



Patch-based adaptive weighting with segmentation and scale (PAWSS) for visual tracking in surgical video

Xiaofei Du^a, Maximilian Allan^b, Sebastian Bodenstedt^c, Lena Maier-Hein^d, Stefanie Speidel^c, Alessio Dore^e, Danail Stoyanov^{a,*}

^a Wellcome / EPSRC Centre for Interventional and Surgical Sciences (WEISS), University College London, UK

^b Intuitive Surgical Inc., USA

^c Karlsruhe Institute of Technology, Karlsruhe, Germany

^d Division of Computer-Assisted Medical Interventions (CAMI), German Cancer Research Center (DKFZ), Heidelberg, Germany

^e Deliveroo, London, UK

ARTICLE INFO

Article history:

Received 25 February 2018

Revised 17 December 2018

Accepted 3 July 2019

Available online 4 July 2019

Keywords:

Visual object tracking

Tracking-by-detection

Computer assisted interventions

Surgical instrument tracking

ABSTRACT

Vision-based tracking is an important component for building computer assisted interventions in minimally invasive surgery as it facilitates estimation of motion for instruments and anatomical targets. Tracking-by-detection algorithms are widely used for visual tracking, where the problem is treated as a classification task and a tracking target appearance model is updated over time using online learning. In challenging conditions, like surgical scenes, where tracking targets deform and vary in scale, the update step is prone to include background information in model appearance or to lack the ability to estimate change of scale, which degrades the performance of classifier. In this paper, we propose a Patch-based Adaptive Weighting with Segmentation and Scale (PAWSS) tracking framework that tackles both scale and background problems. A simple but effective colour-based segmentation model is used to suppress background information and multi-scale samples are extracted to enrich the training pool, which allows the tracker to handle both incremental and abrupt scale variations between frames. Experimentally, we evaluate our approach on Online Tracking Benchmark (OTB) dataset and Visual Object Tracking (VOT) challenge datasets, showing that our approach outperforms recent state-of-the-art trackers, and it especially improves successful rate score on OTB dataset, while on VOT datasets, PAWSS ranks among the top trackers while operating at real-time frame rates. Focusing on the application of PAWSS to surgical scenes, we evaluate on MICCAI 2015 challenge instrument tracking challenge and in vivo datasets, showing that our approach performs the best among all submitted methods and also has promising performance on in vivo surgical instrument tracking.

© 2019 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY license. (<http://creativecommons.org/licenses/by/4.0/>)

1. Introduction

Minimally invasive surgery (MIS) relies on endoscopic and laparoscopic video cameras to provide the surgeon with vision inside the body. Developing computer assistance for such procedures with multi-modal image overlays, robotics or novel imaging requires tracking of a variety of structures within the surgical site to estimate their motion and update their position. Visual tracking is an appealing approach for this task because it relies only on the

existing camera and it provides information within the surgeon's reference view. But visual tracking in surgical scenes involves significant challenges, especially for long term targets. Several frame samples are displayed in Fig. 1. Take the surgical instrument as a tracking target, it may disappear from the scene or be occluded by tissue via manipulation, also its appearance may significantly changes due to image blurring, bleeding, lighting and scale variations.

The key components of a successful tracking algorithm includes the target representation and how to update the representation over time. In this paper, we incorporate a Patch-based Adaptive Weighting with Segmentation and Scale (PAWSS) into tracking-by-detection, resulting a pragmatic framework, focusing on simple but effective algorithms. Given the initial position (bounding box) of a target, PAWSS divides the target into non-overlapping

* Corresponding author.

E-mail addresses: xiaofei.du.13@ucl.ac.uk (X. Du), Max.allan@intusurg.com (M. Allan), sebastian.bodenstedt@nct-dresden.de (S. Bodenstedt), l.maier-hein@dkfz-heidelberg.de (L. Maier-Hein), stefanie.speidel@nct-dresden.de (S. Speidel), alessio.dore@deliveroo.co.uk (A. Dore), danail.stoyanov@ucl.ac.uk (D. Stoyanov).

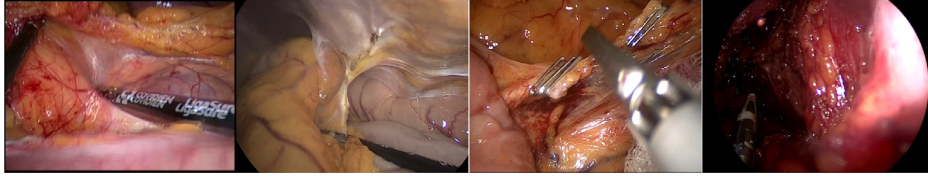


Fig. 1. Challenges of object tracking in surgical scenes, including image blur, tissue occlusion, dramatic scale and lighting variations.

patches. By using a simple but effective colour-based segmentation model, each patch is assigned with a weight which decreases background information influences within the bounding box. Besides, a two-level sampling strategy is introduced to extract multi-scale samples, which enables the tracker to handle both incremental and abrupt scale variations between frames. To reference our method to general tracking approaches, we evaluated and compared it with state-of-the-art methods on Online Tracking Benchmark (OTB) (Wu et al., 2013) and VOT challenge datasets. To show how it performs for surgical scenes, we used MICCAI 2015 instrument tracking datasets with promising results demonstrating that PAWSS is the best performing tracker, which also works in real-time without any specific code optimisation.

2. Related work

Tracking-by-detection: Recently, inspired by the success of object detection algorithms, tracking-by-detection methods has been taking inspiration from advances in machine learning, such as structured output support vector machines (SVM) (Tschantz et al., 2005), boosting (Avidan, 2007; Grabner et al., 2006), Gaussian process regression (Gao et al., 2014) and deep learning (Wang et al., 2015). Tracking-by-detection frameworks build a classifier to distinguish the tracked object from background and update this classifier with new positive observations as well as with negative information. It is inevitable that falsely labelled samples will appear and degrade the model because wrongly labelled samples of background confuse the classifier ultimately leading to drift or failure. Structured Output Tracking with Kernels (Struck) (Hare et al., 2011) adopts a structured output SVM and circumvents the traditional collection of positive and negative samples by integrating the labelling procedure within the learning process. In recent benchmark (Wu et al., 2013) Struck has shown excellent tracking performance compared to prior work.

Patch-based Representations: Recently patch-wise descriptors have been exploited to represent the object appearance (Kim et al., 2015; Chen et al., 2013; Zhang and van der Maaten, 2014). A bounding box is divided into cells or patches and low-level features are used to construct features of these patches, which represent local structural information. A major challenge for tracking-by-detection methods is that the bounding box usually not only includes the object but also some background information. Background changes differently to the moving object and causes inaccurate information transfer through the model update. To address this problem, different methods have been proposed to decrease the effects of background information such as assigning different weights based on the pixel spatial location or appearance similarity (Comaniciu et al., 2003; He et al., 2013; Lee et al., 2014). SOWP (Kim et al., 2015) exploits this concept by incorporating Random Walk with Restart (RWR) simulations to assign weights to patches. RWR simulations exploit the similarity between neighbouring patches and their relevance or self-similarity to the object appearance. Stationary distributions can be obtained to represent likelihoods that each patch belongs to either foreground or background. Patch weights are designed according to likelihoods so that

foreground patches would have relatively larger weights. We introduce a different weighting method to patches by incorporating a colour-based segmentation model. Previous papers have integrated a segmentation step into tracking (Codec et al., 2013; Duffner and Garcia, 2013), but these methods are sensitive to segmentation results since they directly track the segmented object patches free from the constraints of bounding box. By applying a segmentation step to patch weights instead we manage to enhance performance and avoid this sensitivity.

Surgical instrument tracking: For surgical instrument tracking, information from different sources has been used for instrument tracking. Typically colour, gradient or texture (Uecker et al., 1995; Cano et al., 2008) is employed to represent the appearance model. The work (Reiter and Allen, 2010) proposed to learn the instrument appearance online by combining multiple features, and explores new areas as the instrument moves in or out of view. To make feature of the instrument more distinctive, artificial markers were designed and mounted to the instrument (Wei et al., 1997; Zhang and Payandeh, 2002; Tonet et al., 2007; Zhang et al., 2017). Although attaching markers on instrument makes tracking more robust and simple, the idea of modifying instruments is usually avoided since it changes the surgical procedure. Also, artificial markers may introduce inconvenience, such as biological hazard or retrofittable difficulty. Instrument shape can be simplified or explored using a prior model to confine the search space (Pezementi et al., 2009). To classify the target from background, a random forest was learnt to classify instrument in pixel-wise fashion, then the binary classification output was used to estimate the pose of a prior 3D instrument model through optimization within a level set framework (Allan et al., 2013). Then, it was improved by combining constraints from feature points, temporal motion model with stereo setup (Allan et al., 2014). Multi-part appearance model (Allan et al., 2015) and articulated degrees-of-freedom (Allan et al., 2018) of robotic instruments can be used to align the prior model with low level optical flow constraints. In addition, cues such as robotic kinematics (Ye et al., 2016) can also be used as external constraints.

3. Proposed algorithm

3.1. Patch-based descriptor

Given the location (bounding box Ω) of the object, to represent the object appearance, we used patch-based descriptor shown in Fig. 2. Ω is evenly decomposed into n_φ non-overlapping patches $\{\varphi_i\}_{i=1}^{n_\varphi}$. Low-level feature vector $\vec{\phi}$ is extracted for each patch. Patch-based descriptor of Ω can be constructed by concatenating features of all the patches in their spatial order. Since background information is potentially included in the bounding box, we incorporate a global probabilistic segmentation model (Collins et al., 2005; Duffner and Garcia, 2013) to assign weights $\{w_i\}_{i=1}^{n_\varphi}$ to the patches based on their colour appearance, resulting a weighted descriptor:

$$\vec{\Phi}_\Omega = [w_1\vec{\phi}_1, \dots, w_{n_\varphi}\vec{\phi}_{n_\varphi}] \quad (1)$$

where w_i is the weight of the feature $\vec{\phi}_i$ of the i -th patch φ_i .

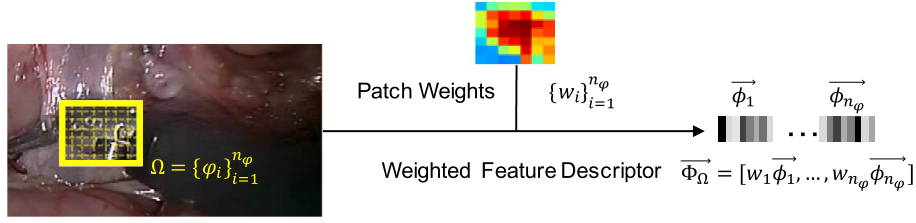


Fig. 2. Patch-based descriptor Φ_Ω . Given a bounding box Ω , it is equally decomposed into n_ϕ patches $\{\phi_i\}_{i=1}^{n_\phi}$. For the i -th patch ϕ_i , low-level feature vector ϕ_i is extracted, and is assigned with a weight w_i . Then, the descriptor Φ_Ω is constructed by concatenating features of all patches, weighted by patch weights. Note that example patch weights are shown by the highlighted bounding box. Warmer colour indicates higher weight value.

3.2. Probabilistic segmentation model for patch weighting

The global segmentation model is based on colour histogram by using a recursive Bayesian formulation to discriminate foreground and background. Let $y_{1:t}$ be the colour observation of a pixel from frame 1 to t , c be the class of a pixel. In our application, a pixel is classified as foreground ($c = 1$) or background ($c = 0$) by its colour observation. The foreground probability distribution $p(c_t = 1|y_{1:t})$ at frame t is based on tracked results from previous frames

$$\begin{aligned} p(c_t = 1|y_{1:t}) &= Z^{-1} p(y_t|c_t = 1) \\ &\sum_{c_{t-1}} p(c_t = 1|c_{t-1}) p(c_{t-1}|y_{1:t-1}) \\ p(c_t = 1|c_{t-1} = 1) &= 0.6 \quad p(c_t = 1|c_{t-1} = 0) = 0.4 \end{aligned} \quad (2)$$

where c_t is the class of a pixel at frame t : 0 for background, and 1 for foreground, and Z is a normalization constant, which can be ignored in practice. The transition probabilities for foreground and background $p(c_t|c_{t-1})$ where $c \in \{0, 1\}$ are empirical choices as in Duffner and Garcia (2013). Foreground histogram $p(y_t|c_t = 1)$ and background histogram $p(y_t|c_t = 0)$ are initialized from all the pixels inside the bounding box and from those which are surrounding the bounding box (with some margin between) in the first frame, respectively. For the following frames, the colour histogram distributions are updated using tracked result.

$$p(y_t|c_t = 1) = \delta p(y_t|y_t \in \Omega_t) + (1 - \delta) p(y_{t-1}|c_{t-1} = 1) \quad (3)$$

where $0 \leq \delta \leq 1$ is the model update factor. Ω_t represents tracked bounding box in frame t . Instead of treating every pixel equal, the weighting of a pixel also depends on the patch where it is located. Patches with higher weight are more likely to contain object pixels and vice versa. So the colour histogram update for colour observation y_t of current frame t is defined as

$$p(y_t|y_t \in \Omega_t) = \frac{\sum_{i=1}^{n_\phi} w_{i,t-1} N_{y_t \in \phi_{i,t}}}{\sum_{i=1}^{n_\phi} w_{i,t-1} \sum_{x_t} N_{x_t \in \phi_{i,t}}} \quad (4)$$

where $N_{y_t \in \phi_{i,t}}$ represents the number of pixels with colour observation y_t in the i -th patch $\phi_{i,t}$ in frame t , and x_t represents any colour observation in frame t , so the denominator means the weighted number of all the pixel colour observations in the bounding box Ω_t .

The weights $w_{i,1}$ for all the patches are initialized as 1 at the first frame, and then are updated based on the segmentation model

$$w_{i,t} = \delta \bar{w}_{i,t} + (1 - \delta) w_{i,t-1} \quad (5)$$

$$\bar{w}_{i,t} = \frac{\varpi_{i,t}}{\max_{1 \leq i \leq n_\phi} \varpi_{i,t}} \quad (6)$$

$$\varpi_{i,t} = \frac{\sum_{x_t} p(x_t|c_t = 1) N_{x_t \in \phi_{i,t}}}{\sum_{x_t} N_{x_t \in \phi_{i,t}}} \quad (7)$$

where $\varpi_{i,t}$ denotes the average foreground probability of all pixels in the patch $\phi_{i,t}$ in the current frame t , it is normalized so the highest weight update $\bar{w}_{i,t}$ equals 1. The patch weight $w_{i,t}$ is then updated gradually over time. We omit background probability distribution $p(c_t = 0|y_{1:t})$ since it is similar to Eq. (2).

Unlike the weighting strategy in other patch-based methods (Chen et al., 2013; Kim et al., 2015) by analysing the similarities between neighbouring patches, our patch weighting method is simple and straightforward to implement, the weight update for each patch is independent from each other, and only relies on the colour histogram based segmentation model. We show examples of the patch weight development in Fig. 3. The patch weight thumbnails are displayed on the top corner of each frame, which indicate the objectness in the bounding box and also reflect the object deformation over time. Since we update the segmentation model based on previous patch weights, and in turn the segmentation model facilitates updating the weight of all patches. This co-training strategy enhances the weight contrast between foreground and occluded patches, which suppresses background information efficiently.

3.3. Two-level sampling for scale estimation

The tracked object often undergoes complicated transformations during tracking, for example, deformation, scale variations, occlusion, etc. Fixed-scale bounding box estimation is ill-equipped to capture the accurate extents of the object, which would degrade the classifier performance by providing samples which are either partial cropped or include background information.

When locating the object in a new frame, all the bounding box candidates are collected within a search window, and the bounding box with the maximum classification score is selected to update the object location. Rather than making a suboptimal decision by choosing from fixed-scale samples, we augment training sample pool with multi-scale candidates, which is referred as two-level sampling strategy (see Fig. 4). On the first level, all the bounding box samples are extracted with fixed-scale s_{t-1} (the object scale in frame $t-1$). The search window is centered at the Ω_{t-1} with a height/width of r_w , then the weighted patch-based descriptor of all candidates $\{\Omega'_t\}$ are fed into the classifier, and we select the bounding box Ω'_t with the maximum classification score not as the final decision, but as the search center for our second level. After first level, the rough location of the object is narrowed to a smaller area. We then set a smaller search window with search height/width of r_s , centring at the bounding box Ω'_t selected in the first level, and we construct multi-scale candidates $\{\Omega_t\}$ within the search window. All the samples are evaluated by the classifier, and we select the bounding box Ω_t of the sample with the maximum score as the final location of the object.

Obviously, the scales of augmented samples are critical. We consider two complementary strategies that handle both incremental and abrupt scale variations. Firstly, to deal with relatively



Fig. 3. Example patch weights are shown for the highlighted bounding box displayed in the top corner of the image. Warmer colour indicates higher foreground possibility.

small scale changes between frames, we build a scale set S_r

$$S_r = \{s | s = \lambda^m s_{t-1}\} \quad m \in \left[-\frac{n_r - 1}{2}, \dots, \frac{n_r - 1}{2} \right] \quad (8)$$

where λ is a fixed value which is slightly larger than 1.0. It is set to accurately search the scale change. n_r is the scale number in the scale set S_r . s_{t-1} is the scale of the object in frame $t - 1$ compared with the initial bounding box in the first frame. Considering object scale usually does not vary too much between frames, scale set S_r includes scales which are close to the previous frame.

Secondly, when object undergoes abrupt scale changes between frames, scale set S_r is unable to keep pace with the speed of the scale variations. To address this problem, we build an additional scale set S_p by incorporating Lucas–Kanade tracker (KLT) (Bouguet,

2001; Shi and Tomasi, 1994), which helps us estimate the scale change explicitly. We randomly pick n_{pt} points from each patch in the bounding box Ω_{t-1} of frame $t - 1$, and tracked all these points in the next frame t . With sufficient well-tracked points, we can estimate the scale variation between frames by comparing the distance changes of the tracked point pairs.

We illustrated the scale estimation by KLT tracker in Fig. 5. Let p_{t-1}^i denotes one picked point in the previous frame $t - 1$ and its matched point p_t^i in the current frame t . We compute the distance d_{t-1}^{ij} between point-pair (p_{t-1}^i, p_{t-1}^j) , and the distance d_t^{ij} between the matched point-pair (p_t^i, p_t^j) . For all the matched point pairs, we compute the distance ratio between the two frames

$$V = \{s | s = d_t^{ij} / d_{t-1}^{ij}\} \quad i \neq j \quad (9)$$

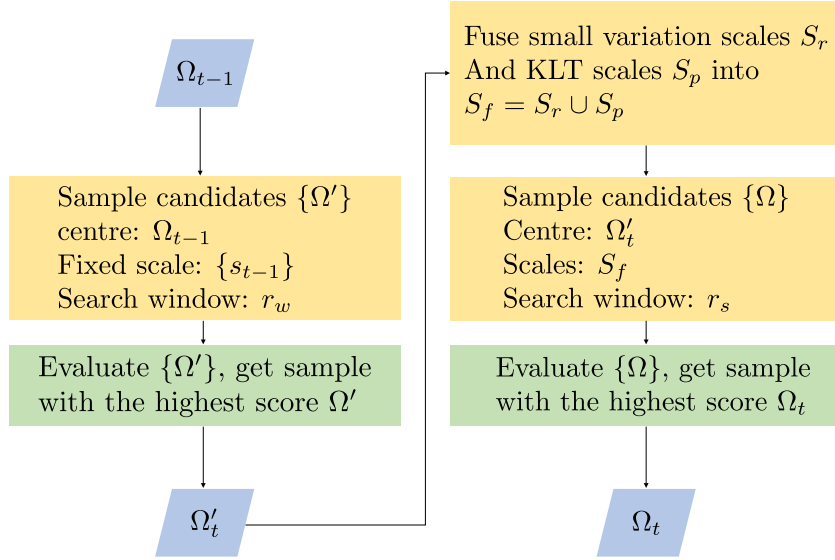
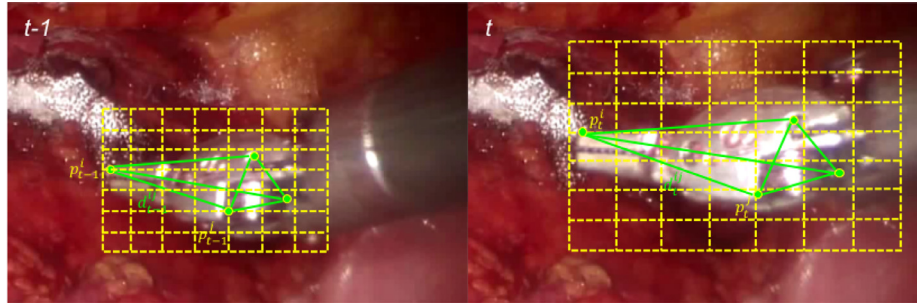


Fig. 4. Two-level sampling strategy workflow.

Fig. 5. Illustration of scale estimation by using the KLT tracker. Random points located on the patches are picked in frame $t - 1$, and are tracked in the next frame t by the KLT tracker, the distance ratio of point pairs (p^t, p^t) between two frames are used for scale estimation. We use 7×7 patch grids, resulting $n_p=49$ in the illustration.

where V is the set with all the distance ratios. We sort V by value and pick the median element $s_p = V_{\text{sorted}}(\frac{n}{2})$ as the potential scale change of the object. To make scale estimation more robust, we uniformly sample the scales ranging between $[1, s_p]$ or $[s_p, 1]$ to construct the scale set S_p .

$$S_p = \left\{ s | s = 1 + i \frac{s_p - 1}{n_p - 1} \right\} \quad 0 \leq i < n_p \quad (10)$$

where n_p is the scale number in the scale set S_p . When the object is out-of-view, occluded or abruptly deforms, the ratio of well-tracked points will be low. In that case, the estimation from the KLT tracker will be unreliable. In our implementation, when the ratio is lower than 0.5, we then set $s_p = 1$, therefore the scale set S_p will only add samples with the previous scale into the candidate pool. Only when there are enough points well tracked, the estimation from the KLT tracker will be trusted. We fuse these two complementary scale sets S_r and S_p into $S_f = S_r \cup S_p$ to enrich our sample candidate pool. To show the effectiveness, we evaluate our proposed tracker in Section 4 with or without scale set S_p estimated by the KLT tracker.

3.4. Tracking framework

PAWSS can be combined with any tracking-by-detection method. We show the pipeline of the whole framework in Fig. 6. It includes two phases: *evaluation* and *learning*. The evaluation phase is to find the target in a new frame. Given the bounding box Ω_{t-1}

in the previous frame $t - 1$, sample candidates are extracted in a search window, which centers at Ω_{t-1} in the current frame t , unlike other tracking-by-detection approaches, we adapt a two-level sampling strategy for accurate scale estimation (Section 3.3). Via the colour-based segmentation model, weights of all patches are updated as in Section 3.2, and the descriptors of all samples are computed via patch weighting. Descriptors of all samples are fed into classifier and the one with the highest output score is picked as the best sample. The location Ω_t of the best sample shows where the target is in the current frame at time t . Between frames, the target appearance changes due to deformation, occlusions, light and scale variations, therefore, the classifier and the segmentation model needs to be learnt online to keep up with the changes. The best sample among all samples represents the most similar one compared to the target. For one thing, pixel colour distribution of the best sample is used to update the segmentation model. For another, samples are extracted around the best sample in order to collect foreground and background information. Descriptors of all samples are computed and used to train the classifier online to better discriminate the target from neighbouring background. The procedure starts again for the next frame.

In our implementation, we incorporate PAWSS into Struck Hare et al. (2011). The algorithm relies on an online structured output SVM learning framework which integrates learning and tracking. It directly predicts the location displacement between frame, avoiding the heuristic intermediate step for assigning binary labels to training samples, which achieves top performance in OTB dataset Wu et al. (2013).

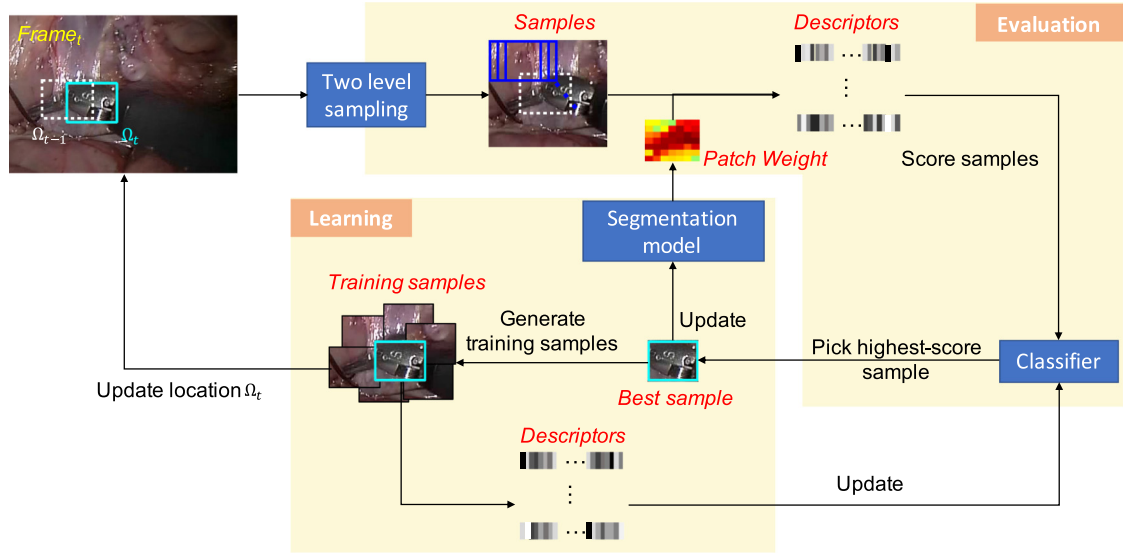


Fig. 6. Tracking framework. Given the target location Ω_{t-1} in the previous frame at time $t - 1$, the framework is to predict the target location Ω_t in the current frame at time t . The framework includes evaluation and learning phases. In evaluation phase, multi-scale samples are extracted via two-level sampling strategy, and then are fed into the classifier to pick the one with the highest score. The location of the sample is considered as the new location Ω_t . The sample is also used for updating the segmentation model and the classifier in the learning phase.

Table 1

Parameter setting of the framework in all experiments.

Number of patches n_ϕ	$7 \times 7 = 49$
Base of scale estimation λ	1.003
Number of scales for small scale changes n_r	11
Number of scales for abrupt scale changes n_p	11
Updating factor of classifier η	0.3
Updating factor of segmentation model δ	0.1

4. Results

Implementation details: Our algorithm is publicly available online¹ and is implemented in C++ and performs at about 7 frames per second with an i7-2.5 GHz CPU without any optimisation. We listed the parameter setting in Table 1. To illustrate the generalization of our proposed framework, we use the same parameter setting through all experiments. For structured output SVM, we are using a linear kernel and the parameters are empirically set as $\delta = 0.1$ in Eqs. (3) and (5), $\lambda = 1.003$ in Eq. (8), the scale numbers of the scale set are $n_r = n_p = 11$. The number of extracted points from each patch $n_{pt} = 5$. The updating threshold for the classifier is set as $\eta = 0.3$. For each sequence, we scale a frame to make sure the minimum side length of the bounding box is larger than 32 pixels, and the search window r_w is fixed to $(W + H)/2$, where W and H represents the width and height of the scaled bounding box, respectively, and the search window r_s is fixed to 5 pixels. We tested different low-level feature combinations in Section 4.1 and found that the combination of HSV colour and gradient features (HSV+G) achieves the best results. The patch number affects the tracking performance, too many patches increase the computation and too less patches do not robustly reflect the local appearance of the object. We tested different patch numbers, and selected $n_\phi = 49$ to strike a performance balance.

4.1. Online Tracking Benchmark (OTB)

OTB dataset (Wu et al., 2013) includes 50 sequences tagged with 11 attributes, which represent the challenging aspects for

tracking such as illumination variation, occlusion, deformation et al. The tracking performance is quantitatively evaluated using both precision rate (PR) and success rate (SR), as defined in (Wu et al., 2013). PR/SR scores are depicted using precision plot and success plot, respectively. The precision plot shows the percentage of frames whose tracked centre is within certain Euclidean distance (20 pixels) from the centre of the ground truth. Success plot computes the percentage of frames whose intersection over union overlap with the ground truth annotation is within a threshold varying between 0 and 1, and the area under curve (AUC) is used for SR score. To evaluate the effectiveness of incorporating the scale set proposed by the KLT tracker, we provide two versions of our tracker as PAWSSa and PAWSSb: PAWSSa only includes scale set S_r , while PAWSSb includes both S_r and S_p for scale estimation.

Comparison using different features: Selecting right features to describe the object appearance plays a critical role in tracking. The most desirable feature property is its uniqueness so that the object can be distinguished from background. Raw intensities or colour features are usually used for histogram-based appearance representations, while edge or gradient information are less sensitive to illumination changes. Generally, many tracking approaches use a combination of these diverse features to represent the object (Hare et al., 2011; Grabner et al., 2006; Li et al., 2013). To evaluate the performance of our proposed approach, we tested different low-level features such as HSV colour, RGB colour, the combination of colour and gradient features (HSV+G, RGB+G) for constructing the descriptor in Table 5.1. The RGB histogram is 24-dimensional with 8 bins for each channel, and the HSV colour histogram is 20-dimensional including 8 bins for H and S channels respectively and 4 separate bins for V channel. The gradient histogram is 16-dimensional signed gradients ranging from 0 to 360°. We also compared our tracker PAWSSa and PAWSSb with Struck (Hare et al., 2011) and SOWP (Kim et al., 2015). From Table 2, we observe: Augmenting colour with gradient histogram improves the tracking performance by providing diverse structural information of the object. In our experiments, the descriptor comprising combination of HSV colour and gradient features achieves the best results, we use this setting in the following evaluation.

Comparison with state-of-the-art trackers: We use the evaluation toolkit provided by Wu et al. (2013) to generate the

¹ <https://github.com/surgical-vision/PAWSS>.

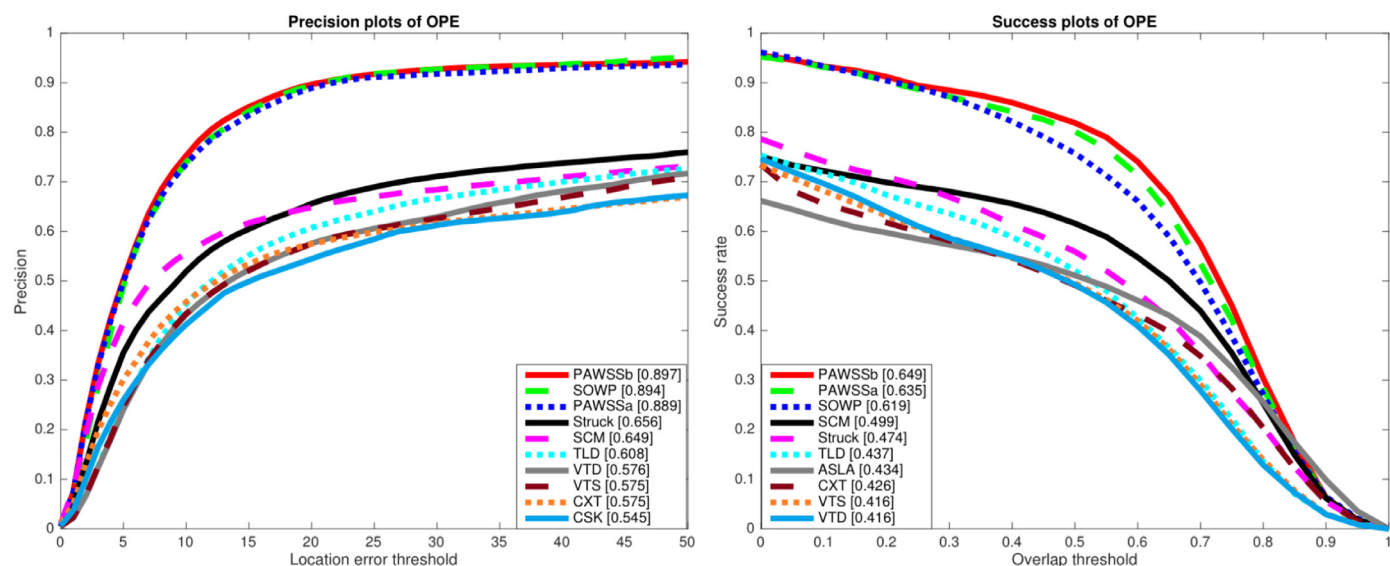


Fig. 7. Comparison of precision and success plots on OTB with the top 10 trackers; PR scores are illustrated with the threshold at 20 pixels and SR scores with the average overlap (AUC) in the legend.

Table 2

The performance of the proposed algorithm compared with different low-level features. PAWSSa and PAWSSb tracker represents our tracker without and with the KLT tracker, respectively.

	PAWSSa	PAWSSb
HSV	0.731 / 0.528	0.742 / 0.545
RGB	0.764 / 0.552	0.749 / 0.544
RGB+G	0.838 / 0.605	0.840 / 0.607
HSV+G	0.889 / 0.635	0.897 / 0.649

precision and success plots for the one pass evaluation (OPE) of the top 10 algorithms in Fig. 7. The toolkit includes 29 benchmark trackers, besides that we also include SOWP tracker. It is shown that PAWSSb achieves the best PR/SR scores among all the trackers. For a more detailed evaluation, we also compared our tracker with state-of-the-art trackers in Table 3. Notice that in all the attribute field, our tracker achieves either the best or the second best PR/SR scores. Our tracker achieves 36.7% gain in PR and 36.9% gain in SR over Struck (Hare et al., 2011). By using a simple patch weighting strategy and training with adaptive scale samples, the performance shows that our tracker provides comparable PR scores, and higher SR score compared with SOWP (Kim et al., 2015). PAWSSa tracker improves SR score by 2.6% considering gradually small changes between frames, PAWSSb improves SR score by 4.8% by incorporating scales estimated by the external KLT tracker. Specifically, when the object undergoes scale variation PAWSS achieves a performance gain of 10.3% in SR over SOWP.

We show tracking results in Figs. 8 and 9 with the top trackers including TLD (Kalal et al., 2012), SCM (Zhong et al., 2012), Struck (Hare et al., 2011), SOWP (Kim et al., 2015) and the proposed PAWSSa and PAWSSb. In Fig. 8, five challenging sequences are selected from the benchmark dataset, which include illumination variation, scale variations, deformation, occlusion or background clusters. PAWSS can adapt when the object deforms in a complicated scene and track the target accurately. In Fig. 9, we select five representative sequences with different scale variations. PAWSS can well track the object with scale variation, while other trackers drift away. The results show that our proposed tracking framework PAWSS can track the object robustly through sequence

by using the weighting strategy to suppress background information within the bounding box, and also by incorporating scale estimation allowing the classifier to train with adaptive scale samples. Please see the supplementary video for more sequence tracking results.

4.2. Visual Object Tracking (VOT) challenges

For completeness, we also validated our algorithm on VOT2014 (25 sequences) and VOT2015 (60 sequences) datasets. VOT datasets use ranking-based evaluation methodology: Accuracy and robustness. Similar to SR rate for OTB dataset, the accuracy measures overlap of the predicted result and the ground truth bounding box, while the robustness measures how many times the tracker fails during tracking. A failure is indicated whenever the tracker loses the target object which means the overlap becomes zero, and it will be re-initialized afterwards. All the trackers are evaluated, compared and ranked based on with respect to each measure separately using the official evaluation toolkit from the challenge.²

VOT2014 VOT2014 challenge includes two experiments: Baseline experiment and region-noise experiment. In baseline experiment, a tracker runs on all the sequences by initializing with the ground truth bounding box on the first frame; while in the region-noise experiment, the tracker is initialized with a random noisy bounding box with the perturbation in the 10% of the ground truth bounding box size. (Kristan et al., 2015b). The ranking plots with 38 trackers are shown in Fig. 10 for comparing PAWSS with the top three trackers: DSST (Danelljan et al., 2014), SAMF (Li and Zhu, 2014), KCF (Henriques et al., 2015) in Table 4. For both the experiments our PAWSS has lower accuracy score 0.58/0.55, but less failures 0.88/0.78 and have a second average rank. But considering the tracking process of the experiments: once a failure is detected, the tracker will be re-initialized, to eliminate the effect of achieving higher accuracy score by more re-initialization steps, we performed experiments without the re-initialization, also shown in Table 4. The results show that PAWSS has the highest accuracy score 0.51/0.48 among all the trackers without re-initialization, which means it is more robust than the other trackers.

² <http://www.votchallenge.net/>.

Table 3

Comparison of PR/SR score with state-of-the-art trackers including Struck (Hare et al., 2011), DSST (Danelljan et al., 2014), SAMF (Li and Zhu, 2014), FCNT (Wang et al., 2015) and SOWP (Kim et al., 2015) in the OPE based on the 11 sequence attributes: illumination variation (IV), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-plane rotation (OPR), out-of-view (OV), background cluttered (BC) and low resolution (LR). The best and the second best results are shown in **red** and **blue** colours respectively.

	Struck	DSST	SAMF	FCNT	SOWP	PAWSSa	PAWSSb
IV(25)	0.558 / 0.428	0.727 / 0.534	0.735 / 0.563	0.830 / 0.598	0.842 / 0.596	0.860 / 0.616	0.880 / 0.648
SV(28)	0.639 / 0.425	0.723 / 0.516	0.730 / 0.541	0.830 / 0.558	0.849 / 0.523	0.849 / 0.564	0.849 / 0.577
OCC(29)	0.564 / 0.413	0.845 / 0.619	0.716 / 0.534	0.797 / 0.571	0.867 / 0.603	0.859 / 0.618	0.872 / 0.634
DEF(19)	0.521 / 0.393	0.813 / 0.622	0.660 / 0.510	0.917 / 0.644	0.918 / 0.666	0.908 / 0.656	0.934 / 0.688
MB(12)	0.551 / 0.433	0.651 / 0.519	0.547 / 0.464	0.789 / 0.580	0.716 / 0.567	0.786 / 0.593	0.783 / 0.603
FM(17)	0.604 / 0.462	0.663 / 0.515	0.517 / 0.435	0.767 / 0.565	0.744 / 0.575	0.784 / 0.572	0.792 / 0.587
IPR(31)	0.617 / 0.444	0.691 / 0.507	0.765 / 0.560	0.811 / 0.555	0.847 / 0.584	0.860 / 0.594	0.852 / 0.600
OPR(39)	0.597 / 0.432	0.763 / 0.554	0.733 / 0.535	0.831 / 0.581	0.896 / 0.615	0.898 / 0.623	0.901 / 0.635
OV(6)	0.539 / 0.459	0.708 / 0.609	0.515 / 0.459	0.741 / 0.592	0.802 / 0.635	0.771 / 0.611	0.828 / 0.645
BC(21)	0.585 / 0.458	0.708 / 0.524	0.694 / 0.517	0.799 / 0.564	0.839 / 0.618	0.847 / 0.632	0.859 / 0.647
LR(4)	0.545 / 0.372	0.459 / 0.361	0.497 / 0.409	0.765 / 0.514	0.606 / 0.410	0.679 / 0.504	0.669 / 0.500
Avg.(50)	0.656 / 0.474	0.777 / 0.570	0.737 / 0.554	0.856 / 0.599	0.894 / 0.619	0.889 / 0.635	0.897 / 0.649

Table 4

The Accuracy (Acc.) and Robustness (Rob.) results of VOT2014 baseline and region-noise experiments with and without-re-initialization compared with the top trackers DSST (Danelljan et al., 2014), SAMF (Li and Zhu, 2014) and KCF (Henriques et al., 2015). The best and the second best results are shown in **red** and **blue** colours respectively.

	Baseline					Region-noise					Avg Rank
	Acc.		Rob.		Acc. (w/o)	Acc.		Rob.		Acc. (w/o)	
	Score	Rank	Failure	Rank	Score	Score	Rank	Failure	Rank	Score	
DSST	0.62	5.16	1.16	8.2	0.47	0.57	4.32	1.28	7.4	0.43	6.27
SAMF	0.61	4.32	1.28	8.68	0.50	0.57	4.2	1.43	8.44	0.48	6.41
KCF	0.62	3.68	1.32	8.68	0.40	0.57	4.84	1.51	9.00	0.36	6.92
PAWSSb	0.58	5.80	0.88	8.00	0.51	0.55	6.08	0.78	5.4	0.48	6.32



— TLD — SCM — Struck — SOWP — PAWSSa — PAWSSb

Fig. 8. Comparison of the tracking results of our proposed tracker PAWSS with SOWP (Kim et al., 2015) and three conventional trackers: TLD (Kalal et al., 2012), SCM (Zhong et al., 2012) and Struck (Hare et al., 2011) on some especially challenging sequences in the benchmark.

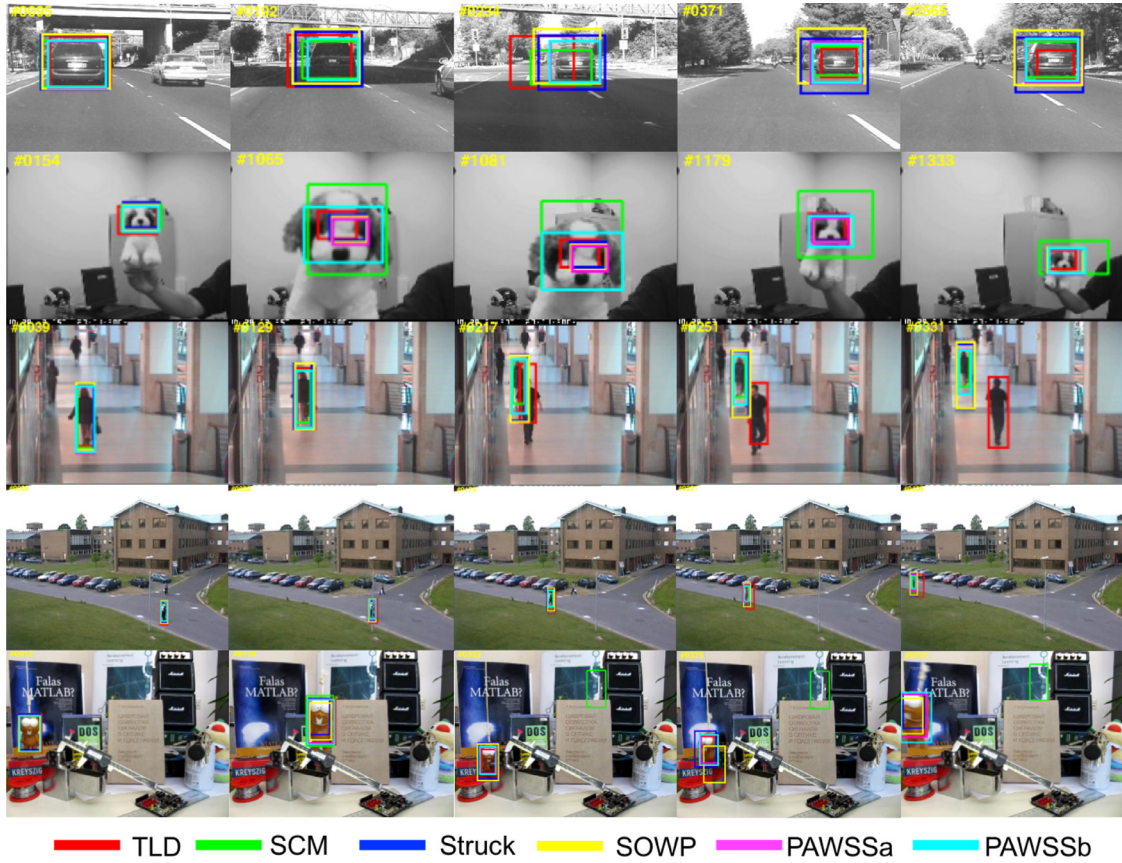


Fig. 9. Comparison of the tracking results of our proposed tracker PAWSS with SOWP (Kim et al., 2015) and three conventional trackers: TLD (Kalal et al., 2012), SCM (Zhong et al., 2012) and Struck (Hare et al., 2011) on some sequences with scale variations in the benchmark.

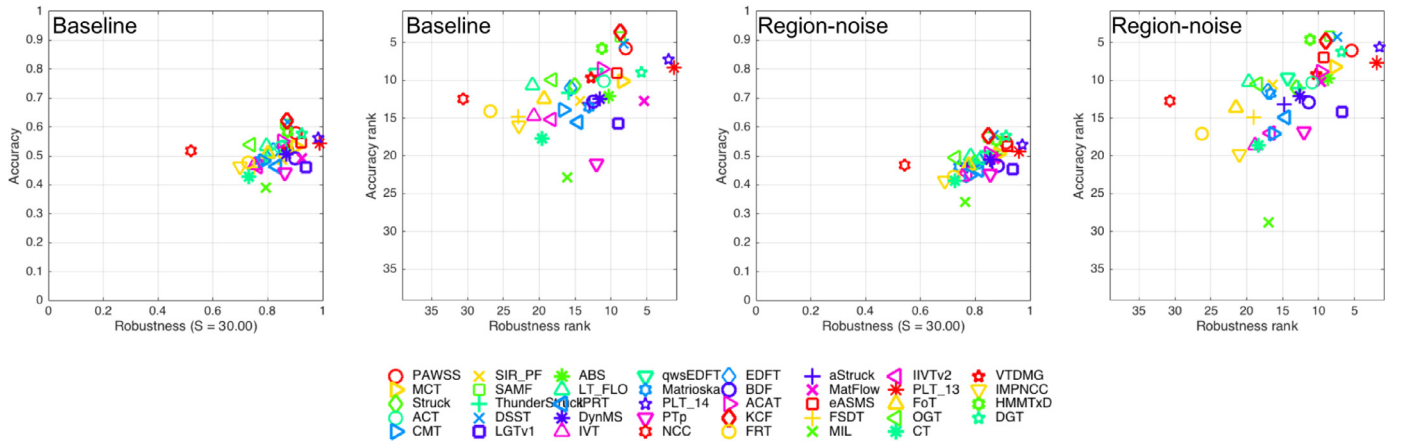


Fig. 10. The accuracy-robustness score and ranking plots with respect to the baseline and region-noise experiments of VOT2014 dataset. Tracker is better if its result is closer to the top-right corner of the plot.

VOT2015 Finally, we evaluated and compared PAWSS with 62 trackers on VOT2015 dataset. VOT2015 challenge only includes baseline experiment, and the ranking plots are shown in Fig. 11. In VOT2013 and VOT2014, average ranking measure is used to determine the performance of the trackers. Although average ranking has taken both accuracy and robustness measure into consideration, it is not theoretically representative as a concrete tracking performance. In VOT2015 (Kristan et al., 2015a), expected average overlap measure is introduced which combines both per-frame accuracies and failures in a principled manner. Compared with the average rank, expected overlap has a more clear practical interpretation.

We list the score / rank and expected overlap of the top trackers from VOT2015 (Kristan et al., 2015a) which are either quite robust or accurate, the above VOT2014 top three trackers DSST (Danelljan et al., 2014), SAMF (Li and Zhu, 2014), KCF (Henriques et al., 2015),³ and the baseline NCC tracker in Table 5 and also shown in the expected average overlap plot Fig. 11. It can be shown that the average rank is not always consistent with the expected overlap. According to the paper (Kristan et al., 2015a), a VOT2015 published *sota* bound criteria (0.2)

³ This is an improved version of the original tracker.

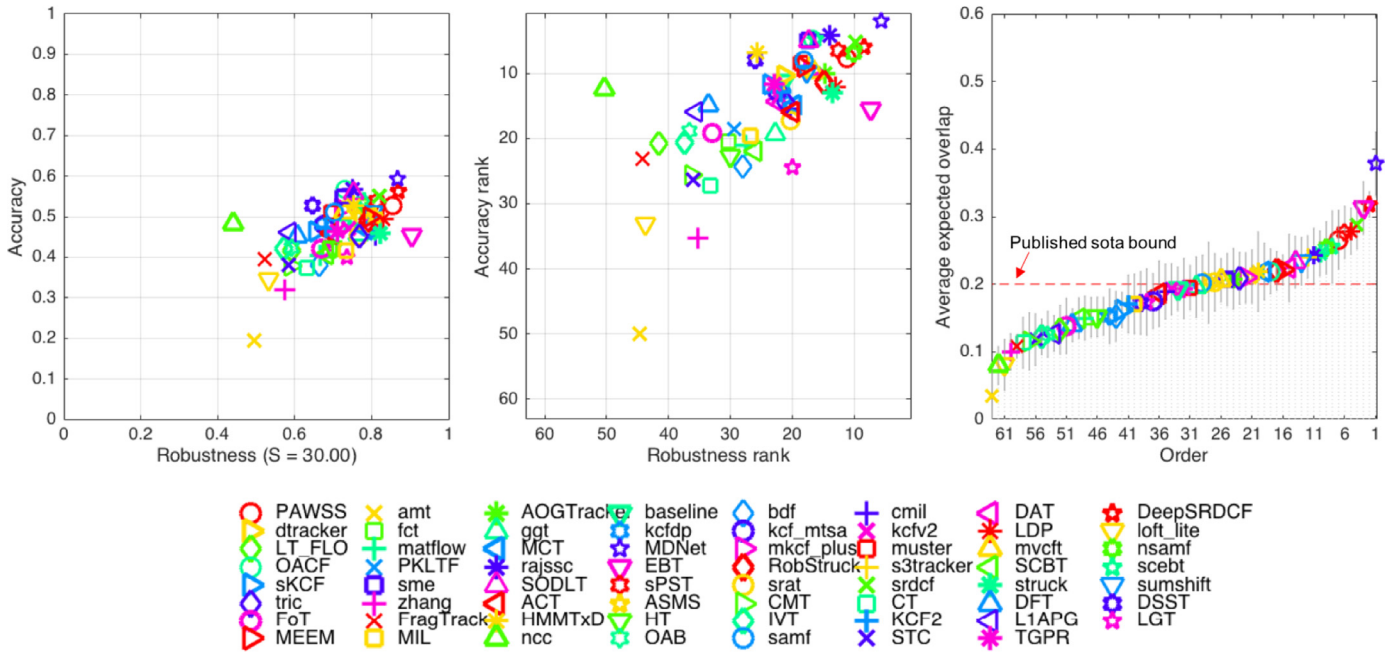


Fig. 11. The accuracy-robustness ranking plots and the expected overlap score ranking plot of VOT2015 dataset. Tracker is better if its result is closer to the top-right corner of the plot. The published sota bound is established based on top trackers in recent years. Any tracker with performance over the boundary is considered as a state-of-the-art tracker.

Table 5

VOT2015 Accuracy (Acc.), Robustness (Rob.), Score/Ranking and expected overlap results from the top trackers of VOT2014, VOT2015 and the baseline tracker. The NCC tracker is VOT2015 baseline tracker. Trackers marked with † are submitted to VOT2015 without publication.

	Baseline				Avg rank	Exp overlap
	Acc.		Rob.			
	Score	Rank	Failure	Rank		
MDNet Nam and Han (2015)	0.59	2.03	0.77	5.68	3.86	0.378
DeepSRDCF Danelljan et al. (2015a)	0.56	5.92	1.00	8.38	7.15	0.318
EBT Wang and Yeung (2014)	0.45	15.48	0.81	7.23	11.36	0.313
SRDCT Danelljan et al. (2015b)	0.55	5.25	1.18	9.83	7.54	0.288
LDP Lukežič et al. (2016)	0.49	12.08	1.30	13.07	12.58	0.279
sPST Hua et al. (2015)	0.54	6.57	1.42	12.57	9.57	0.277
PAWSSb	0.53	7.75	1.28	11.22	9.49	0.266
NSAMF†	0.53	7.02	1.45	10.1	8.56	0.254
RAJSSC Zhang et al. (2015)	0.57	4.23	1.75	13.87	9.05	0.242
RobStruck†	0.49	11.45	1.58	14.82	13.14	0.220
DSST Danelljan et al. (2014)	0.53	8.05	2.72	26.02	17.04	0.172
SAMF Li and Zhu (2014)	0.51	7.98	2.08	18.08	13.03	0.202
KCF Henriques et al. (2015)	0.47	12.83	2.43	21.85	17.34	0.171
NCC*	0.48	12.47	8.18	50.33	31.4	0.080

is established by averaging the tracker performance published in 2014/2015 from top computer vision conferences and journals. The tracker will be considered as a state-of-the-art tracker with performance over this boundary criteria. Our tracker PAWSS is well above the criteria, and is among those top trackers (ranks the 7-th, outperforming 54 trackers), also PAWSS achieves better than any of VOT2014 top trackers on VOT2015 dataset.

4.3. Surgical instrument tracking

PAWSS is a general tracking framework, we also want to evaluate its performance on both *ex vivo* and *in vivo* surgical instrument sequences. In the Endoscopic vision MICCAI2015 Challenge,⁴ one of the sub-challenge focuses on comparing differ-

ent vision-based methods for tracking conventional and articulated laparoscopic instruments in robotic surgery. The dataset has not released ground truth for test data. The official evaluation categorized conventional laparoscopic instrument test set according to the challenging factors including bleeding (C_{blood}), smoke (C_{smoke}), instrument occlusions ($C_{\text{occlusion}}$), multiple instruments (C_{multiple}) and surgical objects such as meshes and clips (C_{objects}). And the robotic laparoscopic instrument dataset includes sequences with multiple instruments (C_{multiple}). For evaluating the tracking performance, Euclidean distance of the centre point between the ground truth and the tracking result of training data is computed and compared separately for these challenging factors. We submitted our proposed method to the challenge, and obtained the performance comparison from the official report.

EndoVis Articulated Robotic Laparoscopic instrument dataset

The articulated instrument dataset is from *ex vivo* interventions,

⁴ <https://endovissub-instrument.grand-challenge.org/>.

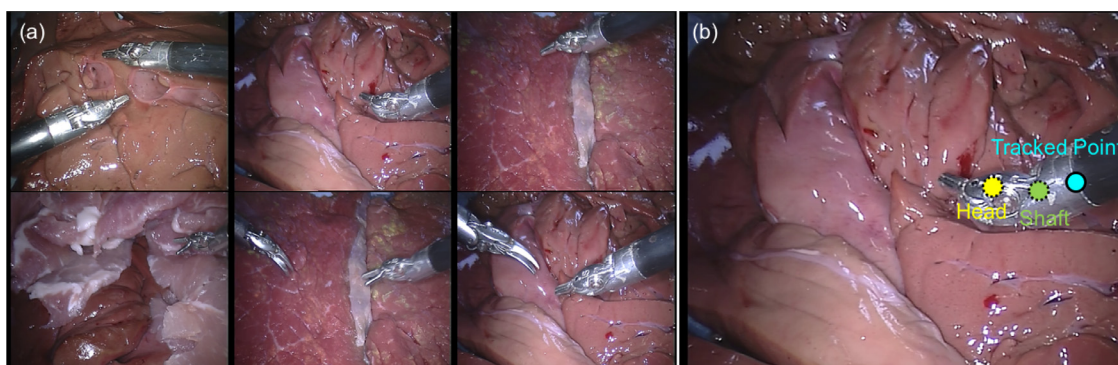


Fig. 12. (a) Example frame from each sequence of *EndoVis* articulated surgical instrument dataset; (b) The original annotation includes the position of the tracked point, in our annotation, we relabeled the tracked point and also added new annotations for the Head and Shaft points.



Fig. 13. Result example frames from each sequence of the *EndoVis* articulated robotic surgical instrument dataset. The result bounding box and centre point is represented in cyan colour, and the ground truth centre point is represented in green colour. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

and the sequences are collected using the da Vinci® (Intuitive Surgical Inc., CA) system with porcine tissue samples. Example frames from each sequence are shown in Fig. 12 (a). The dataset is divided into training and test data. Training data contains four 45 seconds surgery video sequences. For each instrument, the tracked point of the instrument is defined as the intersection between the instrument axis and the border between the shaft and the manipulator. The annotation includes pixel coordinates of the tracked point (Fig. 12 (b)). Test data is composed of 15 additional seconds video from each of the training sequence, and two additional new 60 s video sequences.

Original annotation We have summarized the frame number for each sequence and have shown the accuracy evaluation separately in the original annotation section of Table 6 and Fig. 14 Left. The accuracy is defined as the percentage of tracked frames within the error threshold. Distance (pixels) is averaged over correctly tracked frames. In Fig. 14, it shows accuracy under different threshold. In four train sequences, there are five instruments to be tracked. The average accuracy score for train data is 79.01% for 20 pixel threshold, with a distance error of 8.00 pixels. It is noted that the accuracy score (36.55% for 20 pixel threshold) for sequence 4 is relatively lower compared with the rest sequences.

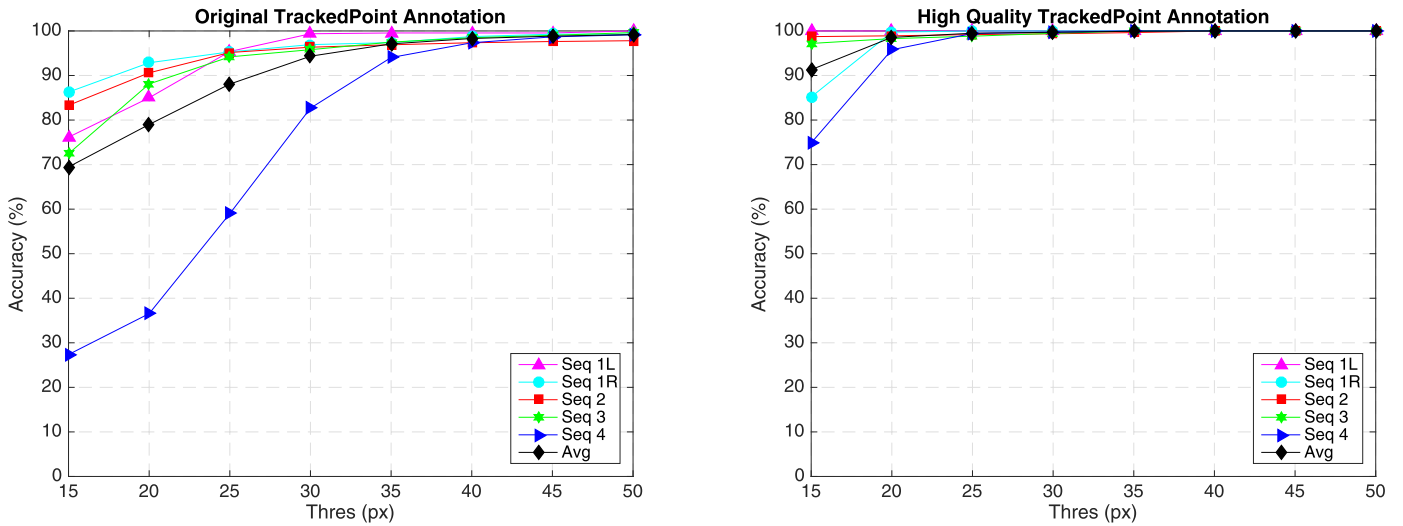


Fig. 14. Tracking accuracy of *EndoVis* Articulated Robotic Surgical Instrument training data under different accuracy threshold with the original and high-quality annotations.

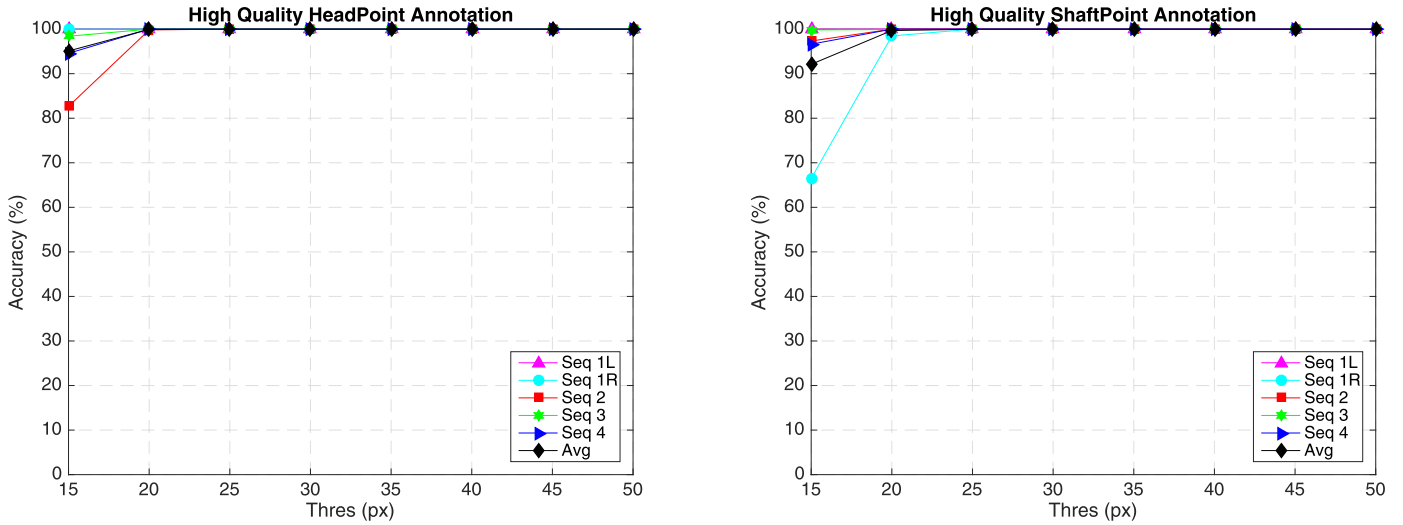


Fig. 15. Accuracy of *EndoVis* Articulated Robotic Surgical Instrument training data under different accuracy threshold with high quality annotation.

Table 6

Accuracy of *EndoVis* articulated robotic surgical instrument training data for the tracked point.

	Seq 1L	Seq 1R	Seq 2	Seq 3	Seq 4	Whole
Original annotation						
In-view (IV) and Out-of-view (OV) Frame Number						
IV	1107	1107	1096	1118	1056	5484
OV	0	0	29	6	67	102
Total	1107	1107	1125	1124	1123	5586
Accuracy (Thres = 20 px)						
Acc. (%)	85.00	92.86	90.60	88.10	36.55	79.01
Dist. (px)	7.42	7.07	7.41	9.64	9.26	8.00
Accuracy (Thres = 30 px)						
Acc. (%)	99.37	96.93	96.35	95.80	82.67	94.33
Dist. (px)	9.76	7.80	8.36	10.71	18.07	10.67
High quality annotation						
In-view (IV) and Out-of-view (OV) frame number						
IV	1107	1107	1099	1105	1066	5484
OV	0	0	26	19	57	102
Total	1107	1107	1125	1124	1123	5586
Accuracy (Thres = 20 px)						
Acc. (%)	100.0	99.73	98.91	98.28	95.78	98.56
Dist. (px)	4.89	9.87	3.29	4.31	11.13	6.65
Accuracy (Thres = 30 px)						
Acc. (%)	100.0	100.0	99.36	99.46	99.72	99.71
Dist. (px)	4.89	9.90	3.38	4.56	11.57	6.83

As we have summarized, the target is out of view several times in sequence 4, reaching 67 frames out of 1123 frames. Tracking-by-detection methods typically cannot handle out-of-view scenario without additional re-detection module. The underlying assumption is that the target is always in frame view, which means Whenever the target is out of frame, the tracker will gradually drift away. This explains the low accuracy of the performance, if the threshold is increased to 30 pixels, the performance has significantly improved, achieving 82.67% for accuracy.

We show some tracking result examples in Fig. 13. The tracked point and bounding box are shown in cyan colour, with the ground truth point shown in green colour. The first column is the first frame of each sequence. As we can see, the quality of the annotation is not consistent through the whole sequence. On certain frames, the annotation is drifted and is not labelled where it is supposed to be. This would certainly affect our performance evaluation result. It is also observed that whenever the instrument is close to the frame border, the tracker will stick to the border and not track the instrument well.

High quality annotation The original annotation is retrieved from the robotic system, which includes the location of the intersection point between the instrument axis and the border between plastic and metal on the shaft, normalized Shaft-to-Head axis

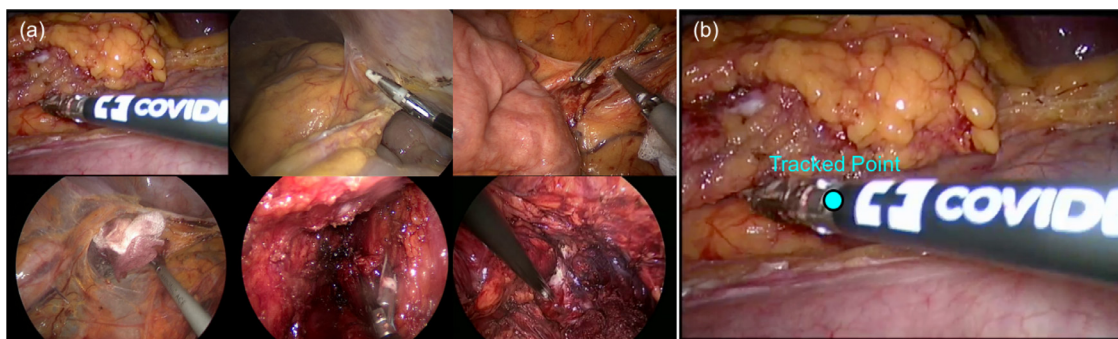


Fig. 16. (a) Example frame from each sequence of *EndoVis* articulated surgical instrument training dataset; (b) The annotation includes the position of the tracked point.

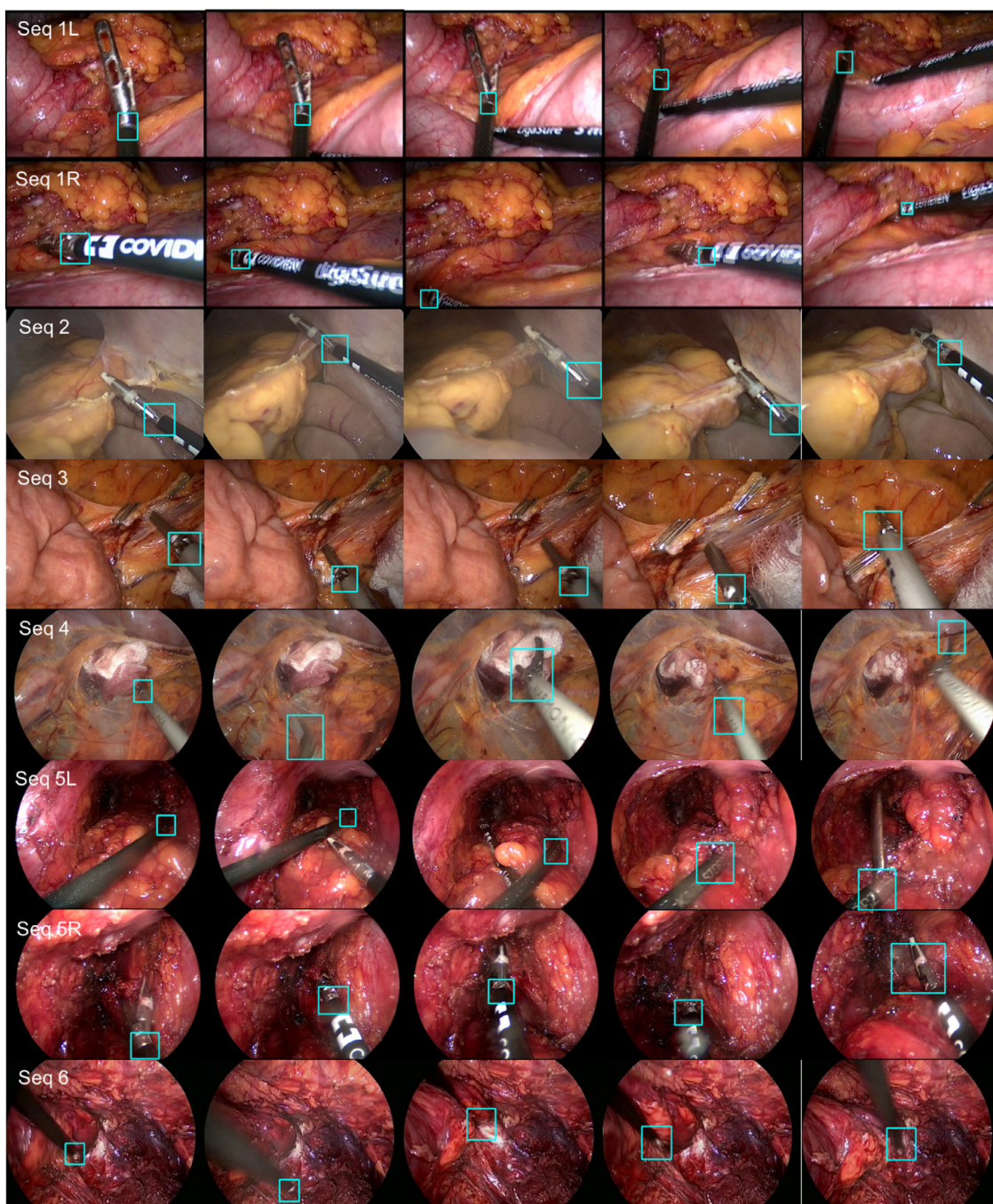


Fig. 17. Result example frames from each test sequence of the *EndoVis* conventional surgical instrument dataset. The result bounding box is represented in cyan colour.

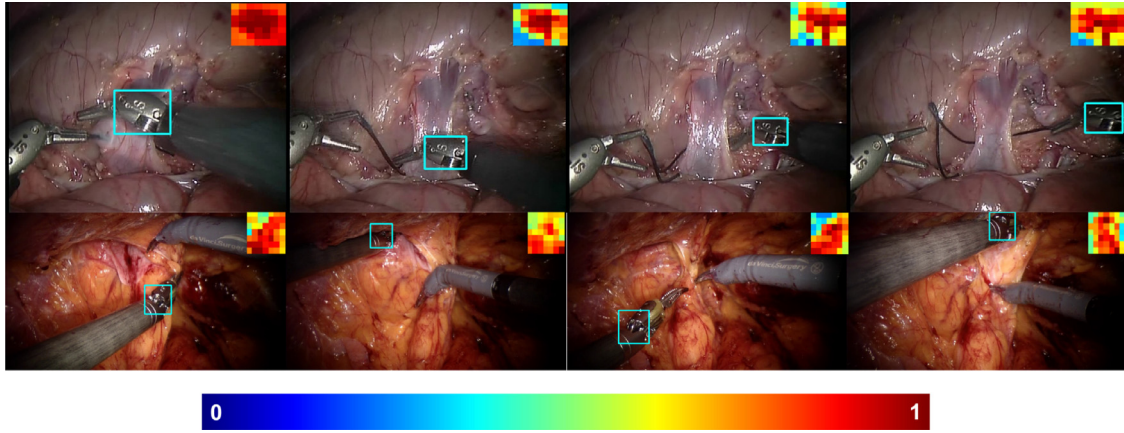


Fig. 18. Instrument Tracking result with patch weight displayed in the top corner of the image.

Table 7

Accuracy of *EndoVis* articulated robotic surgical instrument train data for Head and Shaft points with high quality annotation.

	Seq 1L	Seq 1R	Seq 2	Seq 3	Seq 4	Whole
In-view (IV) and Out-of-view (OV) frame number						
IV	1107	1107	1125	1124	1123	5586
OV	0	0	0	0	0	0
Total	1107	1107	1125	1124	1123	5586
Head accuracy ($Thres = 20$ px)						
Acc. (%)	100.0	100.0	99.82	100.0	100.0	99.96
Dist. (px)	3.06	4.10	10.32	4.52	6.33	5.68
Shaft accuracy ($Thres = 20$ px)						
Acc. (%)	100.0	98.46	100	99.91	100	99.68
Dist. (px)	2.48	12.08	6.82	4.79	6.48	6.51

Table 8

Distance (pixel) comparison with all the submitted methods for the tracked Point of the robotic laparoscopic instrument test set. Multiple instrument challenging subset is evaluated separately.

	$C_{multiple}$	Whole
KIT	113.91	106.60
UGA	40.73	34.94
MOD	45.12	40.16
PAWSS	38.36	29.66

vector and the clasper angle. Since the original annotation does not provide consistent ground truth, the accuracy result does not reflect true performance. We manually labelled the training data, and construct a high quality annotation. In this annotation, we labelled multiple joints of the instrument including the original tracked point, the Head and Shaft point. The original and our proposed annotations are demonstrated in Fig. 12 (b).

We also tracked and evaluated on the Head and Shaft points we defined in our high quality annotation in the high quality annotation section of Table 6 and Fig. 14 right. With new annotation, our average accuracy has increased to 98.56% for 20 pixel threshold, with distance error of 6.65 pixels.

The tracking accuracy evaluation results are displayed in Table 7 and Fig. 15. Our average accuracy has reached 99.96% and 99.68% for 20 pixels threshold, with distance error of 5.68 and 6.51 pixels, respectively.

Comparison performance In Table 8, the distance error (pixel) was computed and compared separately for challenging factor multiple instrument ($C_{multiple}$) with all the submitted methods KIT, UGA, MOD and our method PAWSS. From official report, PAWSS

Table 9

Distance (pixel) comparison with all the submitted methods for the tracked point of the conventional laparoscopic instrument test set. Various challenging subsets are evaluated separately.

	C_{blood}	$C_{multiple}$	$C_{objects}$	$C_{occlusion}$	C_{smoke}	Whole
KIT	233.62	220.87	117.23	225.58	193.85	178.89
UGA	276.44	235.42	228.04	193.82	231.87	217.91
PAWSS	181.59	110.85	68.29	87.11	96.31	96.78

outperforms all the other methods with the lowest average distance error 29.66 pixels.

EndoVis Conventional Laparoscopic Instrument Dataset The conventional instrument dataset contains six in vivo sequences, which are collected from complete laparoscopic colorectal interventions. Similar to the robotic instrument dataset, training data contains 45 s video sequences, and test data is made up of 15 additional seconds videos for each sequence and two new 60 s videos. Compared to *ex vivo* robotic instrument dataset, these sequences reflect complex challenges during surgery, including smoke, bleeding, blurry and various kinds of instruments. In Table 9, the distance error (pixel) was computed and compared separately for each challenging factor with all the submitted methods KIT, UGA and our method PAWSS. From the official report, PAWSS outperforms all the other methods in every challenging subset with the lowest average distance error 96.78 pixels. We show some tracking result examples in Fig. 17. The tracked point is shown in cyan colour, and the first column is the first frame of each sequence in test set. (Fig. 16)

In vivo surgical instrument experiments We also test on some other in vivo sequences and show the result in Fig. 18. As we can see, the tracker works well even under complex in vivo environment. The video is submitted to display the tracking results for the whole sequences.

5. Conclusions

In this paper, we propose a tracking-by-detection framework, called PAWSS, for online object tracking. It uses a colour-based segmentation model to suppress background information by assigning weights to the patch-wise descriptor. We incorporate scale estimation into the framework, allowing the tracker to handle both incremental and abrupt scale variations between frames. The learning component in our framework is based on Struck, but we would like to point out that theoretically our proposed method can also support other online learning techniques with effective background suppression and scale adaption.

The performance of our tracker is thoroughly evaluated on OTB, VOT2014 and VOT2015 datasets and compared with recent state-of-the-art trackers. Results demonstrate that PAWSS achieves the best performance in both PR and SR in OPE for OTB dataset. It outperforms Struck by 36.7% and 36.9% in PR/SR scores. Also, it provides a comparable PR score, and improves SR score by 4.8% over SOWP. On VOT2014 dataset, PAWSS has relatively lower accuracies but the lowest failure rate among the top trackers, we evaluated without re-initialization, and achieves the highest performance. Also on VOT2015 dataset, PAWSS is considered state-of-the-art and is among the top trackers.

For instrument tracking, we also qualitatively and quantitatively evaluated our tracker on public *EndoVis* robotic and conventional surgical instrument datasets, and in vivo surgical instrument sequences. We compared our result with the official GT for the Tracked Point on the robotic instrument dataset, and tracking accuracy reached 79.01% with 20 pixel threshold. As we have shown, the official annotation is not quality consistent, we manually created a high quality multi joint annotation for the dataset. We tested multiple joints (Tracked Point, Head and Shaft Point) on the dataset, and our performance accuracy increased over 98% for all the joints with 20 pixel threshold. From the official challenge report, Our method has shown the lowest tracking error for both robotic and conventional instrument datasets, and it also shown its excellent tracking ability with in vivo sequences dealing with complicated surgical environment. Our framework is designed for general single object tracking. It does not require prior information about the target or any offline training to achieve robust and real-time performance. We would also like to discuss the limitations of our framework. First, if the target disappears and reappears from the scene, the framework does not recover. Second, the target position is represented by rectangle bounding box. Even with the assistance of the segmentation model to distinguish foreground and background, the assumption is that the target occupies most area of the bounding box. If the target only occupies small fraction, the classifier would be polluted and misled by background information and can easily cause tracking failure. In the future, we would like to focus on re-detection module and semantic foreground segmentation.

Declarations of interest

None

Acknowledgements

Xiaofei Du is supported by the [China Scholarship Council \(CSC\)](#) scholarship. The work has been carried out as part of an internship at Wirewax Ltd, London, UK. The work was supported by the [EPSRC \(EP/N013220/1, EP/N022750/1, EP/N027078/1, NS/A000027/1, EP/P012841/1\)](#), The [Wellcome Trust \(WT101957, 201080/Z/16/Z\)](#) and the [EU-Horizon 2020 project EndoVespa \(H2020-ICT-2015-688592\)](#). This work was supported by the Wellcome/EPSRC Centre for Interventional and Surgical Sciences ([WEISS](#)) at UCL ([2031452/16/Z](#)) and [EPSRC \(EP/N027078/1, EP/P012841/1, EP/P027938/1, EP/R004080/1\)](#).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:<https://doi.org/10.1016/j.media.2019.07.002>.

References

Allan, M., Chang, P.-L., Ourselin, S., Hawkes, D.J., Sridhar, A., Kelly, J., Stoyanov, D., 2015. Image based surgical instrument pose estimation with multi-class la-

- belling and optical flow. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015 International Conference on. Springer, pp. 331–338.
- Allan, M., Ourselin, S., Hawkes, D.J., Kelly, J.D., Stoyanov, D., 2018. 3-D pose estimation of articulated instruments in robotic minimally invasive surgery. *IEEE Trans. Med. Imaging* 37 (5), 1204–1213.
- Allan, M., Ourselin, S., Thompson, S., Hawkes, D.J., Kelly, J., Stoyanov, D., 2013. Toward detection and localization of instruments in minimally invasive surgery. *Biomed. Eng. IEEE Trans.* 60 (4), 1050–1058.
- Allan, M., Thompson, S., Clarkson, M.J., Ourselin, S., Hawkes, D.J., Kelly, J., Stoyanov, D., 2014. 2d-3d pose tracking of rigid instruments in minimally invasive surgery. In: *Information Processing in Computer-assisted Interventions, International Conference on*. Springer, pp. 1–10.
- Avidan, S., 2007. Ensemble tracking. *Pattern Anal. Mach. Intell. IEEE Trans.* 29 (2), 261–271. doi:[10.1109/TPAMI.2007.35](https://doi.org/10.1109/TPAMI.2007.35).
- Bouguet, J.-Y., 2001. Pyramidal implementation of the affine Lucas Kanade feature tracker description of the algorithm. *Intel Corp.* 5 (1–10), 4.
- Cano, A.M., Gayá, F., Lamata, P., Sánchez-González, P., Gómez, E.J., 2008. Laparoscopic tool tracking method for augmented reality surgical applications. In: *International Symposium on Biomedical Simulation*. Springer, pp. 191–196.
- Chen, D., Yuan, Z., Wu, Y., Zhang, G., Zheng, N., 2013. Constructing adaptive complex cells for robust visual tracking. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1113–1120. doi:[10.1109/ICCV.2013.142](https://doi.org/10.1109/ICCV.2013.142).
- Collins, R.T., Liu, Y., Leordeanu, M., 2005. Online selection of discriminative tracking features. *Pattern Anal. Mach. Intell. IEEE Trans.* 27 (10), 1631–1643. doi:[10.1109/TPAMI.2005.205](https://doi.org/10.1109/TPAMI.2005.205).
- Comaniciu, D., Ramesh, V., Meer, P., 2003. Kernel-based object tracking. *Pattern Anal. Mach. Intell. IEEE Trans.* 25 (5), 564–577. doi:[10.1109/TPAMI.2003.1195991](https://doi.org/10.1109/TPAMI.2003.1195991).
- Danelljan, M., Häger, G., Khan, F., Felsberg, M., 2014. Accurate scale estimation for robust visual tracking. In: *British Machine Vision Conference*, Nottingham, September 1–5, 2014. BMVA Press doi:[10.5244/C.28.65](https://doi.org/10.5244/C.28.65).
- Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M., 2015. Convolutional features for correlation filter based visual tracking. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 58–66. doi:[10.1109/ICCVW.2015.84](https://doi.org/10.1109/ICCVW.2015.84).
- Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M., 2015. Learning spatially regularized correlation filters for visual tracking. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4310–4318.
- Duffner, S., Garcia, C., 2013. Pixeltrack: a fast adaptive algorithm for tracking non-rigid objects. In: *Computer Vision (ICCV)*, 2013 IEEE International Conference on. IEEE, pp. 2480–2487. doi:[10.1109/ICCV.2013.308](https://doi.org/10.1109/ICCV.2013.308).
- Gao, J., Ling, H., Hu, W., Xing, J., 2014. Transfer learning based visual tracking with gaussian processes regression. In: *Computer Vision-ECCV 2014*. Springer, pp. 188–203. doi:[10.1007/978-3-319-10578-9_13](https://doi.org/10.1007/978-3-319-10578-9_13).
- Godec, M., Roth, P.M., Bischof, H., 2013. Hough-based tracking of non-rigid objects. *Comput. Vision Image Underst.* 117 (10), 1245–1256. doi:[10.1109/ICCV.2011.6126228](https://doi.org/10.1109/ICCV.2011.6126228).
- Grabner, H., Grabner, M., Bischof, H., 2006. Real-time tracking via on-line boosting. In: *BMVC*, 1, p. 6.
- Hare, S., Saffari, A., Torr, P.H., 2011. Struck: structured output tracking with kernels. In: *Computer Vision (ICCV)*, 2011 IEEE International Conference on. IEEE, pp. 263–270. doi:[10.1109/TPAMI.2015.2509974](https://doi.org/10.1109/TPAMI.2015.2509974).
- He, S., Yang, Q., Lau, R., Wang, J., Yang, M.-H., 2013. Visual tracking via locality sensitive histograms. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2427–2434. doi:[10.1109/CVPR.2013.314](https://doi.org/10.1109/CVPR.2013.314).
- Henriques, J.F., Caseiro, R., Martins, P., Batista, J., 2015. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* 37 (3), 583–596. doi:[10.1109/TPAMI.2014.2345390](https://doi.org/10.1109/TPAMI.2014.2345390).
- Hua, Y., Alahari, K., Schmid, C., 2015. Online object tracking with proposal selection. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3092–3100.
- Kalal, Z., Mikolajczyk, K., Matas, J., 2012. Tracking-learning-detection. *Pattern Anal. Mach. Intell. IEEE Trans.* 34 (7), 1409–1422. doi:[10.1109/TPAMI.2011.239](https://doi.org/10.1109/TPAMI.2011.239).
- Kim, H.-U., Lee, D.-Y., Sim, J.-Y., Kim, C.-S., 2015. Somp: spatially ordered and weighted patch descriptor for visual tracking. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3011–3019. doi:[10.1109/ICCV.2015.345](https://doi.org/10.1109/ICCV.2015.345).
- Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Čehovin, L., Fernández, G., Vojir, T., Hager, G., Nebehay, G., 2015. The visual object tracking vot2015 challenge results. In: *Proceedings of the IEEE international conference on computer vision workshops*, pp. 1–23. doi:[10.1109/ICCVW.2015.79](https://doi.org/10.1109/ICCVW.2015.79).
- Kristan, M., Pflugfelder, R., Leonardis, A., Matas, J., Čehovin, L., Nebehay, G., Vojir, T., Fernández, G., Lukežič, A., 2015. The visual object tracking vot2014 challenge results. In: *Computer Vision - ECCV 2014 Workshops: Zurich, Switzerland, September 6–7 and 12, 2014, Proceedings, Part II*, pp. 191–217. doi:[10.1007/978-3-319-16181-5_14](https://doi.org/10.1007/978-3-319-16181-5_14).
- Lee, D.-Y., Sim, J.-Y., Kim, C.-S., 2014. Visual tracking using pertinent patch selection and masking. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3486–3493. doi:[10.1109/CVPR.2014.446](https://doi.org/10.1109/CVPR.2014.446).
- Li, X., Shen, C., Dick, A., Hengel, A., 2013. Learning compact binary codes for visual tracking. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2419–2426.
- Li, Y., Zhu, J., 2014. A scale adaptive kernel correlation filter tracker with feature integration. In: *Computer Vision-ECCV 2014 Workshops*. Springer, pp. 254–265. doi:[10.1007/978-3-319-16181-5_18](https://doi.org/10.1007/978-3-319-16181-5_18).
- Lukežič, A., Čehovin, L., Kristan, M., 2016. Deformable parts correlation filters for robust visual tracking. *arXiv preprint arXiv:1605.03720*.

- Nam, H., Han, B., 2015. Learning multi-domain convolutional neural networks for visual tracking. arXiv preprint arXiv:1510.07945.
- Pezementi, Z., Voros, S., Hager, G.D., 2009. Articulated object tracking by rendering consistent appearance parts. In: *Robotics and Automation (ICRA), 2009 IEEE International Conference on*. IEEE, pp. 3940–3947.
- Reiter, A., Allen, P.K., 2010. An online learning approach to in-vivo tracking using synergistic features. In: *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, pp. 3441–3446.
- Shi, J., Tomasi, C., 1994. Good features to track. In: *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pp. 593–600. doi:10.1109/CVPR.1994.323794.
- Tonet, O., Thoranaghatte, R.U., Megali, G., Dario, P., 2007. Tracking endoscopic instruments without a localizer: a shape-analysis-based approach. *Comput. Aided Surg.* 12 (1), 35–42.
- Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y., 2005. Large margin methods for structured and interdependent output variables. In: *Journal of Machine Learning Research*, pp. 1453–1484.
- Uecker, D.R., Wang, Y., Lee, C., Wang, Y., 1995. Automated instrument tracking in robotically assisted laparoscopic surgery. *J. Image Guided Surg.* 1 (6), 308–325.
- Wang, L., Ouyang, W., Wang, X., Lu, H., 2015. Visual tracking with fully convolutional networks. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3119–3127. doi:10.1109/ICCV.2015.357.
- Wang, N., Yeung, D.-Y., 2014. Ensemble-based tracking: aggregating crowdsourced structured time series data. In: *ICML*, pp. 1107–1115.
- Wei, G.-Q., Arbter, K., Hirzinger, G., 1997. Real-time visual servoing for laparoscopic surgery, controlling robot motion with color image segmentation. *IEEE Eng. Med. Biol. Mag.* 16 (1), 40–45.
- Wu, Y., Lim, J., Yang, M.-H., 2013. Online object tracking: a benchmark. In: *Computer vision and pattern recognition (CVPR), 2013 IEEE Conference on*. IEEE, pp. 2411–2418. doi:10.1109/CVPR.2013.312.
- Ye, M., Zhang, L., Giannarou, S., Yang, G.-Z., 2016. Real-time 3d tracking of articulated tools for robotic surgery. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2016 International Conference on*. Springer, pp. 386–394.
- Zhang, L., van der Maaten, L.J., 2014. Preserving structure in model-free tracking. *Pattern Anal. Mach. Intell. IEEE Trans.* 36 (4), 756–769. doi:10.1109/TPAMI.2013.221.
- Zhang, L., Ye, M., Chan, P.-L., Yang, G.-Z., 2017. Real-time surgical tool tracking and pose estimation using a hybrid cylindrical marker. *Int. J. Comput. Assist. Radiol. Surg.* 12 (6), 921–930.
- Zhang, M., Xing, J., Gao, J., Shi, X., Wang, Q., Hu, W., 2015. Joint scale-spatial correlation tracking with adaptive rotation estimation. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 32–40. doi:10.1109/ICCVW.2015.81.
- Zhang, X., Payandeh, S., 2002. Application of visual tracking for robot-assisted laparoscopic surgery. *J. Field Robot.* 19 (7), 315–328.
- Zhong, W., Lu, H., Yang, M.-H., 2012. Robust object tracking via sparsity-based collaborative model. In: *Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on*. IEEE, pp. 1838–1845. doi:10.1109/CVPR.2012.6247882.

Xiaofei Du received a bachelor degree in Telecommunications at Nanjing University of Posts and Telecommunications and a master in Biomedical Engineering from Tsinghua University, China. Currently, she is a Ph.D student at Centre for Medical Image Computing (CMIC) of University College London (UCL). Her research interests include surgical vision and medical image computing.

Maximilian Allan is a computer vision engineer at Intuitive Surgical in Sunnyvale, CA. He completed his PhD in surgical robot vision with Dan Stoyanov at University College London (UCL) in 2017 working on instrument detection and tracking for the da Vinci robot. He obtained a Master degree in computer science from Imperial College London in 2011 and a Bachelors in physics from Kings College London in 2010.

Sebastian Bodenstedt got a Ph.D degree in Karlsruhe Institute of Technology (KIT), he is currently a postdoc in National Center for Tumor Diseases Dresden, Germany. His research interests include medical robots and computer assisted interventions.

Lena Maier-Hein received the Ph.D degree from Karlsruhe Institute of Technology (KIT) with distinction and conducted her postdoctoral research in the Division of Medical and Biological Informatics at the German Cancer Research Center (DKFZ) and at the Hamlyn Centre for Robotics Surgery at Imperial College London. As an independent junior group leader at the DKFZ, she is now working in the field of computer-assisted medical interventions with a focus on multi-modal image processing, knowledge-based systems and computational biophotonics.

Stefanie Speidel is a Professor in National Center for Tumor Diseases Dresden, Germany. Her research interests include multimodal analysis of intraoperative data, biomechanical soft-tissue navigation as well as surgical data science.

Alessio Dore got a Ph.D degree in the Department of Biophysical and Electronic Engineering (DIBE), University of Genova, Italy. He is a senior data scientist and a manager at Deliveroo, London, UK.

Danail Stoyanov received a BEng degree from King's College London and a Ph.D degree in medical image computing from Imperial College London. He was a Royal Academy of Engineering Research Fellow when he joined the Centre for Medical Image Computing and the Department of Computer Science, University College London (UCL). He is now a Professor at UCL and an EPSRC Early Career Research Fellow. He is also Chief Scientist at Digital Surgery, London, UK. His research interests include computer assisted interventions, medical image computing and medical robotics.