Long Term Safety Area Tracking (LT-SAT) with Online Failure Detection and Recovery for Robotic Minimally Invasive Surgery

Veronica Penza[⋆◦], Xiaofei Du[†], Danail Stoyanov[†], Antonello Forgione[∗], Leonardo S. Mattos[◦] and Elena De Momi[⋆]

[⋆] *Department of Electronics Information and Bioengineering, Politecnico di Milano, P.zza L. Da Vinci, 32, 20133 Milano, Italy*

[◦] *Department of Advanced Robotics, Istituto Italiano di Tecnologia, via Morego, 30, 16163 Genova, Italy*

[†] *Centre for Medical Image Computing, Department of Computer Science, University College London, United Kingdom*

[∗] *Ospedale Niguarda Ca' Granda, P.zza Dell'Ospedale Maggiore, 3, 20162 Milano, Italy*

Abstract

Despite the benefits introduced by robotic systems in abdominal Minimally Invasive Surgery (MIS), major complications can still affect the outcome of the procedure, such as intra-operative bleeding. One of the causes is attributed to accidental damages to arteries or veins by the surgical tools, and some of the possible risk factors are related to the lack of sub-surface visibilty. Assistive tools guiding the surgical gestures to prevent these kind of injuries would represent a relevant step towards safer clinical procedures. However, it is still challenging to develop computer vision systems able to fulfill the main requirements: (i) long term robustness, (ii) adaptation to environment/object variation and (iii) real time processing.

The purpose of this paper is to develop computer vision algorithms to robustly track soft tissue areas (Safety Area, SA), defined intra-operatively by the surgeon based on the real-time endoscopic images, or registered from a pre-operative surgical plan. We propose a framework to combine an optical flow algorithm with a tracking-by-detection approach in order to be robust against failures caused by: (i) partial occlusion, (ii) total occlusion, (iii) SA out of the field of view, (iv) deformation, (v) illumination changes, (vi) abrupt camera motion, (vii), blur and (viii) smoke. A Bayesian inference-based approach is used to detect the failure of the tracker, based on online context information. A Model Update Strategy (MUpS) is also proposed to improve the SA re-detection after failures, taking into account the changes of appearance of the SA model due to contact with instruments or image noise. The performance of the algorithm was assessed on two datasets, representing *ex-vivo* organs and *in-vivo* surgical scenarios. Results show that the proposed framework, enhanced with MUpS, is capable of maintain high tracking performance for extended periods of time ($\simeq 4min$ - containing the aforementioned events) with high precision (0.7) and recall (0.8) values, and with a recovery time after a failure between 1 and 8 frames in the worst case.

Keywords: long-term tissue tracking, tracking failure detection, model update strategy, robotic minimally invasive surgery.

1. Introduction

The introduction of Robotics in Minimally Invasive Surgery (RMIS) allows overcoming many of the obstacles introduced by traditional laparoscopic techniques, by improving the surgeon's dexterity and the ergonomics during the surgical procedure, and restoring the surgeon's hand-eye coordination (Bravo et al., 2016; Forgione, 2009; Lanfranco et al., 2004). Despite these benefits, the outcome of the surgical procedure can still be compromised by adverse events occurring during the surgery. In robotic abdominal surgery, for example, one of the major complications is intra-operative bleeding due to injuries to vessels (Trinh et al., 2012; Kaouk et al., 2012; Sotelo et al., 2014). Main arteries or veins close to the surgical site can be accidentally damaged during the execution of a surgical procedure, being a major risk factor associated to the surgeon's skill or robotic system reliability (Lorenzo et al., 2011). Vessel damage may also activate a chain of secondary effects, such as the switch to open-surgery approach, a longer anaesthesia time and post-operative bleeding, thus negatively affecting the surgical performance and leading, in the worst case scenario, to patient death (Opitz et al., 2005).

Computer-assisted technologies coupled with robotic surgical systems can enhance the surgeon's capabilities and the control of the surgical tools by providing guidance to the surgical gestures. Specifically, these technologies could be used in abdominal robotic surgery to prevent vessel injury, by intra-operatively identifying and tracking a Region of Interest (ROI) bounding these delicate structures, which would work as active constraints to automatically prevent the robotic arms from touching this area. Intra-

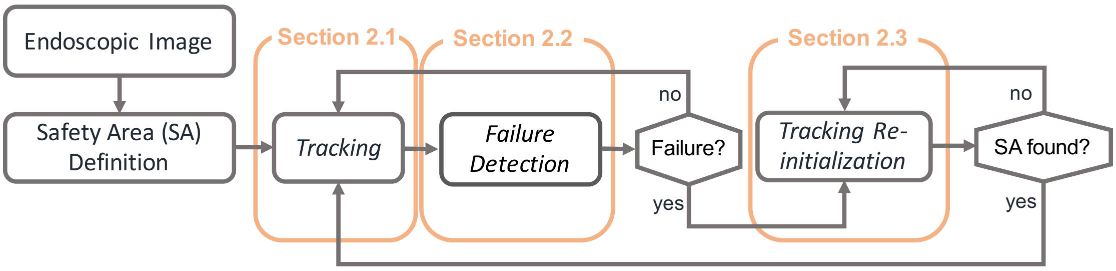


Figure 1: An overview of the proposed framework for long term tracking of a safety area defined on endoscopic images

operative identification of structures of interest has been explored using pre-operative information by means of Augmented Reality (AR) systems (Nicolau et al., 2011; Onda et al., 2014; Penza et al., 2014). However, this approach has to deal with dynamic changes of the anatomy between the data acquisition phase (pre-operative) and the surgical procedure (intra-operative) (Penza et al., 2016; Puerto-Souza et al., 2014; Faria et al., 2014). In fact, these changes can frequently occur due to (i) different pose of the patient with respect to the one in which the pre-operative information was stored, (ii) CO2 abdominal insufflation that presses and changes the shape of the organs, (iii) instrument tissue interaction, and (iv) heart beat and breathing that affect the registration on a smaller scale.

In order to measure the intra-operative tissue movements, computer vision and image processing algorithms have been exploited to track soft tissue areas relying only on the image characteristics (Stoyanov, 2012b). Early works on soft tissue tracking algorithms applied to endoscopic images have been done exploiting optical flow techniques. Stoyanov (2012a) used scene flow estimation techniques for the recovery of 3D structures and motion of the operating field from stereoscopic images, propagating this information to obtain a denser surface deformation identification. The main advantages of such methods are the sub-pixel accuracy and low execution time. However, for long-term endoscopic videos, the tissue area appearance may change or can be partially or totally occluded by instruments or camera movements. For these reasons, such algorithms typically accumulate errors resulting in tracking drift, or fail in case of occlusion.

Recently, different attempts have been implemented in order to build a long-term tracking system with enough robustness and reliability for long video sequences (in the order of minutes), which would be suitable for real surgical scenarios. This issue has been addressed using feature-based approaches, since they are invariant to rotation, scale changes of the area to track, and they are able to find feature matches between non consecutive frames, yet affecting accuracy and computational time. Yip et al. (2012) described a history preserving strategy to achieve long term tracking, without handling the effects of instrument occlusion and shading. A probabilistic framework to track affine-invariant anisotropic regions has been developed by Giannarou et al. (2013), where a recover strategy from potential tracking failure has been approached using spatial context and region similarity information to update an Extended Kalman Filter tracking framework. Puerto-Souza and Mariottini (2013) introduced a Hierarchical Multi-Affine (HMA) algorithm to map features between two endoscopic images, allowing to recover features that were lost after a complete occlusion or sudden camera motions. Mountney and Yang (2012) exploited online learning and classification using a context specific feature descriptor, in order to increase the robustness against drift and occlusion. Du et al. (2015) used a triangular geometric mesh model to combine features and intensity information to robustly track soft tissue surface deformation. Affine deformation modelling is used by Schoob et al. (2016) to provide motion compensation in dynamic surgical scenes, and an occlusion detection scheme was proposed to increase robustness against tracking failures. A framework for online tracking and retargeting is proposed by Ye et al. (2016), based on the concept of tracking-by-detection. Moreover, an important aspect to take into account in the development of a soft-tissue long-term tracker is the context information. In visual tracking, different works has been done in the areas of instrument segmentation from endoscopic video (Pezzementi et al., 2009; Bouget et al., 2015; Allan et al., 2015; Bouget et al., 2017).

Despite the progresses made, it is still challenging to develop a framework able to fulfil the main system requirements, as proposed by Yang et al. (2011):

- long-term robustness of the tracking even under complicated conditions recurring into the surgical field of view, such as: (i) tissue motion and deformation, (ii) occlusion by instruments, (iii) area out of field of view, (iv) large camera movements, (v) scale and orientation changes, (vi) blood and smoke changing the scene and (v) tissue specular highlights;
- adaptation to environment variations and changes in the tissue surface itself;
- real-time processing to allow the application in practical surgical scenarios (15 fps or greater value depending on the application).

In this work, we propose a framework for Long-Term Safety Area Tracking (LT-SAT) that is robust and reliable under the aforementioned adverse events in long surgical

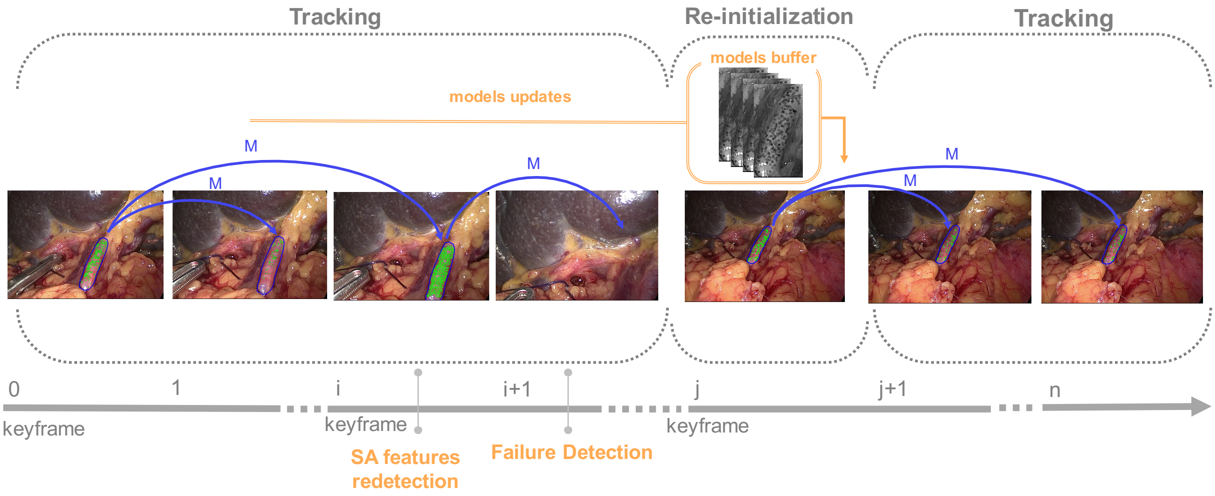


Figure 2: Graphical representation of the proposed framework for soft tissue Safety Area (SA) tracking. On the timeline are highlighted the main steps: Tracking, SA features redetection, failure detection and Tracking Re-initialization. The *keyframes* represent the reference frame, with respect to whom M is computed.

endoscopic sequences. In particular, considering the clinical issues previously described, we decided to focus the attention on tracking areas of interest to be preserved from injury during RMIS, such as main arteries or veins (portal vein, hepatic artery, splenic artery and vein, mesenteric artery and vein) in intervention of liver, pancreas, prostate and colon resection. However, the proposed framework can be applied to any other applications aiming at tracking structures of interest in the surgical field of view. The framework combines the advantages of an optical flow algorithm (Sec. 2.1) with a tracking-by-detection approach (Sec. 2.3), which exploits a novel Model Update Strategy (MUpS) for improving the identification of the Safety Area (SA). Since the optical flow methods are prone to failure, a Bayesian approach is used to detect possible failures, considering online context information (Sec. 2.2). An extensive quantitative analysis on ex-vivo and in-vivo video sequences is presented to demonstrate long-term achievement (Sec. 3). The results and discussion of this analysis are presented in Sec. 4 and conclusion in Sec. 5

2. Methods

The workflow of the proposed framework for long-term soft tissue tracking on endoscopic images is shown in Fig. 1. We assumed that the *Safety Area Definition*, i.e the identification of the structure to be preserved from injury during surgery on the endoscopic image, is done manually or registering a pre-operative model intra-operatively (Puerto-Souza et al., 2014). The SA is defined as a set of m 2D points $p_1, p_2, ..., p_m$ with each $p_i \in R^2$. The basic steps for the *Tracking* of the SA along the video sequence consist in (i) detecting salient features inside the SA (ii) finding corresponding features in the successive frames, and exploiting the matched features to (iii) find the perspective transformation between them (M) and used it to (iv) update the new position of the SA. Due to the presence of image noise, errors in the perspective transformation computation, or total occlusion of the SA, a tracking failure can occur. *Failure Detection* scheme is thus proposed, together with a *Tracking Re-initialisation* strategy to re-detect the SA in the image when visible. The *keyframe* represents the reference frame, that is re-initialised every time a *Tracking Re-initialisation* is performed. Fig. 2 shows more in detail the workflow of the proposed method, described in the following sections.

2.1. Tracking

The tracking of the SA is performed using a feature-based approach. In the first frame, a set of features ($\mathbf{f}_{GFTT}$) are detected inside the SA contour (SA_k), using GFTT detector (Shi and Tomasi, 1994). Kanade-Lucas-Tomasi Tracker (KLT) is then used for feature tracking since, as stated by Tomasi and Kanade (1991), it is fast and reliable in case of (i) small movements, (ii) constant brightness and (iii) constant flow in the local neighbourhood.

The feature tracking is computed estimating a frame-by-frame feature translation. Since this approximation can lead to errors in tracking due to (i) image noise, (ii) intensity changes caused by illumination or camera exposure changes, (iii) artefacts of the image sensor and (iv) specular reflections, the following strategies were implemented to remove outliers:

1. In order to check the matching correctness, an affine consistency check is also performed between the features belonging to the *keyframe* and the features in frame i, as stated by Shi and Tomasi (1994); The estimation of the affine motion between local window around the feature is considered as a measure of dissimilarity to reject wrong matches;

2. Endoscopic images are usually affected by specular reflections due to the tissue characteristics and the proximity of the light source to the tissue. The specular reflections appear as bright regions in the images and are identified applying a thresholding operation on S and V channels and dilatation operations (Lehmann and Palm, 2001). The features located close to specular highlights are discarded.

If $\mathbf{f}_{GFTT_k}$ is the set of features describing the SA_k in the $keyframe$ and $\mathbf{f}_{GFTT_i}$ is the corresponding set of tracked features in the frame i, the tracking of the SA is performed as follows:

$$SA_i = M \cdot SA_k \tag{1}$$

where SA is the SA contour for the *keyframe* and for the frame i and M is the perspective transformation computed between $\mathbf{f}_{GFTT_k}$ and $\mathbf{f}_{GFTT_i}$. M is computed and applied with respect to the $keyframe$ and not with respect to the previous frame $i-1$ in order to avoid drifting and accumulating errors during tracking, as it is shown in Fig. 2. The perspective transform M is computed using the RANSAC strategy (Fischler and Bolles, 1981), which is robust in populations with an high number of outliers.
Using these strategies in long video sequences, the number of matched features decreases in time, compromising the reliability of M and thus, the tracking. In the proposed workflow, the re-detection of the features is performed each time the features number decreases below the 70% with respect to the features detected in the frame *keyframe*. The frame in which a re-detection is computed is considered as the new *keyframe*, i.e. the successive M transformations will be computed with respect to the set of features detected in this frame.

2.1.1. Foreground-Background Segmentation

A global Bayesian probabilistic model based on colour histogram is implemented in order to constantly discriminate features describing the SA from the ones describing background or any other object occluding it, inspired by the work of Duffner and Garcia (2013) and Du et al. (2016). A Probability Segmentation Map (PSM) is computed, representing for each pixel of the SA the probability of belonging to the background $p(c = 0)$ or to the foreground $p(c = 1)$. This map is used to keep only the features laying in a pixel with a foreground probability $p(c_i = 1|y_{1:i}) > \tau_{foreground}$. We preferred a pixel-based segmentation in order to maintain an accurate pose estimation of the object to track. A cost aggregation step, which could make this process less sensitive to image noise, was not taken into account to keep the computational time low.

The initialisation of the probabilistic model is done by computing the HSV histogram (with $12x12$ bins for H and S channels and 8 separate bins for V channel) of the rectangular area fitting the SA, to identify the colour characteristics of the foreground/background. Assuming that the area of interest is completely visible in the SA, the foreground histogram is initialised considering the area of the image inside the convex hull of the features detected, as proposed by Du et al. (2016), while the background is initialised using the pixel values outside the convex hull, as shown in Fig. 3b.

In the successive frames, in order to deal with appearance changes of the tissue, the HSV colour distribution of the pixels, previously identified as background/foreground, is used to update the current probability distribution, as stated in Eq. 2. Consequently, the probability of each pixel to belong to the background or foreground, is determined by its HSV colour, the previously $(i-1)$ computed probability distribution, and on transition probabilities for foreground and background $p(c_i|c_{i-1})$. The transition probabilities were chosen in order to disadvantage the transition from background to foreground. In fact, in the case of a SA occlusion by an instrument for a long time, the pixels belonging to the instrument have to remain part of the background to ensure that the instrument is not becoming part of the tracked model. On the other hand, pixels belonging to the tissue have to take into account appearance changes and thus have a higher translational probability.

The PSM update can be described by the following Eq:

$$\begin{aligned} p(c_i|y_{1:i}) = \frac{p(y_i|c_i = 1)}{z} \\ \sum_{c_{i-1}} p(c_i = 1|c_{i-1})p(c_{i-1}|y_{1:i-1}) \end{aligned} \tag{2}$$

where c_i is the class of the pixel at frame i (where $c \in \{0, 1\}$), $y_{1:i}$ is the pixel's HSV colour from frame 1 to i, and z is a normalisation constant to keep the probabilities sum to 1.

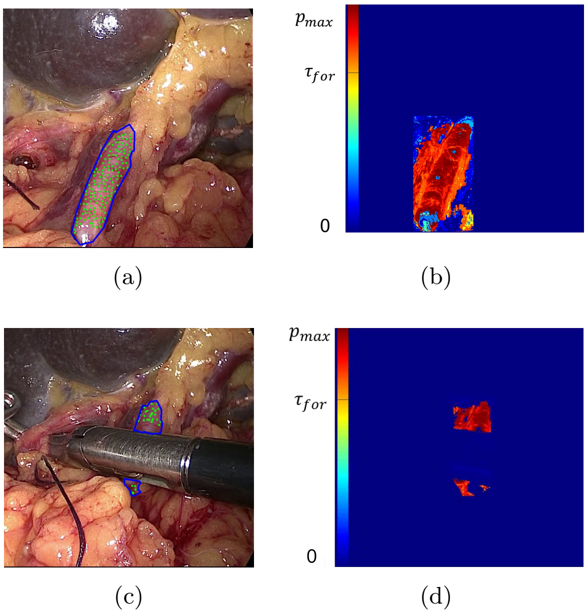
For not compromising the update of the model in case of partial occlusion of the SA, a clustering of the features inside the SA is computed using the `kmeans` OpenCV function, and only the pixels belonging to the convex hulls of the clustered features ($n_{cluster}$) are considered during the update of the foreground histogram (see Fig. 3c). An example of how the features are discarded depending on the computed PSM is illustrated in Fig. 3.

2.2. Failure Detection

A Bayesian approach is used to estimate the joint failure probability of the tracker, defined as $p(F|A, B, C, D)$, and caused by a combination of the multi clues A, B, C and D.

A is the number of tracked features (n_{feat}) inside the SA, necessary for the computation of M. If n_{feat} is less than 4, M cannot be computed;

B is the percentage of features lost in frame i with respect to the number of features in the *keyframe* (p_{lost}). A high percentage of lost features could indicate the presence of a partial occlusion or sudden changes in the scene;



(a) (b)

(c) (d)

Figure 3: On the top, the SA definition (left) and the correspondent Probability Segmentation Map (right) are shown. In the left image, it is also possible to see, drawn with a green line, the convex hull defined on the entire set of features and used for the initialisation of the foreground probability. On the bottom left, a frame with partial occlusion is shown. Here again, the green lines represent the convex hulls defined on the feature clusters. The correspondent PSM is shown on the right, where it is possible to see how the instrument occluding the SA has a low probability of belonging to the foreground (in blue). The values of the colour bar of the PSMs ranges from 0 to the maximum probability value of belonging to foreground of the frame (normalised with respect to the maximum value for providing a better visualisation). τ_{for} is the foreground threshold, beyond which the pixels are considered part of the foreground.

C is the validity of the perspective transform M computed between $\mathbf{f}_{GFTT_k}$ and $\mathbf{f}_{GFTT_i}$ (v_M), considered valid if: (i) the z coordinate of the transformed points is positive, and (ii) the ninth element of the homography transformation is non-zero, which means a non-valid perspective matrix;

D is the standard deviation of the optical flow distribution (std_{of}) in terms of image velocity directions. A wide distribution indicates errors in the matching stage due to a sudden change of the scene.

The conditional probability table was defined by assigning to each of the clues a probability distribution, as shown in Fig. 4. For the clues A, B and D, that are continuos variables, a Gaussian distribution was chosen, and for the clue C a probability cross table was used, since v_M can assume only two values (0 or 1). If P(F| A,B,C,D) > p_{th}, the framework switch to the *Tracking Re-initialisation* (See Sec. 2.3).

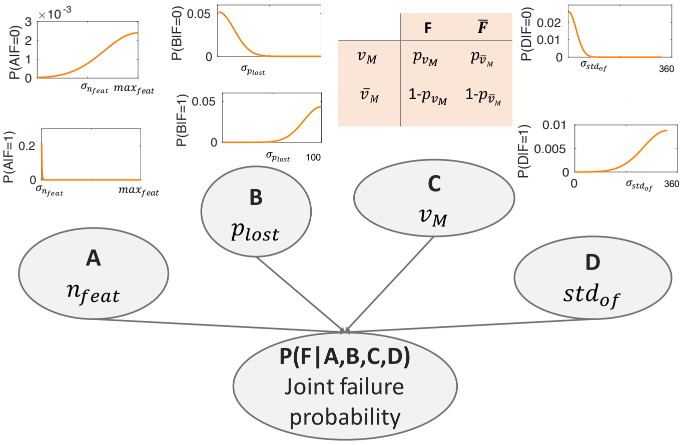


Figure 4: Graphical representation of the joint probability used to estimate the tracking failure (F). The probability distributions associated to the clues (A, B, C, D) are shown.

The parameters of the probability distributions (made of the sigma values for each distribution and the probability values for C) were chosen in order to reach a high accuracy on the detection of the tracking failure. In order to find the parameters giving the maximum accuracy, a Monte Carlo sampling approach was used. The parameters were sampled within pre-defined ranges, selected from experimental observations, and the cost function, used to determine the best parameter set, was defined as the weighted accuracy of the failure classification, as described by the following Eq.:

R 8

$$accuracy = \frac{k_1 \cdot TP + k_2 \cdot TN}{k_1 \cdot P + k_2 \cdot N} \tag{3}$$

where TP, TN, FP and FN are respectively the number of true positive, true negative, false positive and false negative classifications, and $k_1 = 0.8$, $k_2 = 0.2$ are the weights assigned to favour an high rate of TP.

The joint probability was iteratively computed, varying the randomly sampled parameters, on a subset of video sequences (training set), where the evidence values (n_{feat}, p_{lost}, v_M, std_{of}) were used as input, together with the ground truth information (manually defined when the failure of the tracking really occurred). The parameter set giving an accuracy higher than 0.9 was chosen after 1000 iterations.

2.3. Tracking Re-initialization

If a failure during the tracking is detected, a tracking-by-detection approach based on the generalized Hough transform (Ballard, 1981) is used to find the SA model in the current frame i, inspired by Seib et al. (2012). The re-detection of the SA is performed in three phases, as described in the following subsections.

2.3.1. Model Initialization

In the first *keyframe*, in which the SA is defined, a model of the SA is stored. SURF features ($\mathbf{f}_{SURF_k}$) and descriptors (Bay et al., 2006) are computed inside the SA, since its scale and rotation invariant characteristics are necessary to match features between non-consecutive frames, as in the case of Tracking Re-initialization. KLT initialized with

GFTT would not be useful in this case, since it searches feature matches locally, without taking into account possible large displacements of the SA and being invariant to rotation and scale. The model is characterized by the feature position (x, y), scale (σ) and orientation (θ), and the centroid (c_0) of the area. These feature descriptors, considered with respect to the centroid, uniquely characterise the SA, enabling the SA recognition at any frame.

2.3.2. Model Update

The model defined in the first frame is not always enough to re-detect the SA in long video sequences, since changes in the tissue appearance may occur. For this reason, we used a fixed number of models (n_{models}) chosen following a novel Model Update Strategy (MUpS). These models (m_j, where $j = 1...n_{models}$ is the index of the model) are stored in a buffer. They should be different enough from the first model to represent small variations. However, in order to avoid the collection of erroneous models, a similarity with the first model should be ensured. As a measure of similarity we choose the Bhattacharyya distance (BD) between the color histogram of the model in the *keyframe* and in the current frame i, inspired by (Giannarou et al., 2013). It is defined as:

$$BD(H^{first}, H^{curr}) = \sqrt{1 - \rho(H^{first} H^{curr})} \tag{4}$$

where H is the normalized histogram density defined as $H = \{h_{bin}\}_{bin=1...m}$, with $\sum_{bin=1}^{m} h_{bin} = 1$.
ρ is the Bhattacharyya coefficient computed from the following Eq.:

$$\rho(H^{first}, H^{curr}) = \sum_{bin=1}^{m} \sqrt{h_{bin}^{first} h_{bin}^{curr}} \tag{5}$$

In order to make the similarity measure robust against illumination variations, we opted for using the combination of H and S channels from the HSV image instead of the RGB channels used by Giannarou et al. (2013). Thus, the strategy to update the model, also described in Alg. 1, can be explained as follows:

- The model stored in the first frame is always kept fixed in order to always have a valid reference;
- The new model (m_i) is collected only if the BD is within the range $\delta_{BD} = \{\delta_{min}, \delta_{max}\}$. Indeed, a value of $BD < \delta_{min}$ means that the current model is too similar to the first one and storing it in the buffer would not add any meaningful information, while $BD > \delta_{max}$ means that the model is too different from the first one and it should not be stored in the buffer to avoid collecting erroneous models. To find out the best values for δ_{min} and δ_{max}, we computed the histogram of BD values of a training dataset, and we observed that the 10$^{\text{th}}$ and 90$^{\text{th}}$ percentile were best representatives for strongly similar and strongly dissimilar models, respectively;

Algorithm 1 Model Update Strategy

```
procedure UPDATEWEIGHT(model, $\beta_1$, $\beta_2$)
    $model.weight = \beta_1 \cdot model.BD + \beta_2 \frac{model.nT_{used}}{model.nT_{\overline{used}}}$
end procedure
procedure UPDATEMODEL(modelBuffer, modelNew, BDmax, BDmin, m, $\beta_1$, $\beta_2$)
    if modelNew.BD > BDmin and modelNew.BD < BDmax then
        if sizeof(modelBuffer) < $n_{models}$ then
            add modelNew to modelBuffer
            UPDATEWEIGHT(modelNew, $\beta_1$, $\beta_2$)
        else
            modelWeakest = model with lowest weight in modelBuffer
            if $modelWeakest.nT_{\overline{used}} > m$ then
                replace modelWeakest with modelNew in modelBuffer
                UPDATEWEIGHT(modelNew, $\beta_1$, $\beta_2$)
            end if
        end if
    end if
    return modelBuffer
end procedure
```

- A weight is assigned to each model belonging to the buffer, as:

$$w_j = \beta_1 \cdot BD_j + \beta_2 \frac{nT_{used}}{nT_{\overline{used}}} \tag{6}$$

  where nT_{used} is the number of times the model was previously used for the SA recognition and the number of times the model was not used is $nT_{\overline{used}}$. β_1 and β_2 are the weights assigned to the two parameters determining the goodness of the model (w_j);
- If the buffer is not full, the new model m_i is added to it;
- If the buffer is full, the model m_j with the minimum weight w_j will be replaced by the new model m_i only if:

$$nT_{\overline{used}} > t \tag{7}$$

  where t was empirically chosen.

2.3.3. Model Recognition

SURF features are detected on the entire frame i and then are matched with the features belonging to the set of m_i using a nearest-neighbor matching. As a first outlier rejection stage, the wrong matches are rejected if the ratio between the closest and the second-closest descriptor distances is lower than a threshold τ_l (Lowe, 2004). The possible SA poses, represented by the position (x, y), scale (σ) and orientation (θ), are clustered in a multi-dimensional Hough-space accumulator, as shown in Fig. 5. The coarse grid represents translation in x and y direction, while in each of these cells, the bins along x axis represent σ and

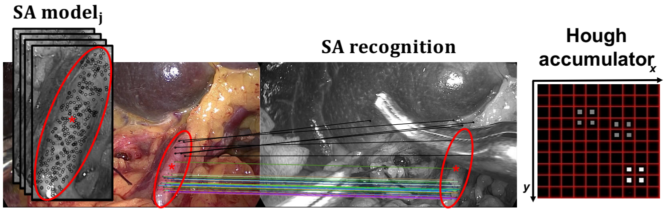


Figure 5: On the left, the process for the recognition of the SA in the new frame. The features detected in the current frame are matched with the ones belonging to the models (colored and black lines indicate respectively right and wrong matches). On the right, the Hough accumulator is shown: the two axes indicate the feature position, and, inside each cell, the horizontal and vertical translation encode the scale and rotation, respectively. Each feature match votes for a possible SA position, increasing the Hough space accumulator, represented on the right. Right matches increment the same Hough accumulator cells, leading to a maximum (white squares), while the wrong matches votes are scattered (gray squares).

the ones along the y axis represent θ. Each feature match independently votes for a possible SA position, orientation and scale, increasing the corresponding bin in the accumulator. The new centroid position c_i is estimated as:

$$\mathbf{c_i} = (c_{xi}, c_{yi}) = \mathbf{v} + \mathbf{p_i} \tag{8}$$

where p_i is the feature position in frame i and $\mathbf{v}$:

$$\mathbf{v} = \begin{pmatrix} cos(\alpha) & -sin(\alpha)) \\ sin(\alpha) & cos(\alpha) \end{pmatrix} (\mathbf{c_0} - \mathbf{p_0}) \frac{\sigma_s}{\sigma_0} \tag{9}$$

$\mathbf{v}$ is the translation vector from the centroid of the model $\mathbf{c_0}$ to the position of a feature $\mathbf{p_0}$ in the model, normalized with the scale ratio of the feature in frame i (σ_i) and of the feature of the model (σ_0), and rotated depending on $\alpha = |\theta_0 - \theta_i|$, i.e. the rotation angle between respectively the feature rotation of the model and of the frame i. The maximum in the Hough-space returns the set of features $\mathbf{f}_{SURF_i}$ of the model that best match the SA, a shown in Fig. 5.

Every time a SA is recognised, the Tracking algorithm is re-initalized with the same workflow described in Sec. 2.1, establishing a new *keyframe*. In this phase, the probability segmentation map has a fundamental role, since the SA can still be partially occluded. Keeping only the features belonging to the foreground prevents from failure. If the SA is not recognised, the algorithm waits until it is visible again.

3. Experimental Evaluation

The evaluation is focused at demonstrating the robustness of the algorithm against: (i) partial occlusion (PO), (ii) total occlusion (TO), (iii) SA out of the field of view (OFV), (iv) deformation (DEF), (v) illumination changes (IC), (vi) abrupt camera motion (ACM), (vii), blur (BLR), and (viii) smoke (SMK), which are the main events, often happening during surgeries, that can affect the reliability of a tracker. In order to assess the performance of the algorithm against these events, we used *ex-vivo* and *in-vivo* datasets.

Parameters	definition	values
	Tracking	
$n_{cluster}$	number of feature clusters	8
$p(c_i = 0) \| c_{i-1} = 0)$	background to background transitional probability	0.9
$p(c_i = 0) \| c_{i-1} = 1)$	background to foreground transitional probability	0.1
$p(c_i = 1) \| c_{i-1} = 0)$	foreground to background transitional probability	0.6
$p(c_i = 1) \| c_{i-1} = 1)$	foreground to foreground transitional probability	0.4
$\tau_{foreground}$	foreground threshold	0.7
	Failure Detection	
max_{feat}	maximum feature number for distribution A	$n_{feat_{SA}} \cdot 2$
$\sigma_{n_{feat}}(F = 0)^*$	sigma value of the non-failure probability distribution of A	$\frac{max_{feat}}{3}$
$\sigma_{n_{feat}}(F = 1)^*$	sigma value of the failure probability distribution of A	6
$\sigma_{p_{lost}}(F = 0)^*$	sigma value of the non-failure probability distribution of B	19
$\sigma_{p_{lost}}(F = 1)^*$	sigma value of the failure probability distribution of B	53
$p_{\bar{v}_M}(F = 0)^*$	non-failure probability for C	0.1
$p_{v_M}(F = 1)^*$	failure probability for C	0.47
$\sigma_{std_{of}}(F = 0)^*$	sigma value of the non-failure probability distribution of D	43
$\sigma_{std_{of}}(F = 1)^*$	sigma value of the failure probability distribution of D	165
p_{th}	failure threshold	0.4
	Tracking Re-initialization	
n_{models}	number of models used by the MUpS	10
t	minimum number of times a model can be used before being replaced	5
δ_{min}	10th percentile of BD distribution	0.1
δ_{max}	90th percentile of BD distribution	0.5
β_1	model weight parameter	0.5
β_2	model weight parameter	0.5

Table 1: Summary of the parameters used in the evaluation of the algorithm. All the parameter values were empirically chosen except for the ones marked with *, whose values were computed as described in Sec. 2.2.

R 4-
R 5

The *ex-vivo* dataset is made of endoscopic images of ex-vivo organs (goat kidney, pig liver). It was developed simulating surgical scenarios in a controlled way, recreating typical events happening during surgery. The videos were recorded using a da Vinci® stereo camera and the robotic system (Intuitive Surgical Inc., CA) at the Surgical Robot Vision group (University College London, London, UK). All the videos were recorded at 25fps with an image resolution of 720 × 576.

The *in-vivo* dataset consist of videos of real surgical oper-

ations performed at Ospedale Niguarda Ca' Granda (Milan, Italy). The videos were captured with a monocular STORZ endoscope, at $25fps$ and a resolution of 1280×720. All the data were appropriately anonymised. Details of each video sequences in terms of duration, number of frames, and a brief description are presented in Fig. 6
For each sequence, we created a Ground Truth (GT), in the form of a 2D polygon around the area of interest, with a interframe step of 10. This was performed manually by an operator with the supervision of an expert surgeon. For a more accurate evaluation, the same frames were also labeled with one of the following attributes: SA visible (SAV), PO, TO, OFV, DEF, IC, ACM, BLR and SMK. These datasets and the associated ground truth are available online for the benefit of the community [1]. Tab. 2 shows the percentage of frames with each attribute for each video of the two datasets.

The performance was assessed using precision and recall curves, and the F-measure (Wu et al., 2013). For each video, the precision value α was computed as:

$$\alpha = \frac{TP}{TP + FP}$$

where TP is the number of true positives of the SA tracked and FP is the number of false positives of the SA tracked. The recall value β is defined as:

$$\beta = \frac{TP}{TP + FN}$$

where FN is the number of false negatives of the SA tracked. The F-measure γ is the harmonic mean of precision and recall:

$$\gamma = 2 \cdot \frac{\alpha \cdot \beta}{\alpha + \beta}$$

The metrics used to for the definition of true positives of the SA is the overlap ratio, measured in pixels, and defined as:

$$\o = \frac{|T \cap G|}{|T \cup G|}$$

where T is the set of SA tracking results, and G is the set of GT.

The *precision and recall curves* were computed varying the overlap ratio threshold used to identify the TP values. The F-measure was computed considering $\o > (0.2, 0.5, 0.8)$.

In case of partial occlusion, the SA also included the occluding object. So, to take into consideration only the area of interest (foreground), the PSM was used to keep only the pixels inside the SA with $p > \tau_{foreground}$, which effectively discarded all pixels belonging to the background (and thus the occluding object). The overlap ratio was computed considering only the foreground area. The evaluation was performed with and without the MUpS, in order to assess its contribution. Precision and recall curves where computed for both cases. A detailed analysis of the robustness of the LT-SAT tracker against the events (SAV, PO, DEF, IC, ACM, BLR, SMK) was performed computing precision and recall curves, considering all the video sequences together. Results are represented using the Area Under the Curve (AUC).

The *recovery time* after the failure was computed as the mean number of frames between the lost of the SA tracking and the correct re-detection ($\o > 0.5$) for each video sequence, with and without MUpS.

In these experiments, the accuracy of the segmentation is implicitly evaluated with precision curves. In fact, the PSM was used to determine the foreground area inside the SA, and this area is used to compute the overlap ratio used in precision and recall evaluation.

The code was implement in C++, using the OpenCV library for the management of the images and KLT library[2] for KLT algorithm implementation, since the OpenCV version of the KLT tracker does not include the affine consistency check. The code released by Seib et al. (2012) was used for the tracking by detection approach. The program was running on a system with GNU/Linux operating system, and a CPU Intel Core i5-3230M with four cores. The parameters used for this evaluation are summarized in Tab.1. The same parameter values were used for all the video sequences.

R 5

4. Results and Discussion

In Fig. 6 example images from each video sequences of the *in-vivo* and *ex-vivo* dataset are shown. In the first column, the SA, as defined in the first frame, is shown. The second and third column show two characteristic frames showing one of the aforementioned events. It is worthy to point out that, in the analysed videos, the SAs represent different tissue surfaces, and there is a high percentage of frames with partial or total occlusion, where the target is not visible, as shown in Tab.2, allowing the assessment of the algorithm under different conditions.
Fig. 7 shows an example of the trend of the variables representing the clues (A, B, C, D) used to estimate the joint failure probability P. The second last row of the figure represents when the Tracking Re-initialisation is active. From these data, we can observe different events (highlighted in Fig. 7 with numbered orange boxes) that trigger the tracking re-initialisation:

1. The percentage of lost features drastically increases since the area is moving out of the camera field of view;
2. The number of features inside the SA decreases drastically due to an instrument occlusion. This event combined with an increase of the optical flow standard distribution (caused by wrong matches) leads to a failure of the KLT tracker;

[1] http://nearlab.polimi.it/medical/dataset/

[2] https://cecas.clemson.edu/stb/klt/

	ex-vivo dataset			*in-vivo* dataset					
	EV1	EV2	EV3	IV1	IV2	IV3	IV4	IV5	IV6
SAV [%]	58.14	52.55	66.55	28.50	54.62	20.83	31.25	30.39	41.31
PO [%]	7.43	7.45	12.55	40.17	27.18	6.94	37.50	13.24	12.46
TO [%]	7.86	11.45	14.18	4.50	0.00	4.17	0.00	0.00	0.00
OFV [%]	12.57	10.36	3.82	23.50	0.53	1.39	0.00	0.00	2.62
DEF [%]	3.71	8.00	0.00	1.00	1.58	0.00	0.00	2.94	8.52
IC [%]	0.00	0.00	0.00	0.00	1.85	13.89	0.00	14.71	2.62
ACM [%]	5.57	3.64	0.00	0.67	1.85	5.56	0.00	7.84	4.92
BLR [%]	4.71	6.55	2.91	1.67	5.80	47.22	31.25	15.20	24.59
SMK [%]	0.00	0.00	0.00	0.00	6.60	0.00	0.00	1.72	2.95

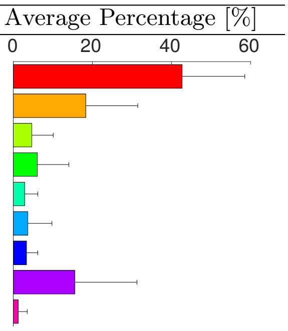


Table 2: Percentage of frames with Safety Area Visible (SAV), Partial Occlusion (PO), Total Occlusion (TO) , Out Of Field of View (OFV), Deformation (DEF), Illumination Change (IC), Abrupt Camera Motion (ACM), Blur (BLR) and Smoke (SMK) for each video of the two datasets.

3. The standard distribution of the optical flow increases due to a sudden movement of the camera; The gradient of colour represents the direction of the arrows and shows a scattered behaviour around the SA, representing erroneous features matching. KLT tracker alone would fail since it is not robust to sudden movement of the scene;
4. The homography is invalid due to wrong feature matches caused by partial instrument occlusion. In this case, KLT would continue to work, however tracking wrong features (i.e. not belonging to the area of interest) and compromising the SA tracking.

The Bayesian joint probability computation was adjusted to be very sensible to possible failures, because a false negative is not critical for our application. All the events aforementioned are examples of cases in which KLT fails, demonstrating the need of a failure detection and SA re-detection strategies. Moreover, if a failure is not detected at the first frame, the tracking shows a degradation and the Bayesian joint probability will estimate the failure most probably after a few frames, re-initialising correctly the SA.

The precision and recall curves are shown in Fig. 8. These curves demonstrate the performance of the tracking and the effect of the MUpS. Considering all frames, the precision and recall values are strong, as it is confirmed by the F-measure (Tab. 3). Fig. 9 shows the AUC of the precision and recall curves for all the different events, considering all the video datasets. Following this figure, the effect of the MUpS improves the recall values in all the cases, to the detriment of slightly lower precision values. The bar plot highlights the robustness of the LT-SAT tracker with MUpS against all the events, considering challenging *in vivo* video sequences.

Tab. 3 reports the *recovery time* used to re-detect the safety area after a failure. As we can observe, the MUpS improves significantly the recovery time.

The computational time of the framework does not fulfil the requirement needed for real time application, since it reached only $\simeq 1.60 fps$. Nevertheless, since the current implementation of the code was more focused on the development and testing of the algorithm performance, the computational performance can be optimised by improving the software architecture and memory management.

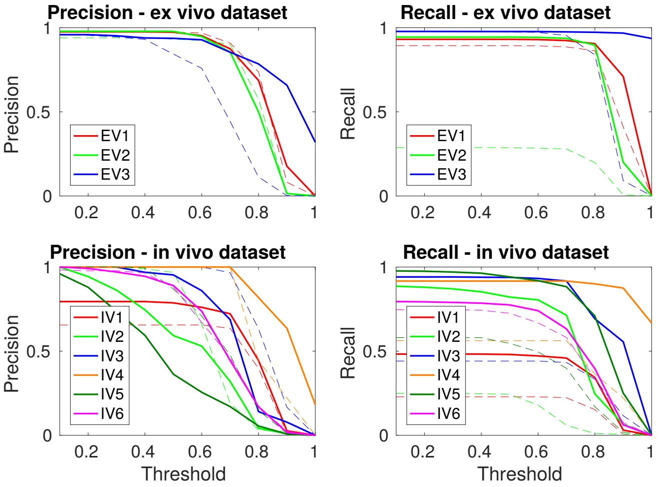


Figure 8: Precision and Recall curves for all the video sequences of the *ex-vivo* (first row) and *in-vivo* (second row) datasets. The dashed and continuous lines represent the results without and with the Model Update Strategy (MUpS), respectively.

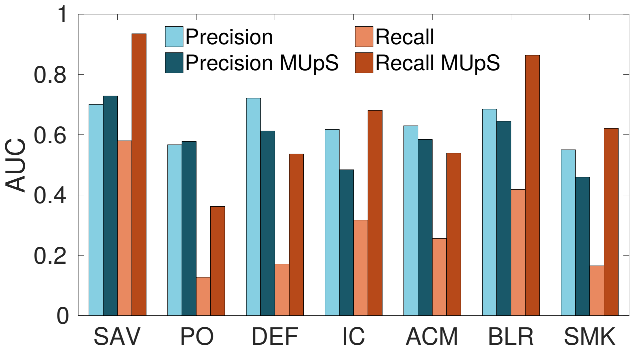


Figure 9: Bar plot of the Area Under the Curve (AUC) for Precision and Recall curves considering all the video sequences of the *ex-vivo* and *in-vivo* datasets.

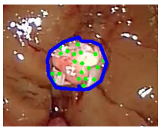
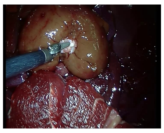
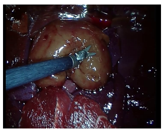
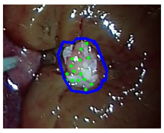
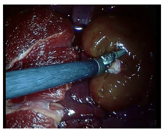
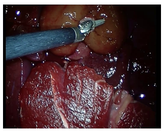
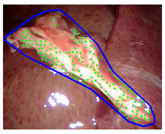
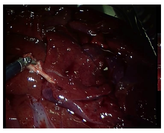
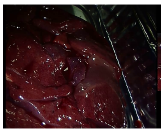
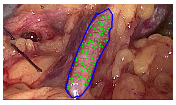
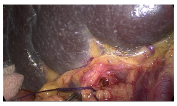
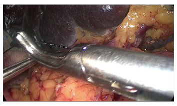
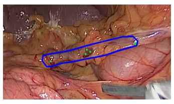
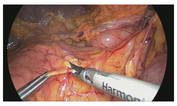
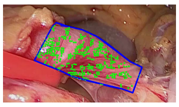
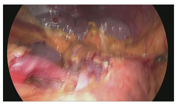
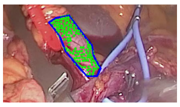
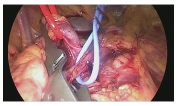
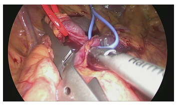
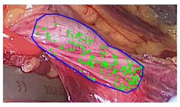
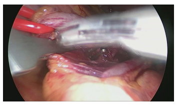
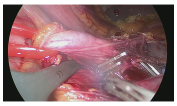
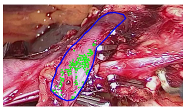
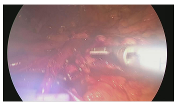
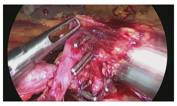
	SA	Frame A	Frame B	res	duration	n frames	description
EV1				720x576	04m:40s	7000	The video shows an exposed goat kidney and the SA is defined on the main vessel entering in the kidney.
EV2				720x576	03m:52s	5800	The video shows a similar surgical field of view as in EV1 with a different kidney.
EV3				720x576	03m:40s	5500	The video shows a pig liver and the SA is defined on a vessel.
IV1				1280x720	04m:03s	6080	These sequences were extracted from a video of a pancreatectomy procedure.
IV2				1280x720	02m:31s	3780	
IV3				1280x720	00m:28s	710	
IV4				1280x720	00m:06s	150	
IV5				1280x720	02m:42s	4070	
IV6				1280x720	02m:01s	3035	

Figure 6: Image samples from the *ex-vivo* (1^{st}-3^{rd} rows) and *in-vivo* (4^{th}-8^{th} rows) datasets. The first column shows the SA defined in the first frame. The second and third columns show two characteristic frames (Frame A & Frame B), showing Partial Occlusion (PO), Total Occlusion (TO), Deformation (DEF), Illumination Changes (IC), Abrupt Camera Motion (ACM), Blur (BLR) or Smoke (SMK). The last columns show, in this order: the image resolution, the duration and number of frames and a brief description of the video sequence content.

R 2

	ex-vivo dataset			*in-vivo* dataset					
	EV1	EV2	EV3	IV1	IV2	IV3	IV4	IV5	IV6
γ_{low}	0.93/**0.95**	0.44/**0.96**	0.97/**0.97**	0.34/**0.60**	0.40/**0.91**	0.61/**0.97**	0.72/**0.96**	0.73/**0.92**	0.85/**0.88**
γ_{medium}	0.93/**0.95**	0.44/**0.96**	0.90/**0.96**	0.34/**0.60**	0.39/**0.69**	0.61/**0.95**	0.72/**0.96**	**0.67**/0.52	0.80/**0.83**
γ_{high}	**0.80**/0.78	0.30/**0.64**	0.20/**0.87**	0.22/**0.39**	0.02/**0.07**	**0.44**/0.23	0.40/**0.86**	**0.16**/0.10	0.22/**0.24**
r_{time}	0.84/**0.47**	37.50/**0.88**	7.00/**2.00**	16.00/**8.04**	3.13/**0.57**	5.17/**0.25**	3.00/**0.00**	**3.49**/12.75	7.11/**5.78**

Table 3: F-measure values (without/with MUpS) for three different overall thresholds (low = 0.2, medium = 0.5, high =0.8) and Recovery Time [# frames] (without/with MUpS)

5. Conclusion

In this paper, we proposed a framework for Long-term Safety Area Tracking (LT-SAT), which aims to be used to

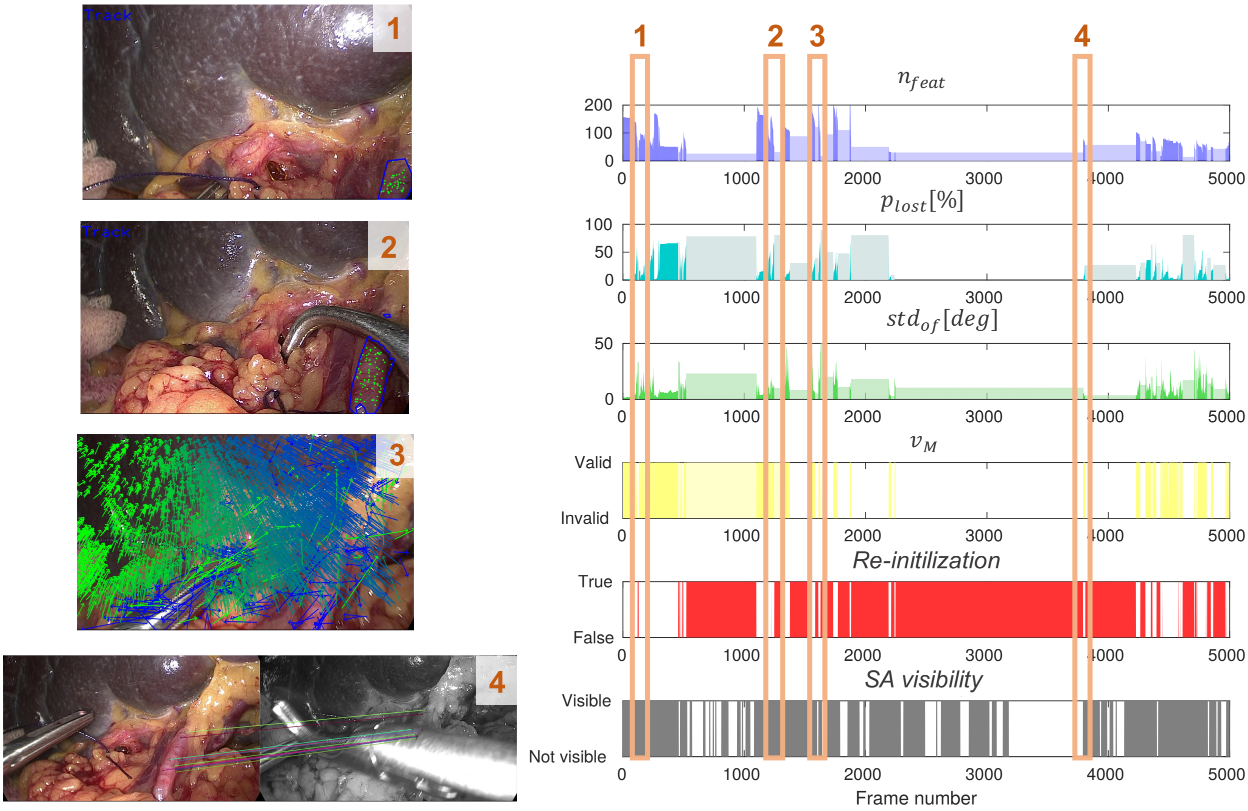


Figure 7: Example of clue values (n_{feat}, p_{lost}, std_{of}, v_M) used to estimate the joint failure probability. On the left, images representative of the clues are shown. The tracked safety area is represented by a blue line and the features by green circles. Image 1 and 2 represent cases in which n_{feat} and p_{lost} decrease due to an OFV and PO. The arrows on the image 3 represent the optical flow computed by KLT tracker, where the colour gradient shows the direction of the arrows and highlights a scattered behaviour around the SA, due to ACM. The lines in image 4 represent the wrong matches between consecutive frames, caused by a PO. On the right, the graphs show the evolution of the clues along the video sequence. The orange boxes highlight the clue values corresponding to the images in the left. The last two rows show, respectively, the frames in which the Tracking Re-initialisation is active and the visibility of the SA. When the framework is in Re-initialisation modality, the clue values are not measurable, and thus they are represented with a lighter colour since their values are not representative.

preserve SA from injury during RMIS. We decided to focus on tracking vessels in the field of abdominal surgery, moved by the need for preventing bleeding during different kinds of surgical procedures. Despite this, we believe that our algorithm is applicable and useful for other applications where it is required the tracking of visible structures in the surgical field of view.

The overall results show that the framework fulfils the main requirements stated in Sec. 1, such as (i) the long-term robustness under complicated conditions, thanks to the combination and improvement of state-of-the-art tracking strategies with a Bayesian-based failure detection scheme; and (ii) the adaptation to environment/object changes, thanks to the novel strategy of updating the models used for the SA re-initialisation. The hybrid combination of KLT tracker, based on GFTT, with tracking-by-detection approach, based on SURF features, aims at exploiting the strengths of the two approaches. The KLT tracker is robust, accurate and computationally cheap in case of small movements, while tracking-by-detection approach would be more computationally expensive and less accurate than KLT if used to track the SA frame-by-frame. The choice of the GFTT for the frame-by-frame tracking is supported by the fact that in robotic surgery, the camera is not manually handled, and it could be robotically held in a fixed position for a long period, resulting in small movements of the SA. On the other hand, the strength of the SURF detector is the invariance to rotation and scale changes that allows to recover the SA, even if it disappears for many frames and reappears in a totally different pose. Here, SURF is strengthened by a generalised Hough Transform approach to discard outliers. Moreover, the implemented method reduces the drift effect caused by incremental tracking in the following manner: (i) In the KLT tracking, the transformation of the SA is done with respect to the key frame and not with respect to the previous frame; (ii) The tracking re-initialisation contributes to clear any degenerative drift behaviour from the KLT tracking by resetting the pose of the SA; (iii) the first model is always kept in the buffer list since newer model could include drift.
The analysis of the *ex-vivo* and *in-vivo* videos show that the framework is capable of maintaining good tracking performance for extended periods of time ($\simeq 4min$), covering the entire video sequences in which the vessel had to be

tracked. The high precision values confirm that the performance of the framework is within the specifications required by the surgeons.

The real improvement given by the proposed framework with respect to state-of-the-art algorithms consists in the robustness, represented by recall values. Particularly, in contrast with the literature, we tested the algorithm on long video sequences (between 5000 and 7000 frame), simulating and considering many of the events happening in surgery and that can affect the performance of the tracking. This extensive evaluation on long video sequences allows to state that the algorithm is able to work properly and robustly in a real surgical scenario.

However, there are few possible cases in which the proposed framework would not recover from tracking failure:

- In case the image is blurred or noisy the recovery of the SA can be delayed, with an average recovery time reported in Tab. 3;
- In case of sudden big deformation, the homograpy transformation computed between sets of matched features can be erroneous since it assumed that the points represented by the features are laying on a plane, and the tracking re-initialisation would not be able to recover the new SA pose, since model recognition 2.3.3 is based on the relation between each feature and the centroid of the model;
- In case the appearance of the SA is greatly changed during a total occlusion. This situation would not allow to the Model Update Strategy (MUpS) to store the model appearance changes, preventing its recovery by the Tracking re-initialisation. In the video sequences used for the evaluation, the LT-SAT tracker with MUpS was always able to recover the model after a failure, with the recovery time reported in Tab. 3.

Thus, the main weakness of the algorithm is the difficulty in tracking the SA during a big deformation. The use of several SAs could help in overcoming this problem (as also stated by Puerto-Souza and Mariottini (2013)). Indeed, a global SA can be considered as a group of spatially distributed SAs, each one assuming its small portion of tissue as a planar surface. Then, multiple local affine transformations can be computed between the SAs' matches. In case of using multiple SAs, our implementation would need to be scaled up taking into account multiple affine transformations in the tracking stage, a global failure probability, multiple buffers of models for the re-initialisation, and deformable mesh model or any other similar strategy to combine the tracking information of each SAs to estimate the global pose of the SA.

The robust long-term tracking of the SA has been proposed in this paper to preserve delicate areas (i.e. vessels) from injuries during surgery. In practice, the 2D tracking of the SA could be used as a prior identification of the area of the image to reconstruct in 3D. Thus, having a 3D point cloud of the delicate area in any time of the surgery, can be exploited to know where this area is located with respect to the robotic instruments (in case a robotic system is used). Visual, auditory or haptic sensory channels can be exploited to warn the surgeon about the SA-instruments distance and augment the operation with such a feedback (Enayati et al., 2016).

Future work will aim at addressing the issue of robust tracking under big deformations, exploiting deformation modelling techniques. Also since the current implementation of the algorithm is not able to run in real time, next steps will include software architecture improvements and code optimisation, which should reduce considerably the computational time. At this point, authors aim at integrating this framework with a dense 3D reconstruction algorithm, already developed by Penza et al. (2016), in order to obtain a 3D area tracking, and integrate the overall system in a robotic platform.

Acknowledgements

Authors would like to thank Max Allan for his kind help during the recording of the videos used for the dataset, and the DVRK community and Intuitive Surgical for the DVRK data.

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

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 labelling and optical flow, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 331–338.

Ballard, D.H., 1981. Generalizing the hough transform to detect arbitrary shapes. Pattern recognition 13, 111–122.

Bay, H., Tuytelaars, T., Van Gool, L., 2006. Surf: Speeded up robust features, in: Computer vision–ECCV 2006. Springer, pp. 404–417.

Bouget, D., Allan, M., Stoyanov, D., Jannin, P., 2017. Vision-based and marker-less surgical tool detection and tracking: a review of the literature. Medical Image Analysis 35, 633–654.

Bouget, D., Benenson, R., Omran, M., Riffaud, L., Schiele, B., Jannin, P., 2015. Detecting surgical tools by modelling local appearance and global shape. IEEE transactions on medical imaging 34, 2603–2617.

Bravo, R., Arroyave, M., Trépanier, J., Lacy, A., 2016. Robotics in general surgery: Update and future perspectives. Advances in Robotics & Automation 2015.

Du, X., Allan, M., Dore, A., Ourselin, S., Hawkes, D., Kelly, J.D., Stoyanov, D., 2016. Combined 2d and 3d tracking of surgical instruments for minimally invasive and robotic-assisted surgery. International journal of computer assisted radiology and surgery 11, 1109–1119.

Du, X., Clancy, N., Arya, S., Hanna, G.B., Kelly, J., Elson, D.S., Stoyanov, D., 2015. Robust surface tracking combining features, intensity and illumination compensation. International journal of computer assisted radiology and surgery 10, 1915–1926.

Duffner, S., Garcia, C., 2013. Pixeltrack: a fast adaptive algorithm for tracking non-rigid objects, in: Proceedings of the IEEE international conference on computer vision, pp. 2480–2487.

Enayati, N., De Momi, E., Ferrigno, G., 2016. Haptics in robot-assisted surgery: challenges and benefits. IEEE reviews in biomedical engineering 9, 49–65.

Faria, C., Sadowsky, O., Bicho, E., Ferrigno, G., Joskowicz, L., Shoham, M., Vivanti, R., De Momi, E., 2014. Validation of a stereo camera system to quantify brain deformation due to breathing and pulsatility. Medical physics 41.

Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24, 381–395. URL: `http://doi.acm.org/10.1145/358669.358692`, doi:`10.1145/358669.358692`.

Forgione, A., 2009. In vivo microrobots for natural orifice transluminal surgery. current status and future perspectives. Surgical oncology 18, 121–129.

Giannarou, S., Visentini-Scarzanella, M., Yang, G.Z., 2013. Probabilistic tracking of affine-invariant anisotropic regions. IEEE transactions on pattern analysis and machine intelligence 35, 130–143.

Kaouk, J.H., Khalifeh, A., Hillyer, S., Haber, G.P., Stein, R.J., Autorino, R., 2012. Robot-assisted laparoscopic partial nephrectomy: step-by-step contemporary technique and surgical outcomes at a single high-volume institution. European urology 62, 553–561.

Lanfranco, A.R., Castellanos, A.E., Desai, J.P., Meyers, W.C., 2004. Robotic surgery: a current perspective. Annals of surgery 239, 14–21.

Lehmann, T.M., Palm, C., 2001. Color line search for illuminant estimation in real-world scenes. JOSA A 18, 2679–2691.

Lorenzo, E.I., Jeong, W., Park, S., Kim, W.T., Hong, S.J., Rha, K.H., 2011. Iliac vein injury due to a damaged hot shears tip cover during robot assisted radical prostatectomy. Yonsei medical journal 52, 365–368.

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. International journal of computer vision 60, 91–110.

Mountney, P., Yang, G.Z., 2012. Context specific descriptors for tracking deforming tissue. Medical image analysis 16, 550–561.

Nicolau, S., Soler, L., Mutter, D., Marescaux, J., 2011. Augmented reality in laparoscopic surgical oncology. Surgical oncology 20, 189–201.

Onda, S., Okamoto, T., Kanehira, M., Suzuki, F., Ito, R., Fujioka, S., Suzuki, N., Hattori, A., Yanaga, K., 2014. Identification of inferior pancreaticoduodenal artery during pancreaticoduodenectomy using augmented reality-based navigation system. Journal of hepato-biliary-pancreatic sciences 21, 281–287.

Opitz, I., Gantert, W., Giger, U., Kocher, T., Krähenbühl, L., et al., 2005. Bleeding remains a major complication during laparoscopic surgery: analysis of the salts database. Langenbeck's Archives of Surgery 390, 128–133.

Penza, V., Ortiz, J., De Momi, E., Forgione, A., Mattos, L., 2014. Virtual assistive system for robotic single incision laparoscopic surgery, in: 4th Joint workshop on computer/robot assisted surgery, pp. 52–55.

Penza, V., Ortiz, J., Mattos, L.S., Forgione, A., De Momi, E., 2016. Dense soft tissue 3d reconstruction refined with super-pixel segmentation for robotic abdominal surgery. International journal of computer assisted radiology and surgery 11, 197–206.

Pezzementi, Z., Voros, S., Hager, G.D., 2009. Articulated object tracking by rendering consistent appearance parts, in: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, IEEE. pp. 3940–3947.

Puerto-Souza, G.A., Cadeddu, J.A., Mariottini, G.L., 2014. Toward long-term and accurate augmented-reality for monocular endoscopic videos. IEEE Transactions on Biomedical Engineering 61, 2609–2620.

Puerto-Souza, G.A., Mariottini, G.L., 2013. A fast and accurate feature-matching algorithm for minimally-invasive endoscopic images. IEEE transactions on medical imaging 32, 1201–1214.

Schoob, A., Laves, M.H., Kahrs, L.A., Ortmaier, T., 2016. Soft tissue motion tracking with application to tablet-based incision planning in laser surgery. International journal of computer assisted radiology and surgery , 1–13.

Seib, V., Kusenbach, M., Thierfelder, S., Paulus, D., 2012. Object recognition using hough-transform clustering of surf features. RoboCup@ home Technical Challenge .

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, IEEE. pp. 593–600.

Sotelo, R., Bragayrac, L.A.N., Machuca, V., Cortes, R.G., Azhar, R.A., 2014. Avoiding and managing vascular injury during robotic-assisted radical prostatectomy. Therapeutic advances in urology , 1756287214553967.

Stoyanov, D., 2012a. Stereoscopic scene flow for robotic assisted minimally invasive surgery, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 479–486.

Stoyanov, D., 2012b. Surgical vision. Annals of biomedical engineering 40, 332–345.

Tomasi, C., Kanade, T., 1991. Detection and tracking of point features. School of Computer Science, Carnegie Mellon Univ. Pittsburgh.

Trinh, Q.D., Sammon, J., Sun, M., Ravi, P., Ghani, K.R., Bianchi, M., Jeong, W., Shariat, S.F., Hansen, J., Schmitges, J., et al., 2012. Perioperative outcomes of robot-assisted radical prostatectomy compared with open radical prostatectomy: results from the nationwide inpatient sample. European urology 61, 679–685.

Wu, Y., Lim, J., Yang, M.H., 2013. Online object tracking: A benchmark, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2411–2418.

Yang, H., Shao, L., Zheng, F., Wang, L., Song, Z., 2011. Recent advances and trends in visual tracking: A review. Neurocomputing 74, 3823–3831.

Ye, M., Giannarou, S., Meining, A., Yang, G.Z., 2016. Online tracking and retargeting with applications to optical biopsy in gastrointestinal endoscopic examinations. Medical image analysis 30, 144–157.

Yip, M.C., Lowe, D.G., Salcudean, S.E., Rohling, R.N., Nguan, C.Y., 2012. Tissue tracking and registration for image-guided surgery. IEEE transactions on medical imaging 31, 2169–2182.