best experimental result. We also note that any other color space could be used instead of RGB.

The proposed system should be able to handle many difficult scenarios, such as occlusions and the presence of distracting objects. For, example, it has been shown that even for individuals of different races, the skin color distributions are very similar [24]. To ensure a more robust and distinctive feature set, the target reference model also includes SIFT keypoints [25] detected in the target region and stored in a *Reservoir of Features* (RF). SIFT features increase the distinctiveness of the tracking algorithm to distinguish the target from other similar objects that may enter the field of view. In fact, SIFT was successfully used for distinguishing between multiple instances of the same object such as in the face recognition problem[26, 27, 28]. In this way, we implicitly handle situations where objects of the same category as the target co-occur (*e.g.* tracking a face in the presence of several faces), and

Algorithm 1 Reducing the search space at frame t

Input: frame t, particle states after processing frame $t-1$

Output: reduced search space, new particle states

Assumption: processing frame t with $t > 2$

for $i = 1$ **to** N **do**
- generate a random number $r_i \in [0, 1]$
- find the particle $s_{t-1}^{(j)}$ with the smallest j verifying $c_{t-1}^{(j)} \geq r_i$
- generate a new particle $s_t^{(i)}$ for the selected particle $s_{t-1}^{(j)}$, with $s_t^{(i)} = f(s_{t-1}^{(j)})$
- evaluate similarity between $\hat{p}_t^{(i)}$ and $\hat{\mathbf{q}}$ {Eq. 3 and 4}
- compute the weight $\pi_t^{(i)}$ for $s_t^{(i)}$

end for
- select the N^* best particles
- normalize weights $\pi_t^{(n)}$ to get $\sum_{n=1}^{N^*} \pi_t^{(n)} = 1$
- compute cumulative probabilities $c_t^{(n)}$

thus we avoid using an additional mechanism to track and distinguish *distracters* as in [17]. Other than the keypoint descriptors, we also exploit the spatial layout of keypoints to encode structural properties of objects. The target structural constraints and the voting method that we use for prediction correction are explained later. We note that our method is not specific to SIFT. Even faster keypoint detector/descriptor combination may be used, although SIFT remains one of the most reliable methods under various image transformations [29].

3.3. Reducing the search space

The target search is firstly guided by particle filtering [30]. Each particle is a circular region characterized by its color distribution as explained above. The possible target states at frame t are represented by N weighted particles $\{s_t^{(i)} : i = 1, ..., N\}$ where the weight $\pi_t^{(i)}$ reflects the importance of the particle. The weight of a generated particle $s_t^{(i)}$ depends on the similarity between its color distribution $\hat{p}_t^{(i)}$ and the reference color model $\hat{\mathbf{q}}$. We define the distance between the two distributions as:

$$d(\hat{\mathbf{q}}, \hat{p}_t^{(i)}) = \sqrt{1 - \rho[\hat{q}, \hat{p}_t^{(i)}]} \tag{3}$$

where

$$\rho[\hat{q}, \hat{p}_t^{(i)}] = \sum_{u=1}^{m} \sqrt{\hat{q}_u . \hat{p}_{u,t}^{(i)}} \tag{4}$$

is the Bhattacharyya coefficient between $\hat{\mathbf{q}}$ and $\hat{p}_t^{(i)}$.

After generating N particles on the current frame, the area covered by the N^* best particles (i.e. the particles having the highest weights) is considered as a coarse estimation of the target state, and thus constitutes a reduced search space where keypoints will be detected and matched. Moreover, we use the N^* states selected at frame t for generating N particles at frame $t + 1$. Note that to simplify computations, we assign a cumulative weight $c^{(n)}$ to each pair $(s^{(n)}, \pi^{(n)})$ where $c^{(N^*)} = 1$. The cumulative weight $c^{(n)}$ for the n^{th} particle is calculated as $c^{(n)} = c^{(n-1)} + \pi^{(n)}$, where $c^{(1)} = \pi^{(1)}$. In this manner, for each particle $s^{(n)}$ we assign the interval $[c^{(n-1)}, c^{(n)}] \subset [0, 1]$ to allow a random particle selection (see steps 2 and 3 in Alg. 1). Our space reduction algorithm is summarized in Alg. 1.

3.4. Tracking keypoints

Keypoint detection and matching will consider only the reduced search space defined by the N^* best particles. By reducing the search region to the most important candidate particles, we avoid detecting features, computing local descriptors and matching them on the entire image.

The detected descriptors are then matched with those of the target model (features from the reservoir RF) based on the Euclidian distance. Similarly to the criterion used in [25], we determine if a match is correct by evaluating the ratio of distance from the closest neighbor to the distance of the second closest. For our algorithm, we keep only the matches for which the distance ratio is less than $\theta_m = 0.7$. Given the final set of matched pairs, we consider the particle having the highest matching score as a preliminary state of the target (see figure 1b). A more formal description of the preliminary prediction is provided in Alg. 2. Since the preliminary prediction considers only matching scores, without guaranteeing an accurate localization of the selected particle, the structural properties of the predicted region will be analyzed in a correction step to provide an accurate estimation of the target location.

3.5. Applying structural constraints

In this step, we aim to correct the preliminary prediction by applying a learned structural model of the target. The model is learned from reliable measurements (*i.e.* when a good tracking is achieved), and the internal structural properties are considered as a part of the object appearance model.

Internal structural model. The target keypoints extracted on the target region at different times of its lifecycle are stored in the reservoir

Algorithm 2 Preliminary prediction at frame t

- detect features on the reduced search space
for all detected_features $f^{(i)}$ **do**
    - compute Euclidian distance with features from RF
    - compute $dist\_ratio = \frac{dist(f^{(i)}, closest\_neighbor)}{dist(f^{(i)}, 2^{nd}\_closest\_neighbor)}$
    if $dist\_ratio \leq \theta_m$ **then**
        - match $f^{(i)}$ with *closest_neighbor*
        - update matching scores for the particles containing $f^{(i)}$
    end if
end for
- $preliminary\_prediction_t$= the particle having the highest score

of features RF. Instead of automatically eliminating old keypoints, we only remove those that become "non-persistent". RF is thus formed by recent and old keypoints, representing both old and recent object properties. Other than its descriptor summarizing the local gradient information, every keypoint is characterized by a *voting profile* (μ, w, Σ) where:

- $\mu = [\Delta_x, \Delta_y]$ is the average offset vector that describes the keypoint's location with respect to the target region center;
- w is the keypoint's weight considered as a persistence indicator to reflect the feature co-occurence with the target, and to allow eliminating "bad" keypoints;
- Σ is the covariance matrix used as a spatial consistency indicator, depending on the motion correlation with the target center.

Voting. Every matched keypoint f that is located on the preliminary target region votes for the potential object position $\mathbf{x}$ by $P(\mathbf{x}|f)$. Note that we accumulate the votes for all the pixel positions inside the reduced search space. Given the *voting profile* of the feature f, we estimate the voting of f with the Gaussian probability density function:

$$P(\mathbf{x}|f) \propto \frac{1}{\sqrt{2\pi|\Sigma|}}\; exp\,(-0.5\,(\mathbf{x}_f - \mu)^\top\,\Sigma^{-1}(\mathbf{x}_f - \mu)), \tag{5}$$

where $\mathbf{x}_f$ is the relative location of $\mathbf{x}$ with respect to the keypoint coordinates. The probability of a given pixel in the voting space is estimated by accumulating the votes of keypoints weighted by their persistence indicators w. The probability for a given pixel position $\mathbf{x}$ in the voting space at time t is estimated by:

$$P_t(\mathbf{x}) \propto \sum_{i=1}^{|RF|} w_t^{(i)} P_t(\mathbf{x}|f^{(i)}) \mathbb{1}_{\{f^{(i)} \in F_t\}}, \tag{6}$$

where $\mathbb{1}_{\{f^{(i)} \in F_t\}}$ is the indicator function defined on the set RF (reservoir of features), indicating if the considered feature $f^{(i)}$ is among the matched target features set F_t at frame t. The target position is then found by analyzing the voting space and selecting its peak to obtain the corrected target state as shown in figure 1c.

Update. It has been previously shown that an adaptive target model, evolving during the tracking, is the key to good performance [31]. In our algorithm, the target model (including color, keypoints, and structural constraints) is updated every time we achieve a good tracking using a discriminative approach. Our definition of a good tracking is inspired by the Bayesian evaluation method used in [32], referred as *histogram filtering.* Using the target histogram $\hat{\mathbf{q}}$ (calculated for the target region annotated in the first frame), and the background histogram $\hat{\mathbf{q}}_{bg}$ (calculated for the area outside the reduced search space), we compute a filtered histogram $\hat{\mathbf{q}}_{filt} = {}^{\hat{\mathbf{q}}}/_{\hat{\mathbf{q}}_{bg}}$ in every iteration. The latter represents the likelihood ratios of pixels belonging to the target. The likelihood ratios are used to calculate a backprojection map on the target region. Quality evaluation is done by analyzing the back-projection map and thresholding it to determine the percentage of pixels belonging to the target. Every time the evaluation procedure shows sufficient tracking quality, the target model is updated at frame t with a learning factor α as follows:

$$\hat{q}_t = (1 - \alpha)\hat{q}_{t-1} + \alpha\hat{q}_{new} \tag{7}$$

$$\hat{q}_{bg,t} = (1 - \alpha)\hat{q}_{bg,t-1} + \alpha\hat{q}_{bg,new} \tag{8}$$

$$w_t^{(i)} = (1 - \alpha)w_{t-1}^{(i)} + \alpha\mathbb{1}_{\{f^{(i)} \in F_t\}} \tag{9}$$

$$\Delta_{x,t}^{(i)} = (1 - \alpha)\Delta_{x,t-1}^{(i)} + \alpha\Delta_{x,new}^{(i)} \tag{10}$$

$$\Delta_{y,t}^{(i)} = (1 - \alpha)\Delta_{y,t-1}^{(i)} + \alpha\Delta_{y,new}^{(i)} \tag{11}$$

where $\mu_{new}^{(i)} = [\Delta_{x,new}^{(i)}, \Delta_{y,new}^{(i)}]$ is the current estimate of the voting vector for the feature $f^{(i)}$. After updating the feature weights, we remove from RF all the features whose the persistence indicators become less than the persistence threshold θ_p (*i.e.* $w_t^{(i)} \leq \theta_p$) regardless if they are recent or old, and we add the newly detected features with initial weight w_0. Further, we update the covariance matrix to determine the spatial consistency of the feature by applying:

$$\Sigma_t^{(i)} = (1-\alpha)\Sigma_{t-1}^{(i)} + \alpha\Sigma_{new}^{(i)}, \tag{12}$$

where the new correlation estimate is:

$$\Sigma_{new}^{(i)} = (\mu_{new}^{(i)} - \mu_t^{(i)})(\mu_{new}^{(i)} - \mu_t^{(i)})^{\top}, \tag{13}$$

with $\mu_t^{(i)} = [\Delta_{x,t}^{(i)}, \Delta_{y,t}^{(i)}]$. Note that for the newly detected features, the preliminary persistence indicator is initialized to the covariance matrix $\Sigma = \sigma_0^2 I_2$, where I_2 is a 2 x 2 identity matrix. For consistent features, Σ decreases during the tracking, and thus their votes become more concentrated in the voting space. The overall algorithm is presented in Alg. 3.

4. Experiments

4.1. Experimental setup

We evaluated our SAT tracker by comparing it with four recent state-of-the-art methods on 11 challenging video sequences. Seven sequences of the dataset are publicly available and commonly used in the literature, while four are our own sequences[1]. The *Tiger 1*, *Tiger2* and *Cliff bar* are provided in [1] and the *David indoor* and *Sylvester* are from [33]. The *Girl* and *occluded face 1* video sequences are respectively from [34] and [35]. The sequences *jp1*, *jp2*, *wdesk*, and *wbook* (with 608, 229, 709, and 581 frames respectively) were captured in our laboratory using a Sony SNC-RZ50N camera. The video

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

frames are 320x240 pixels captured at a frame rate of 15 fps. For quantitative evaluation, we manually labeled the ground truth of our four sequences. Some of the sequences are available only in grayscale format (*Tiger 1*, *Tiger2* , *Sylvester*, and *Cliff bar*). For these videos, we slightly adapted our algorithm (especially the color model) to use grayscale information instead of RGB color information.

The four methods that we used for our comparison are the SuperPixel Tracker (SPT) [36], the Sparsity-Based Collaborative Tracker (SBCT) [19], the Adaptive Structural Tracker (AST) [20], and the Online Multiple Support Instance Tracker (OMSIT) [37]. The source codes of these trackers are available on the authors' respective websites. The authors also provide various parameter combinations. For fairness, we tuned the parameters of their methods so that for every video sequence, we always use the best combination among the ones that they proposed. Most of the parameters of SAT were set to default values for all the sequences, and only three parameters were tuned to optimize the performance of the tracker:

- N^*: the number of particles defining the reduced search space.
- θ_u: the threshold on the percentage of pixels belonging to the target that is required to update the appearance model.
- θ_p the persistence threshold used to determine if the keypoint should be removed from the reservoir.

table 1 shows the optimized parameter values for 5 video sequences from our dataset.

parameters	*girl*	*tiger 1*	*David indoor*	*occluded face 1*	*Wdesk*
N^*	30	100	100	40	80
θ_u	0.6	0.75	0.55	0.7	0.65
θ_p	0.3	0.4	0.2	0.2	0.3

Table 1: The optimized parameter values used in SAT with each video from the subset including *girl*, *tiger 1*, *David indoor*, *occluded face 1*, and *Wdesk*.

We quantitatively evaluated the performance of the trackers using the success rate and the average location error. To measure the success rate, we calculate 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. Tracking is considered as a success for a given frame, if OR is larger than 0.5. The evaluation of the Center Location Error (CLE) is based on the relative position errors between the center of the tracking result and that of the ground truth. Table 2 presents the success rates and the average center location errors for the compared methods. In order to analyze in depth the compared methods on several video sequences, we also prepared two plots for every video sequence: 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 understanding more in details the behavior of the trackers since the success rate and the average location error just summarize the performance of the tracker on a given sequence. Note that we averaged the results over five runs in all our experiments.

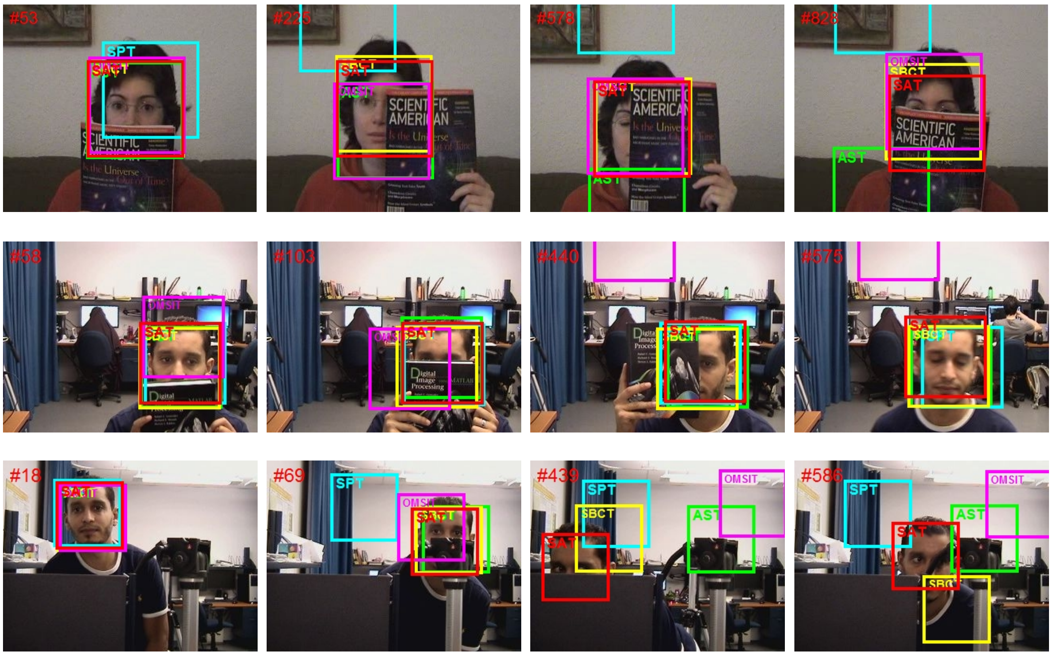

Figure 2: Tracking results for video sequences with long-term occlusions: *Occluded face 1, Wbook, Wdesk*. Green, magenta, yellow, cyan, and red rectangles correspond to results from AST, OMSIT, SBCT, SPT, SAT.

Sequence	SPT S	SPT E	SBCT S	SBCT E	AST S	AST E	OMSIT S	OMSIT E	SAT S	SAT E
David indoor	62	36	60	34	38	69	*63*	*27*	**100**	**10**
girl	*84*	**9**	2	201	18	53	1	66	**85**	*10*
occluded face 1	6	117	**100**	**5**	26	85	81	23	**100**	*14*
tiger 1	**61**	*17*	25	108	31	38	3	75	*51*	**15**
tiger 2	*46*	*23*	16	189	31	29	6	45	**70**	**16**
Sylvester	39	32	49	34	*73*	**10**	3	99	**79**	*14*
Cliff bar	52	**22**	24	77	**70**	35	8	74	*60*	*25*
Jp1	18	35	78	18	*84*	*17*	4	97	**89**	**7**
Jp2	39	*31*	*55*	69	*55*	45	17	39	**94**	**7**
Wdesk	14	80	*57*	*34*	32	81	10	123	**90**	**11**
Wbook	99	11	**100**	**5**	**100**	*9*	9	132	**100**	12
average	47	*38*	*52*	70	51	43	19	73	**84**	**13**

Table 2: Success rate (S) and average location error (E) results for SAT and the four other trackers: **Bold red** font indicates best results, *blue italics* indicates second best.

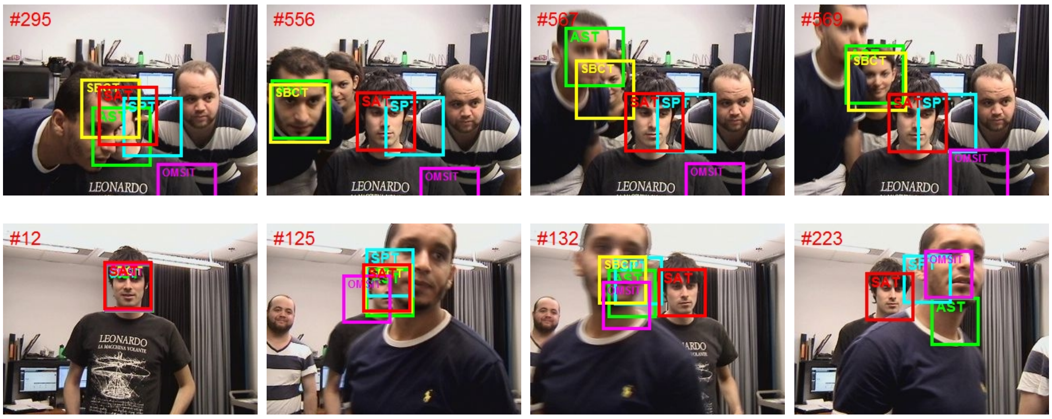

Figure 3: Screenshots of face tracking in moderately crowded scenes under short-term occlusions. In the *Jp1* sequence (first row), the tracked face is the one that is in the center of the scene. The same person is tracked while he is walking in the *Jp2* sequence. Green, magenta, yellow, cyan, and red rectangles correspond to results from AST, OMSIT, SBCT, SPT, SAT.

4.2. Experimental results

Long-time occlusion: Figure 2 demonstrates the performance of the compared trackers when tracking faces under long-time partial occlusions. In the *Occluded face 1* and the *wbook* sequences, the target faces remain partially occluded for several seconds while they barely move. The corresponding plots in figures 6 and 7 show that some trackers drift away from the target face, to track the occluding object (*e.g.* between frames 200 and 400 in *Occluded face 1*). Because it is specifically designed to handle partial occlusions via its structure-based model, our tracker was able to track the faces successfully in practically all the frames. SBCT has also achieved a good performance with a slightly lower average location error. In fact, SBCT is also designed to handle occlusions using a scheme that considers only the

Figure 4: Screenshots of tracking results for some of the sequences with illumination change (*david indoor*) and background clutter (*Cliff bar, Tiger1, Tiger2*). Green, magenta, yellow, cyan, and red rectangles correspond to results from AST, OMSIT, SBCT, SPT, SAT.

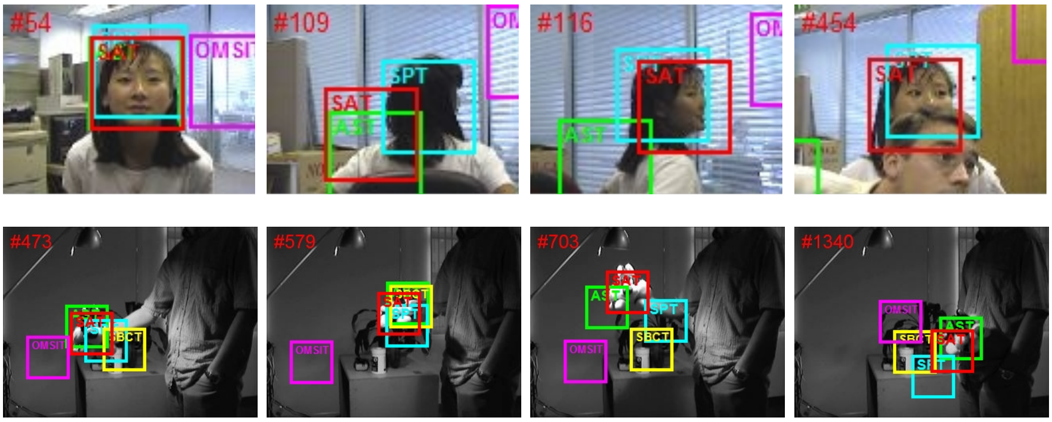

Figure 5: Tracking results for video sequences with abrupt motion and/or out of plane rotation: *Girl* and *Sylvester* sequences. Green, magenta, yellow, cyan, and red rectangles correspond to results from AST, OMSIT, SBCT, SPT, SAT.

patches that are not occluded. The target face in *Wdesk* undergoes severe partial occlusions many times while moving behind structures of the background. SAT and SBCT track the target correctly until frame 400. At this point the person performs large displacements, and SBCT drifts away from the face. Nevertheless, our tracker continues the tracking successfully while the tracked person is trying to hide behind structures of the background, achieving a success rate of 90%. The superiority of the proposed method in this experiment highlights the importance of using structural constraints defined by keypoint regions that are more invariant than the patches used in SBCT when such a situation occurs.

Moderately crowded scenes: Figure 3 presents the results of face tracking in a moderately crowded scene (four persons). In the *Jp1* video, we aim to track a target face in presence of other faces that may partially occlude the target. Although the success rates of 84% and 78% respectively

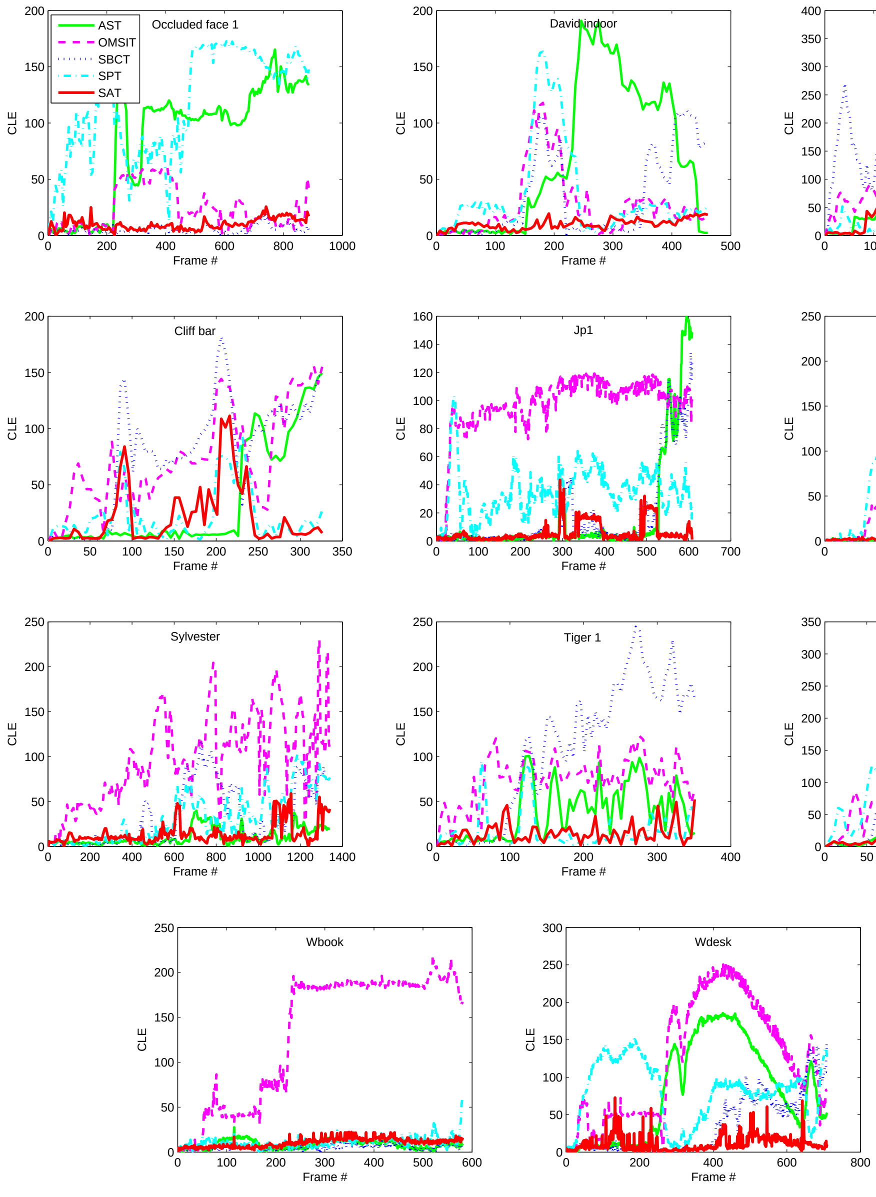

Figure 6: Center location error plots.

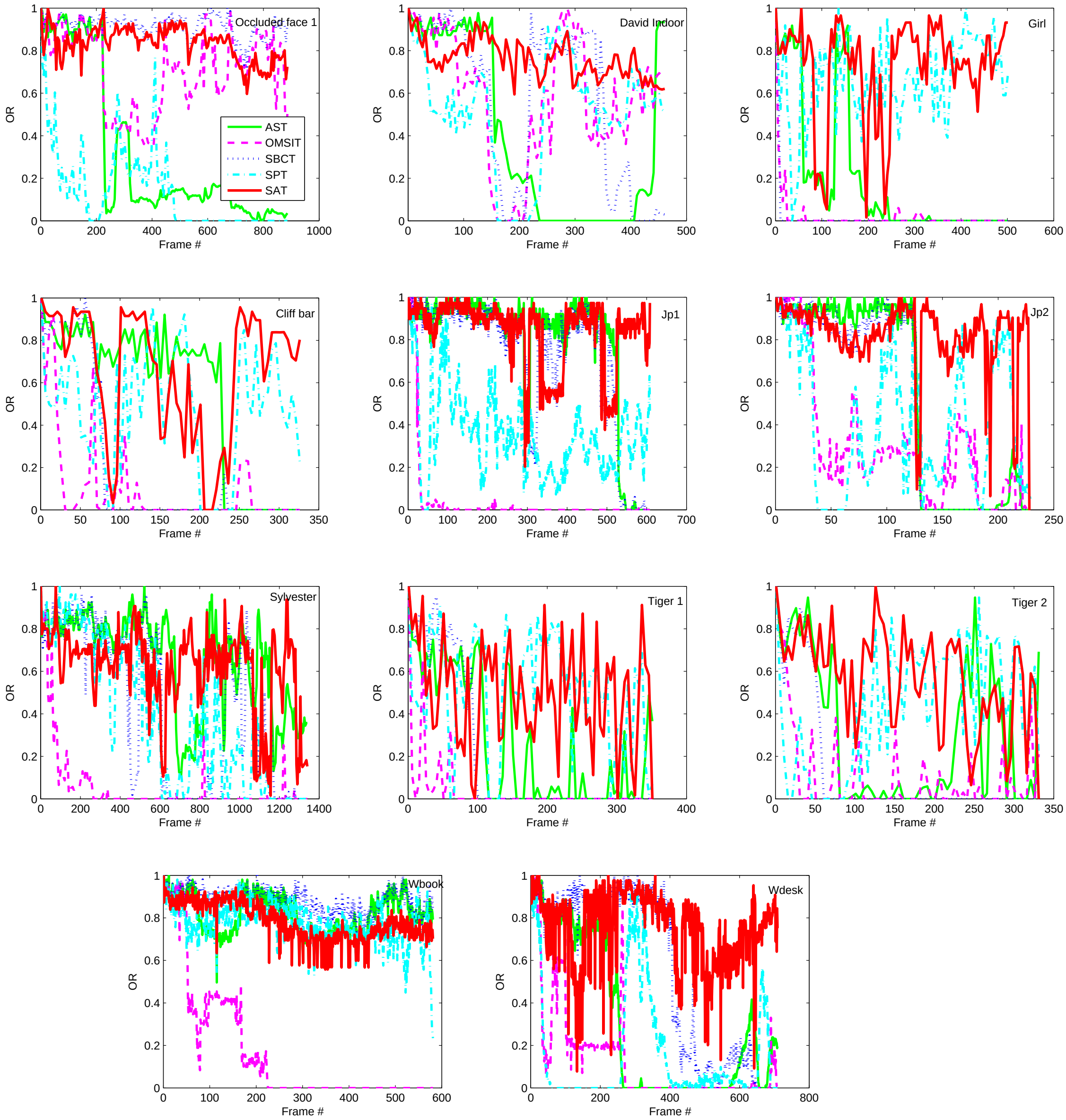

Figure 7: Overlap ratio plots.

for AST and SBCT indicate good performance in general, the two trackers drift twice, first at frame 530, and a second time at frame 570, to track other faces occluding or neighboring the target face. However, our tracker is not affected by the presence of similar objects around the target, even if partial occlusion occurs. This is mainly due to the distinctiveness of SIFT features compared to the local patches used in AST and SBCT to characterize the target. In this manner, SIFT features allow our tracker to handle situations where multiple instances of the same target object co-occur. In the *jp2* sequence, we track a walking person in a moderately crowded scene with four randomly moving persons. Here, we track a person's face that crosses in front or behind another walking person that may completely occlude the target for a short time. Except the proposed method, none of the trackers is able to relocate the target after full occlusion by another person. For example, SBCT confused the target with the occluding face like in the video sequence *Jp1*. In this situation, SAT detects a total occlusion (since no features are matched). Our tracker continues searching the target based on color similarity without updating the appearance model. Tracking is finally recovered as soon as a small part of the target face becomes visible and feature matching becomes possible again.

Illumination change: In the *David indoor* video, the illumination changes gradually as the person moves from a dark room to an illuminated area (see figure 4). While most of the trackers were able to keep track of the person in more than 60% of the frames, SAT was the only tracker to achieve a success rate of 100%. In addition, SAT had the best performance on the *Sylvester* sequence in which the target object appearance changes drastically due to abrupt illumination change. These two experiments show the superiority of our appearance model, which is the only one among the five models, to include keypoints that are robust against lighting variations. Note that every time we update the reservoir of features, we replace the descriptors of all matched keypoints by their latest version computed on the current frame. This technique helps also to reflect appearance changes of keypoint regions (caused by illumination, viewpoint change, etc.), which facilitates matching features.

Background clutters: In the *Cliff bar* video, the background (the book) and the target have similar textures. Figure 4 shows that SBCT and OMSIT drift away from the target in most video frames. AST, SPT, and the proposed tracker were able to achieve a better performance despite the difficulty of this sequence. In fact, the target undergoes drastic appearance changes due to high motion blur. This caused drifts for all trackers several times (*e.g.* see the corresponding CLE and OR plots at frame 80). In the *Tiger 1* and *Tiger 2* sequences, the tracked object exhibits fast movements in a cluttered background with frequent and various occlusion level. Owing to our voting mechanism that predicts the exact position of the target from the visible keypoints, our SAT tracker overcomes the frequent occlusion problem outperforming the other methods. All the other methods fail to locate the stuffed animal, except SPT that achieved better results due to its discriminative appearance model that facilitates the distinction between the object and the background based on superpixel over-segmentation. Note that our method also presents a discriminative aspect, since it uses information on the background color distribution to evaluate the tracking quality (see the update subsection under section 3.5).

Abrupt motion and out of plane rotation: The target object in *Sylvester* undergoes out of plane rotation and sudden movements during more than 1300 frames. Most of the trackers, except AST and ours do not perform well. In the *girl* video, the tracked face undergoes both pose change and 360 degrees rotations abruptly. Our method had the highest success rate and was significantly more robust and accurate than most of the methods as we can see in figure 5. SAT handled efficiently pose change and partial occlusion and our tracking was successful as long as the girl's face was at least partly visible. The target was lost only during the frames where it is completely turned away from the camera (see the OR plot, frames 87-116 and 187-250), but tracking is recovered as soon as the face reappears.

Computational cost: Our tracker was implemented using Matlab on a PC with a Core i7-3770 CPU running at a 3.4 GHz. SAT algorithm is designed to maintain a reasonable computational complexity. In fact, we extract local features in a limited image region determined by particle filtering, in order to reduce the computational cost of keypoint detection and local descriptors creation. The particle filter generates $N = 400$ particles, among which only N^* particles are considered as a reduced search space, and for generating the N particles on the subsequent frame. In practice, the computation time of our tracker is closely related to the number of detected keypoints voting for the object position, which mainly depends on the object size and texture. As an example, the video sequences *tiger 1* and *tiger 2*, with a small target size, are processed at nearly one second per frame. On the other hand, when the object size is larger such as in the *occluded face 1*,

	SPT	SBCT	AST	OMSIT	SAT
time/video	1854.31	1990.52	259.84	1327.23	707.41
time/frame	3.95	4.24	0.55	2.82	1.51
ranking	4	5	1	3	2

Table 3: Processing time comparison on the *David indoor* sequence. time/video: the total processing time (seconds), time/frame: the average processing time for one frame (seconds).

SAT requires up to 3 seconds to find the target on certain frames. The table 3 provides a computation time comparison for the five trackers on the face tracking video *David indoor*. All the compared trackers were implemented in Matlab by the authors, and run on the same described computer. According to the performed measures, our algorithm requires in average 1.51 s to process one frame, which is the second best execution time. We note that AST achieved the shortest time, processing one frame in 0.55 s.

Application constraints and risk of failure: The proposed tracker uses SIFT algorithm as an external mechanism to detect the target keypoints. Generally, our method achieves high accuracy when a significant number of keypoints are detected on the target object. On the other hand, the tracking quality may decrease if the target region is not sufficiently textured, or if it is too far from the camera (object details not visible). As an example, we verified that the face tracking application requires a maximum distance of 10 meters between the tracked person and the camera. At this distance, SIFT allows detecting between two and four keypoints in most face tracking scenarios. Furthermore, a drastic decrease in the number of visible target keypoints increases the drifting risk, regardless of the target type. In practice, our tracker relies on keypoint matching only if at least three keypoints from the reservoir are matched on the current frame. Otherwise, SAT applies the particle filter (that we use to reduce the search space) to track the object based on its global color distribution. Another limitation may result from the use of a small number of particles to limit the keypoint detection region. Indeed, the target may undergo large displacements between consecutive frames due to fast movements or low frame rates (e.g. real-time tracking using a remote IP camera). As a result, the target object may be located outside the keypoint detection area, causing tracking failure. If this situation occurs, tracking can be recovered only if the target reappears in the reduced search space. Note that this problem can be solved at the cost of an additional computation time, by increasing the number of particles (N^*) forming the reduced search space.

5. Conclusion

In this paper, we proposed a robust tracking algorithm named SAT (Structure Aware Tracker). Our core idea is to exploit the structural properties of the target, in a voting-based method, to provide accurate location prediction. The target is described by color distribution, keypoints, and their geometrical constraints encoding the object internal structure. This multi-features appearance model is learned during tracking and thus incorporates new structural properties in an online manner. Numerous experiments in a comparison with four state-of-the-art trackers, on eleven challenging video sequences, demonstrate the superiority of the proposed method in handling multiple tracking perturbation factors. Our results also highlight the importance of encoding the object structure via keypoint regions, that are more invariant and stable than other types of patches (*e.g.* the local patches encoding the object spatial information in AST and SBCT).

Acknowledgements

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

References

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

[2] Z. Kalal, J. Matas, K. Mikolajczyk, Pn learning: Bootstrapping binary classifiers by structural constraints, in: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 49–56.

[3] L. Zhang, L. van der Maaten, Structure preserving object tracking, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, 2013, pp. 1838–1845.

[4] J. M. LukasCerman, V. Hlavac, Sputnik tracker: Having a companion improves robustness of the tracker, in: Image Analysis: 16Th Scandinavian Conference, Scia 2009, Oslo, Norway, June 15-18, Proceedings, Vol. 5575, Springer, 2009, p. 291.

[5] M. Yang, Y. Wu, G. Hua, Context-aware visual tracking, Pattern Analysis and Machine Intelligence, IEEE Transactions on 31 (7) (2009) 1195–1209.

[6] L. Wen, Z. Cai, Z. Lei, D. Yi, S. Z. Li, Online spatio-temporal structural context learning for visual tracking, in: Computer Vision–ECCV 2012, Springer, 2012, pp. 716–729.

[7] 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.

[8] A. Saffari, M. Godec, T. Pock, C. Leistner, H. Bischof, Online multi-class lpboost, in: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, 2010, pp. 3570–3577. doi:10.1109/CVPR.2010.5539937.

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

[10] G. D. Hager, M. Dewan, C. V. Stewart, Multiple kernel tracking with ssd, in: Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, Vol. 1, IEEE, 2004, pp. I–790.

[11] S. Hare, A. Saffari, P. H. Torr, Struck: Structured output tracking with

kernels, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 263–270.

[12] G. Shu, A. Dehghan, O. Oreifej, E. Hand, M. Shah, Part-based multiple-person tracking with partial occlusion handling, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 1815–1821.

[13] H. Zhou, Y. Yuan, C. Shi, Object tracking using sift features and mean shift, Computer Vision and Image Understanding 113 (3) (2009) 345–352.

[14] 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.

[15] W. Bouachir, G.-A. Bilodeau, Visual face tracking: A coarse-to-fine target state estimation, 2013 International Conference on Computer and Robot Vision 0 (2013) 45–51. doi:http://doi.ieeecomputersociety.org/10.1109/CRV.2013.18.

[16] S. Gu, C. Tomasi, Branch and track, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 1169–1174.

[17] T. B. Dinh, N. Vo, G. Medioni, Context tracker: Exploring supporters and distracters in unconstrained environments, in: Computer Vision and

Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 1177–1184.

[18] K. M. Yi, H. Jeong, B. Heo, H. J. Chang, J. Y. Choi, Initialization-insensitive visual tracking through voting with salient local features, 2013 International Conference on Computer Vision (ICCV).

[19] 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.

[20] 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.

[21] 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.

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

[23] V. Belagiannis, F. Schubert, N. Navab, S. Ilic, Segmentation based particle filtering for real-time 2d object tracking, Computer Vision–ECCV 2012 (2012) 842–855.

[24] H.-M. Sun, Skin detection for single images using dynamic skin color modeling, Pattern recognition 43 (4) (2010) 1413–1420.

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

[26] 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.

[27] 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.

[28] 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. URL `http://dx.doi.org/10.1007/s11263-007-0085-5`

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

[30] M. Isard, A. Blake, Condensation: conditional density propagation for visual tracking, International journal of computer vision 29 (1) (1998) 5–28.

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

[32] K. Bernardin, F. Van De Camp, R. Stiefelhagen, Automatic person detection and tracking using fuzzy controlled active cameras, in: Computer

Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, IEEE, 2007, pp. 1–8.

[33] D. A. Ross, J. Lim, R.-S. Lin, M.-H. Yang, Incremental learning for robust visual tracking, International Journal of Computer Vision 77 (1-3) (2008) 125–141.

[34] S. Birchfield, Elliptical head tracking using intensity gradients and color histograms, in: Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on, IEEE, 1998, pp. 232–237.

[35] 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.

[36] 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.

[37] 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.

Algorithm 3 Predicting the target location

- initialize RF, $\hat{q}$, $\hat{q}_{bg}$
for all $frames$ **do**
    - reduce the search space: Alg. 1
    - predict a preliminary state: Alg. 2
    for all *voting_space_positions* $\mathbf{x}$ **do**
        for all $matched\_features$ $(f^{(i)} \in F_t)$ **do**
            - estimate $P(\mathbf{x}|f^{(i)})$: (Eq. 5)
        end for
        - estimate location probability $P(\mathbf{x})$: (Eq. 6)
    end for
    - *target_location* = *select_peak*(*voting_space_positions*) {tracker's output for the current frame}
    if (*update_condition* $==$ *true*) **then**
        -update $\hat{q}_t$ and $\hat{q}_{bg,t}$: (Eq. 7 & 8)
        for all $matched\_features$ $(f^{(i)} \in F_t)$ **do**
            - update $\mu_t^{(i)}$ (Eq. 10 & 11)
            - update $\Sigma_t^{(i)}$ (Eq. 12)
        end for
        - update $w_t^{(i)}$ (Eq. 9) for the entire reservoir
        - remove non-persistent features ($i.e.\, w_t^{(i)} \leq \theta_p$)
        for all $newly\_detected\_features\, f^{(i)}$ **do**
            - add $f^{(i)}$ to RF
            - $\mu_t^{(i)} = [\Delta_{x,new}^{(i)}, \Delta_{x,new}^{(i)}]$; $\Sigma_t^{(i)} = \sigma_0^2 I_2$; $w_t^{(i)} = w_0$
        end for
    end if
end for