Structure-Aware Keypoint Tracking for Partial Occlusion Handling

Wassim Bouachir and Guillaume-Alexandre Bilodeau
LITIV lab., Department of Computer and Software Engineering,
École Polytechnique de Montréal,
P.O. Box 6079, Station Centre-ville, Montréal,
(Québec), Canada, H3C 3A7
{wassim.bouachir, gabilodeau}@polymtl.ca

Abstract

This paper introduces a novel keypoint-based method for visual object tracking. To represent the target, we use a new model combining color distribution with keypoints. The appearance model also incorporates the spatial layout of the keypoints, encoding the object structure learned during tracking. With this multi-feature appearance model, our Structure-Aware Tracker (SAT) estimates accurately the target location using three main steps. First, the search space is reduced to the most likely image regions with a probabilistic approach. Second, the target location is estimated in the reduced search space using deterministic keypoint matching. Finally, the location prediction is corrected by exploiting the keypoint structural model with a voting-based method. By applying our SAT on several tracking problems, we show that location correction based on structural constraints is a key technique to improve prediction in moderately crowded scenes, even if only a small part of the target is visible. We also conduct comparison with a number of state-of-the-art trackers and demonstrate the competitiveness of the proposed method.

1. Introduction

Detecting and tracking objects in unconstrained environments are key components in many applications such as automated video surveillance and human computer interaction systems. Despite considerable progress in constructing strong appearance models and robust tracking procedures, the tracking problem remains complex due to many real life difficulties (*e.g.* appearance changes, regions with similar appearance as the target, occlusions, complex object motion, etc.). In this work, we address the problem of tracking an object with arbitrary motion, with no prior knowledge other than its state in the first video frame. This is often called *model-free* tracking [2, 15]. With model-free trackers, the only available input is the target image region manually annotated in the first frame using a geometric shape. Tracking is thus a challenging task due to (1) the lack of sufficient information on object appearance (only one example is available), (2) the inaccuracy in distinguishing the target from the background (a geometric shape generally contains the object and some background), and (3) the inevitable object appearance change over time.

Generally, a tracking algorithm includes two main components: the appearance model that represents the object characteristics, and the search strategy to predict the target state on every processed frame. In the proposed tracking algorithm, our appearance model includes color features for coarse estimation of the target state, and keypoints to add invariance and for encoding the object structure. In our search strategy, we use probabilistic tracking in conjunction with deterministic keypoint matching to provide a preliminary estimate of the target state. Object structural constraints are then applied to find an accurate prediction, taking into account the spatial disposition of keypoints. Our approach for representing the target structure is related to previous works on *context tracking* [10, 17, 25–27]. The main idea of these works is to focus not only on the tracked object, but also on its spatial context including other elements of the scene for which motion is correlated with the target. While our approach is inspired by the idea of context tracking, the spatial constraints of our model represent instead the geometric structure of the target. The main contributions and differences of our work from previous works are: (1) the explicit use of the spatial keypoint layout learned online to define a set of internal structural features for the target, and (2) a threefold search strategy, including a coarse estimation, a preliminary prediction, and a correction step. Experiments show the efficiency of our search strategy, demonstrating that the use of our structural model leads to a substantial improvement in tracking accuracy.

2. Related works

Keypoint-based tracking methods have attracted much attention during the last decade. This is mainly due to their invariance against various image perturbation factors (*e.g.* rotation, changes in illumination, viewpoint, etc.) [23]. Moreover, they are naturally suited to handle occlusions as partial matches between points are sufficient for most tracking scenarios. In this approach, the object is modeled as a set of keypoints detected by an external mechanism (*i.e.* a keypoint detector) [5, 12]. Once the keypoints are detected, and their descriptors are computed, the object localization can be achieved according to two possible approaches: matching in the case of a generative tracker, and classification in the case of a discriminative tracker. Generative approaches store keypoint descriptors in a database. The descriptors are designed to be invariant to various perturbation factors (*e.g.* noise, scale, illumination, etc.) and can be matched with those of the target model in a nearest-neighbor fashion. Discriminative approaches consider matching as a binary classification problem: each keypoint is classified as a keypoint from the background, or a keypoint from the target model. The classifier is learned either offline or online, considering the background and the object observed under various transformations.

In recent works, some authors argued that focusing only on the target features without considering their context does not ensure the tracker robustness in real life applications [6, 10, 11, 17, 27]. For this reason, Cerman et al. [17] improved object tracking by using a *companion* which corresponds to image regions exhibiting the same motion as the tracked object. In [27], the authors propose to track multiple *auxiliary objects* defined as the spatial context that can help the target tracker. These auxiliary objects have consistent motion correlation with the tracked target and thus help to avoid the drifting problem. In [11], the authors consider the spatial relationship between the target and similar objects and track all of them simultaneously to eliminate target confusion. In addition to considering the geometric structure of the scene, their tracker uses features extracted from similar objects to enhance the target model. In a more general approach, Grabner et al. [10] introduced the *supporters* as useful features for predicting the target position. 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 demonstrated that motion coupling of supporters allows locating the target even if it is completely occluded. In a later work, Dinh et al. [6] used supporters for context tracking, and added the notion of *distracters* which are regions co-occurring with the target while having a similar appearance. In this way, their tracker explicitly handles situations where several objects similar to the target are present.

Although context tracking methods expand the target model and maximize the use of available information in the scene, finding the motion correlation between the target and surrounding object is a costly task that often requires analyzing the whole image in every frame. While our Structure-Aware Tracker (SAT) is inspired by context tracking, our idea and motivation greatly differ from those described above. In fact, the proposed 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 visible (unoccluded) keypoints. 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 color, keypoints, and spatial constraints) allowing to track targets with changing appearance and non-rigid structures.

3. Tracking method

3.1. Motivation and overview

Figure 1 illustrates the core idea of our Structure-Aware Tracker, showing its functioning in a situation of partial occlusion. Particle filtering is firstly applied to reduce the search space and to provide a coarse estimation of the target. Keypoints are then detected on the most likely regions (defined by the best particles as shown in figure 1a), and matched with those of the target model to provide a preliminary estimate of the target location. The preliminary estimate corresponds to the particle with the highest matching score as shown in figure 1b. Since the selection of the best particle considers only the number of local features matched with the target model, the preliminary estimate does not guarantee that the best particle is located accurately. An example is shown in figure 1b where only a few number of local features were matched because the target face is partially occluded by another face (only four keypoints are visible). As a consequence, about 40 % of the circular region (target location estimate) contains background pixels. Our intuition is that knowing the internal structure of the target, we can perform a correction step where a set of structural constraints is applied to find an accurate prediction and avoid tracker drift. In practice, the correction is carried out by a voting mechanism where the available keypoints determine a more accurate target position (figure 1c and figure 1d). Once a good prediction is achieved, the target model is updated and adapted to appearance and structure changes. The keypoint set is thus re-evaluated based on the following two properties:

- the individual keypoint persistence reflected by its weight value;

(a) Reducing the search space. (b) Preliminary prediction. (c) Applying structural constraints. (d) Correcting the prediction.

Figure 1: Illustration of partial occlusion handling in a moderately crowded scene. The tracked face is partly occluded in the middle of the scene.

- the spatial consistency of the keypoint that depends on the motion correlation with the target center;

These two properties are used as voting parameters for each keypoint of the model. If a keypoint of the background is erroneously included in the target model, these two properties will reduce the effect of its vote. Moreover, the keypoint will be removed from the model if its persistence indicator decreases significantly. The SAT algorithm steps are explained in more details in the following subsections.

3.2. Reducing the search space

Appearance model. The target model describes a circular region using two types of features: (1) the RGB color probability distribution, and (2) the target keypoints. When constructing the m-bin color histogram $\hat{\mathbf{q}} = \{\hat{q}\}_{u=1...m}$, some parts of the background may be included in it, as some background pixels may lie inside the circular region. To reduce their effect in the probability distribution calculation, we use a kernel function $k(x)$ that assigns smaller weights to pixels farther from the kernel center. The color histogram is thus computed for the h pixels inside the circular 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: $k(d_i) = {}^{70}/_{81}(1 - d_i^3)^3$. To ensure a more robust and distinctive feature set, the target model also includes SIFT keypoints [16] detected in the object region. Including keypoints in the target model ensures robustness against noise, scale changes, and lighting condition variations. In addition, 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 features were successfully used for distinguishing between multiple instances of the same object such as in the face recognition problem [7, 21, 22]. In this way, we implicitly handle situations where objects of the same category of the target co-occur (*e.g.* tracking a face in the presence of several faces), and thus we avoid using an additional mechanism to track and distinguish distracters as in [6].

Probabilistic search. Once the target is modeled in the first frame, the search is guided on every subsequent frame by a probabilistic particle filtering. In our method, each particle is a circular region characterized by its color distribution as explained before. To evaluate the similarity between the reference color model $\hat{\mathbf{q}}$ and the color distribution $\hat{p}_t^{(n)}$ of a generated particle $s_t^{(n)}$, we define the distance between the two distributions as:

$$d(\hat{\mathbf{q}}, \hat{p}_t^{(n)}) = \sqrt{1 - \rho[\hat{q}, \hat{p}_t^{(n)}]} \tag{2}$$

where

$$\rho[\hat{q}, \hat{p}_t^{(n)}] = \sum_{u=1}^{m} \sqrt{\hat{q}_u . \hat{p}_{u,t}^{(n)}} \tag{3}$$

is the Bhattacharyya coefficient between $\hat{\mathbf{q}}$ and $\hat{p}_t^{(n)}$. The particle weights are thus updated in each iteration, depending on their color similarities with the target model. The area covered by the N^* best particles in the image (*i.e.* the particles having the highest weights) represents a coarse estimation of the target state, and thus constitutes a reduced search space (see figure 1a) where keypoints will be detected and matched.

3.3. Tracking keypoints

The second stage of the tracking algorithm relies on keypoints. In our work, we use SIFT as a keypoint detector and descriptor. In addition to their distinctiveness, SIFT features are reasonably invariant to changes in illumination, image noise, rotation, scaling, and changes in viewpoint between two consecutive frames [13, 14, 23]. For each subsequent frame, 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 based on the Euclidian distance. Similarly to the criterion used in [16], 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 having a distance ratio less than 0.7. Given the final set of matched pairs, we consider the particle having the highest number of matched features (figure 1b). The structural constraints of this region are then applied to provide an accurate estimation of the target location as described in the following.

3.4. Applying internal structural constraints

In this stage, we aim to apply a learned structural model of the target to predict more accurately its position. The model is learned from reliable measurements (*i.e.* when good tracking is achieved), and the internal structural properties can be considered as a part of the object appearance model.

Internal structural model. The target keypoints are stored in a reservoir of features that we denote RF. 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 keypoints location with respect to the target region center;
- w is the keypoints 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. Each matched keypoint f that is located on the particle selected in the second step (section 3.3) 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:

$$P(\mathbf{x}|f) \propto \frac{1}{\sqrt{2\pi|\Sigma|}}\, exp\left(-0.5\,(\mathbf{x}-\mu)^\top \Sigma^{-1}(\mathbf{x}-\mu)\right). \quad (4)$$

The probability of a given pixel in the voting space is estimated by accumulating the votes of keypoints weighted by their persistence indicators w. More formally, the probability for a given pixel position $\mathbf{x}$, in the voting space at time t, is estimated by:

$$P_t(\mathbf{x}) \propto \sum_{i=1}^{|RF|} w_t^{(i)} P_t(\mathbf{x}|f^{(i)}) \mathbb{1}_{\{f^{(i)} \in F_t\}}, \quad (5)$$

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. Thus, the target position is found simply by analyzing the voting space and selecting its peak to obtain the corrected target state.

Update. It has been previously shown that an adaptive target model, evolving during tracking, is the key to good performance [19]. In our algorithm, the target model (including color, keypoints, and structural constraints) is updated every time we achieve good tracking. Our definition of good tracking is inspired by the Bayesian evaluation method used in [3], referred as *histogram filtering*. Using the target histogram $\hat{\mathbf{q}}$ and the background histogram $\hat{\mathbf{q}}_{bg}$ (calculated for the area outside the reduced search space), we compute a filtered histogram $\hat{\mathbf{q}}_{filt} = \hat{\mathbf{q}}/\hat{\mathbf{q}}_{bg}$ at each iteration. It represents the likelihood ratio of pixels belonging to the target. The likelihood ratio is used to calculate a backprojection map on the target region. Quality evaluation is done by analyzing the backprojection 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:

$$\hat{q}_t = (1-\alpha)\hat{q}_{t-1} + \alpha\hat{q}_{new} \quad (6)$$

$$\hat{q}_{bg,t} = (1-\alpha)\hat{q}_{bg,t-1} + \alpha\hat{q}_{bg,new} \quad (7)$$

$$w_t^{(i)} = (1-\alpha)w_{t-1}^{(i)} + \alpha\mathbb{1}_{\{f^{(i)} \in F_t\}} \quad (8)$$

$$\Delta_{x,t}^{(i)} = (1-\alpha)\Delta_{x,t-1}^{(i)} + \alpha\Delta_{x,new}^{(i)} \quad (9)$$

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

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

$$\Sigma_t^{(i)} = (1-\alpha)\Sigma_{t-1}^{(i)} + \alpha\Sigma_{new}^{(i)}, \quad (11)$$

Algorithm 1 Predicting the target position

- initialize RF, $\hat{q}$, $\hat{q}_{bg}$
for all $frames$ **do**
  - reduce the search space (sec. 3.2)
  - detect and match keypoints with RF (sec. 3.3)
  - predict preliminary target state (sec. 3.3)
  for all $voting\_space\_positions$ **x do**
    for all $matched\_features$ $(f^{(i)} \in F_t)$ **do**
      - estimate $P(\mathbf{x}|f^{(i)})$: (Eq. 4)
    end for
    - estimate location probability $P(\mathbf{x})$: (Eq. 5)
  end for
  if $(update\_condition == true)$ **then**
    -update $\hat{q}$ and $\hat{q}_{bg,t}$: (Eq. 6 & 7)
    for all $matched\_features$ $(f^{(i)} \in F_t)$ **do**
      - update $\mu_t^{(i)}$ (Eq. 9 & 10)
      - update $\Sigma_t^{(i)}$ (Eq. 11) and $w_t^{(i)}$ (Eq. 8)
    end for
    -remove inconsistent 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

where the new correlation estimate is:

$$\Sigma_{new}^{(i)} = (\mu_{new}^{(i)} - \mu_t^{(i)})(\mu_{new}^{(i)} - \mu_t^{(i)})^\top, \qquad (12)$$

with $\mu_t^{(i)} = [\Delta_{x,t}^{(i)}, \Delta_{y,t}^{(i)}]$. We finally 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 proposed algorithm is summarized in Alg. 1.

4. Experiments

We performed two sets of experiments to evaluate the performance of our SAT tracker. Firstly, we compared the performance of the SAT tracker with the results of known state-of-the-art methods that use various tracking techniques. In the second set of experiments, comparison is done with a version of our tracker that does not impose structural constraints. In this intermediate version, the predicted state corresponds to the preliminary prediction as shown in figure 1b. This allows us to evaluate the effectiveness of the structural model, especially for tracking faces under partial occlusion and in moderately crowded scenes. To measure the precision P (*i.e.* the success rate), we firstly calculate for each frame the score $S = \frac{area(P_r \cap G_r)}{area(P_r \cup G_r)}$, where P_r is the predicted target region and G_r is the ground truth target region. For a given frame, the tracking is considered as a success if the score S is larger than 0.5. The evaluation of tracking error E is based on the relative position errors between the center of the tracking result and that of the ground truth. The results are averaged over five runs in all our experiments.

4.1. Comparison with state-of-the-art methods

A comparison with several state-of-the-art trackers is presented in this section. We tested the SAT tracker on publicly available challenging video sequences taken from [1, 2, 4, 24]. We compared its performance with the trackers: OAB [8], semiB [9], Frag [1], MIL [2], and the L1-tracker [20]. The experimental results for the compared methods were obtained by using default parameters provided by the authors. The quantitative comparison is shown in table 1.

Our experiments on the video clips *David indoor* and *girl* show the robustness of the SAT tracker when tracking a human face under large camera motion, and severe background and illumination changes. For the *David indoor* video, our tracking was successful in practically all the video frames, and the SAT tracker achieved the best result with a precision of 100%. This is because our structural model allows interpolating reliably the position of the target face during its displacement, and under camera motion. For the *girl* video, 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 invisible, but tracking is recovered as soon as the girl's face reappears.

As expected, the results of the *occluded face 1* sequence show that our tracker outperforms all the other methods because it is specifically designed to handle partial occlusions via its structure-based model. Frag is also designed to handle occlusion using a fragment-based model, but SAT tracker was significantly more accurate with a precision of 100%. This highlights the importance of using structural constraints defined by keypoint regions that are more invariant than other types of patches.

The video sequences *tiger1*, *tiger2*, and *Sylvester* show moving stuffed animals undergoing pose variation, lighting changes, scale variations with frequent occlusion level, fast motion and rotations causing motion blur. For these video sequences, the OAB, Frag, and L1 trackers fail with less than a 50% precision. The other trackers, including SAT, were able to track the target correctly. SAT achieved a good performance for the three sequences, even if the target is often too small and does not contain a large number of keypoints.

Figure 2 presents a few tracking results for the two best trackers in our experiments (*i.e.* SAT and MIL).

Sequence	OAB		SemiB		Frag		MIL		L1		SAT	
	P	E	P	E	P	E	P	E	P	E	P	E
David indoor [24]	34	45	46	37	8	73	*61*	*23*	41	42	**100**	**10**
girl [4]	71	23	50	50	68	26	50	25	**90**	*13*	*85*	**10**
occluded face 1 [1]	*92*	*18*	40	39	52	58	78	27	84	19	**100**	**14**
tiger 1 [2]	25	43	28	39	19	39	**58**	**15**	13	48	*51*	**15**
tiger 2 [2]	44	22	17	29	13	37	*64*	*17*	12	57	**70**	**16**
Sylvester [24]	42	20	68	*14*	34	47	*73*	**11**	46	42	**79**	*14*
Cliff bar [2]	23	33	65	56	22	34	**65**	**14**	38	35	*60*	*25*
average	47	29	45	38	31	45	*64*	*19*	46	37	**78**	**15**

Table 1: Precision (P) and error (E) results for SAT and five state-of-the-art trackers: **Bold red** font indicates best results, *blue italics* indicates second best.

These screenshots demonstrate how the proposed tracker performed with three of the tested videos, and especially its superiority in situations of partial occlusion (*i.e.* sequences *occluded face 1* and *tiger 2*). In general, the proposed method performed well for all the sequences and outperformed all the other algorithms in most of the scenarios and when averaging the precision and error results over all the experiments. For all video clips, SAT achieved the best result in at least one of the used metrics (*i.e.* precision and error), except for the *Cliff bar* video where MIL had a better performance. This can be explained by the low frame rate and the excessively fast movements causing a high motion blur. This situation is illustrated in figure 3 where the target's texture changes completely and abruptly. This causes a major change in both types of the target features: the color distribution (more precisely, the grayscale distribution for the *Cliff bar* video) and keypoints for which positions and characteristics change considerably. Such significant changes between two consecutive frames do not allow a correct coarse estimate using color features, neither a keypoint-based localization. As shown in figure 3, our tracker drifts when such a situation occurs. However, tracking is recovered as soon as a match is found between the current characteristics of the target and the reference appearance model. This enabled SAT to achieve the second best result for the *Cliff bar* sequence.

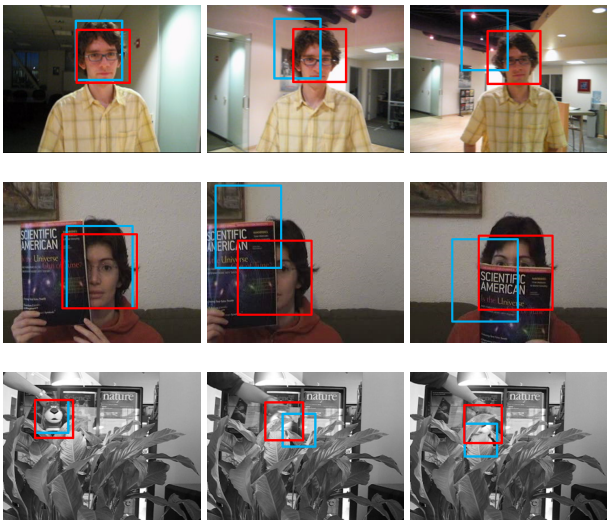

Figure 2: Tracking target location on the video sequences *David indoor, occluded face 1,* and *tiger 2*: screenshots of tracking results for SAT (red) and MIL (cyan) trackers.

(a) #76 (b) #77 (c) #78

Figure 3: Tracking on three consecutive frames of the *Cliff bar* video: due to a sudden texture change (caused by a high motion blur), SAT loses the target and snaps to some part of the background with a similar color distribution.

4.2. Face tracking and occlusion handling

The second experiment was performed on a dataset that includes video sequences captured in a laboratory room using a Sony SNC-RZ50N camera. The room was cluttered with desks, chairs, and technical video equipment in the background. The video frames are 320x240 pixels captured at a frame rate of 15 fps. For quantitative evaluation, we manually labeled the ground truth of the sequences *jp1*, *jp2*, *wdesk*, and *wbook*, with 608, 229, 709, and 581 frames respectively[1]. The purpose of this experiments is to evaluate the effectiveness of the structural model of the SAT tracker by comparison to an intermediate version that do not use structural constraints denoted no-SAT.

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

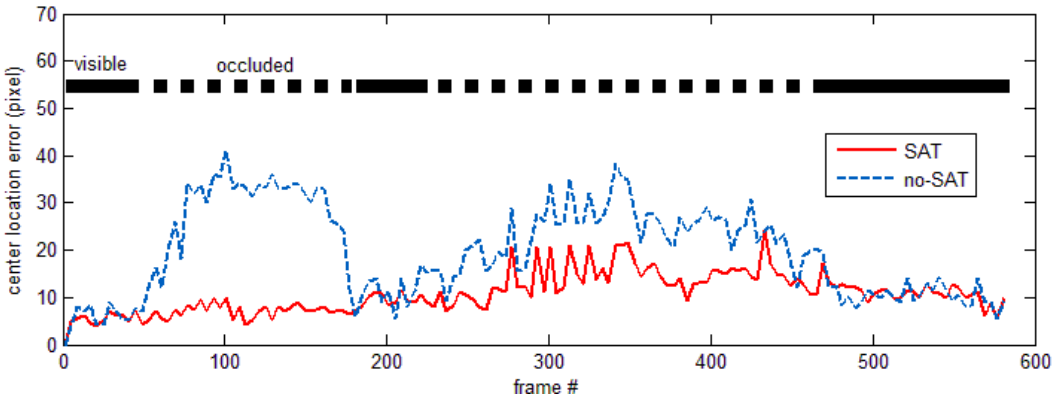

Figure 5: Center location error of SAT versus no-SAT for the *wbook* video: tracking was improved substantially by using internal structural constraints, especially when the target is partly occluded.

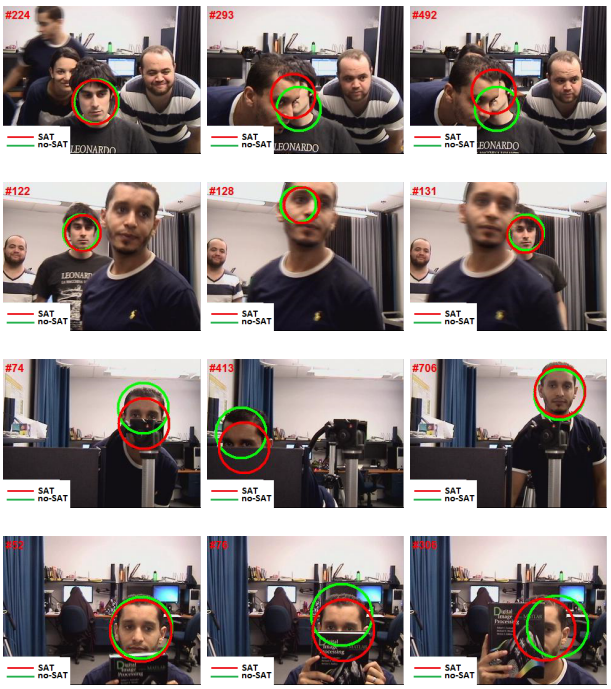

Figure 4: Tracking face location on the sequences *jp1, jp2, wdesk,* and *wbook* using SAT (red) and no-SAT (green) trackers.

Figure 4 shows tracking results on a few key frames for different scenarios where each row corresponds to a video sequence. The goal of the first video sequence *jp1* is to track a person's face in presence of other faces that may partially occlude the target. The face was successfully tracked and distinguished among the other faces in almost all the processed frames. As expected, we observed a decrease of no-SAT tracker accuracy when the target is partially occluded by another face (2nd and 3rd image in the first row). Nevertheless, the SAT tracker continues predicting the location accurately, due to the face internal structural constraints, even if only a small part of the face is visible.

In the video sequence *jp2*, we test the robustness of our algorithm for tracking a randomly moving person in a moderately crowded scene (3-4 persons). Here, we track a person's face that crosses in front or behind another walking person. As shown in the second row of figure 4, both versions can keep track of the target. In the case where the target is completely occluded by another face (second image), the tracker detects a total occlusion (since no features are matched). The tracker continues searching the target based on color similarity without updating the model. Tracking is finally recovered as soon as feature matching becomes possible again.

In the video sequence *wdesk*, we test the ability of our tracker to handle partial occlusion while the target is moving. In this video, the person hides partially behind a desk. During the partial occlusion, the target continues moving laterally and SAT tracker continues predicting accurately the face position, while no-SAT drifts to include parts of the background. This observation highlights the advantage of using structure constraints that keep correcting the prediction even if the occlusion lasts for many frames.

In the *wbook* sequence, the subject use a book to partially hide his face from different sides. With this video, we notice the high accuracy of SAT tracker, performing largely better than no-SAT in presence of long term occlusion. Figure 5 explains the obtained result, showing that the no-SAT tracker error increases during partial occlusion (frames 50 to 180, and 230 to 470). The SAT tracker continues to correctly predict the face position demonstrating a high stability in presence of severe occlusion, which allowed to obtain a precision of 98% against 71% for no-SAT. The complete results for the precision and the average location errors are shown in table 2.

	jp1		*jp2*		*wdesk*		*wbook*	
	P	E	P	E	P	E	P	E
no-SAT	85	9	94	10	70	14	71	20
SAT	**89**	**8**	**97**	**5**	**83**	**10**	**98**	**11**
improvement	4	1	3	5	7	4	27	9

Table 2: Precision (P) and error (E) results for no-SAT and SAT trackers.

5. Conclusion

This paper proposes a Structure-Aware Tracker. The appearance model includes color, keypoints, and their structural constraints. These features are learned during tracking to reflect appearance changes and incorporate new structural properties. Our experiments underline the importance of the structural model to improve tracking accuracy, and comparison with known trackers demonstrated the competitiveness of SAT.

Nevertheless, the effectiveness of our method is closely related to the properties of the keypoint detector. In fact, the target should: (1) be enough textured, and/or (2) not be too small, to allow detecting enough keypoints. For instance, a human face should have a minimum height of 30 pixels to allow detecting a sufficient number of SIFT keypoints (according to experiments done with a SNC-RZ50N camera, and not presented in this paper due to limited space). These limitations can be solved with other feature point detectors that extract points more densely (*e.g.* AGAST [18]).

Acknowledgements

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

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *CVPR*, volume 1, pages 798–805. IEEE, 2006. 5, 6

[2] B. Babenko and M.-H. Y. S. Belongie. Robust object tracking with online multiple instance learning. *TPAMI*, 2011. 1, 5, 6

[3] K. Bernardin, F. Van De Camp, and R. Stiefelhagen. Automatic person detection and tracking using fuzzy controlled active cameras. In *CVPR*, pages 1–8. IEEE, 2007. 4

[4] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *CVPR*, pages 232–237. IEEE, 1998. 5, 6

[5] W. Bouachir and G.-A. Bilodeau. Visual face tracking: A coarse-to-fine target state estimation. *International Conference on Computer and Robot Vision*, 0:45–51, 2013. 2

[6] T. B. Dinh, N. Vo, and G. Medioni. Context tracker: Exploring supporters and distracters in unconstrained environments. In *CVPR*, pages 1177–1184. IEEE, 2011. 2, 3

[7] C. Geng and X. Jiang. Face recognition using sift features. In *ICIP*, pages 3313–3316, 2009. 3

[8] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *BMVC*, volume 1, page 6, 2006. 5

[9] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV*, pages 234–247. Springer, 2008. 5

[10] H. Grabner, J. Matas, L. Van Gool, and P. Cattin. Tracking the invisible: Learning where the object might be. In *CVPR*, pages 1285–1292. IEEE, 2010. 1, 2

[11] S. Gu and C. Tomasi. Branch and track. In *CVPR*, pages 1169–1174. IEEE, 2011. 2

[12] S. Hare, A. Saffari, and P. H. Torr. Efficient online structured output learning for keypoint-based object tracking. In *CVPR*, pages 1894–1901. IEEE, 2012. 2

[13] J. Heinly, E. Dunn, and J.-M. Frahm. Comparative evaluation of binary features. *ECCV*, pages 759–773, 2012. 3

[14] L. Juan and O. Gwun. A comparison of sift, pca-sift and surf. *IJIP*, 3(4):143–152, 2009. 3

[15] Z. Kalal, J. Matas, and K. Mikolajczyk. Pn learning: Bootstrapping binary classifiers by structural constraints. In *CVPR*, pages 49–56. IEEE, 2010. 1

[16] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 3, 4

[17] J. M. LukasCerman and V. Hlavac. Sputnik tracker: Having a companion improves robustness of the tracker. In *SCIA*, volume 5575, page 291. Springer, 2009. 1, 2

[18] E. Mair, G. Hager, D. Burschka, M. Suppa, and G. Hirzinger. Adaptive and generic corner detection based on the accelerated segment test. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision ECCV 2010*, volume 6312 of *Lecture Notes in Computer Science*, pages 183–196. Springer Berlin Heidelberg, 2010. 8

[19] L. Matthews, T. Ishikawa, and S. Baker. The template update problem. *TPAMI*, 26(6):810–815, 2004. 4

[20] X. Mei and H. Ling. Robust visual tracking and vehicle classification via sparse representation. *TPAMI*, 33(11):2259–2272, 2011. 5

[21] A. Mian, M. Bennamoun, and R. Owens. An efficient multimodal 2d-3d hybrid approach to automatic face recognition. *TPAMI*, 29(11):1927–1943, 2007. 3

[22] A. Mian, M. Bennamoun, and R. Owens. Keypoint detection and local feature matching for textured 3d face recognition. *IJCV*, 79(1):1–12, 2008. 3

[23] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *TPAMI*, 27(10):1615–1630, 2005. 2, 3

[24] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *IJCV*, 77(1-3):125–141, 2008. 5, 6

[25] A. Saffari, M. Godec, T. Pock, C. Leistner, and H. Bischof. Online multi-class lpboost. In *CVPR*, pages 3570–3577, 2010. 1

[26] L. Wen, Z. Cai, Z. Lei, D. Yi, and S. Z. Li. Online spatio-temporal structural context learning for visual tracking. In *ECCV*, pages 716–729. Springer, 2012. 1

[27] M. Yang, Y. Wu, and G. Hua. Context-aware visual tracking. *TPAMI*, 31(7):1195–1209, 2009. 1, 2