# Detecting Surgical Tools by Modelling Local Appearance and Global Shape

David Bouget*, Rodrigo Benenson, Mohamed Omran, Laurent Riffaud, Bernt Schiele, and Pierre Jannin

***Abstract*—Detecting tools in surgical videos is an important ingredient for context-aware computer-assisted surgical systems. To this end, we present a new surgical tool detection dataset and a method for joint tool detection and pose estimation in 2d images. Our two-stage pipeline is data-driven and relaxes strong assumptions made by previous works regarding the geometry, number, and position of tools in the image. The first stage classifies each pixel based on local appearance only, while the second stage evaluates a tool-specific shape template to enforce global shape. Both local appearance and global shape are learned from training data. Our method is validated on a new surgical tool dataset of 2 476 images from neurosurgical microscopes, which is made freely available. It improves over existing datasets in size, diversity and detail of annotation. We show that our method significantly improves over competitive baselines from the computer vision field. We achieve 15% detection miss-rate at $10^{-1}$ false positives per image (for the suction tube) over our surgical tool dataset. Results indicate that performing semantic labelling as an intermediate task is key for high quality detection.**

***Index Terms*—Microscope images, object detection, surgical tools, template matching.**

## I. Introduction

PREVENTABLE medical errors in the operating room occur frequently enough to cost tens of thousands of human lives per year in the USA [1]. To reduce such human errors, the medical technology community seeks to augment the capabilities of the surgeon with context-aware computer-assisted surgical systems [2], [3]. The aim of such systems is to optimally inform and guide the surgeon in real-time during the operation according to ongoing surgical tasks. One of the best solutions to recognize a surgical task is to identify surgical tools used and their behaviours (e.g., trajectories). Accurate and fast (i.e., speed of the recording device) tool detection and pose estimation on existing imaging setups are key components to enable the deployment of context-aware systems with minimal changes to existing operating rooms [4].

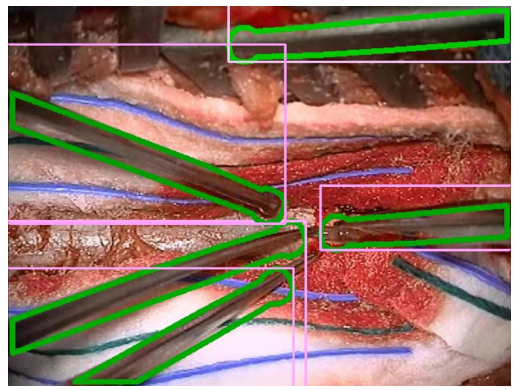


Fig. 1. Example detection results on the new surgical dataset.

### A. Related Work

In a medical context, early works proposed to modify the physical integrity of surgical tools by the addition of external markers with the motivation to ease image-based detection processes. Black multiple-part tags with recognizable patterns [5] and also colour tags varying in size and position [6] have been investigated. More technologically advanced external markers such as light-emitting diodes [7] or RFID tags [8] have also been considered. However, approaches requiring to apply physical modifications to surgical tools encounter many regulation issues as they raise safety concerns, and are not straightforwardly transferable into the operating theatre.

From the literature, two principal categories of image-based techniques arise: techniques performing full-image analysis and techniques re-using the knowledge of detected tools in previous frames through tracking algorithms, the latter being the favoured one. Approaches have included filtering-tracking via particle filters [9] or Kalman filters [10], [11]; contour-tracking relying on the CONDENSATION algorithm [12], [13]; region-based tracking using mutual information as similarity measure [14]; and feature matching from FAST corners [15]. Kumar *et al.* [16] proposed an interesting line of work dwelling in an optimal fusion between outputs from various trackers, taking advantage of feature-based tracking robustness to small motion and region-based tracking robustness to significant motion. Nevertheless, all class-specific tracking methods require a class-specific detector for (re-)initialisation of the tracking procedure. Although high average speed might be obtained using

Manuscript received April 15, 2015; revised June 15, 2015; accepted June 22, 2015. Date of publication June 29, 2015; date of current version November 25, 2015. The work of D. Bouget was supported by Carl Zeiss Meditec AG. *Asterisk indicates corresponding author.*

*D. Bouget is with the Medicis team, INSERM U1099, Université de Rennes 1 LTSI, 35000 Rennes, France (e-mail: david.bouget@univ-rennes1.fr).

R. Benenson, M. Omran, and B. Schiele are with the Department of Computer Vision and Multimodal Computing, Max-Planck Institute for Informatics, 66123 Saarbrücken, Germany (e-mail: first.lastname@mpi-inf.mpg.de).

L. Riffaud is with the Department of Neurosurgery, Rennes University Hospital, 35000 Rennes, France (e-mail: laurent.riffaud@chu-rennes.fr).

P. Jannin is with the Medicis team, INSERM U1099, Université de Rennes 1 LTSI, 35000 Rennes, France (e-mail: pierre.jannin@univ-rennes1.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMI.2015.2450831

tracking, the frame by frame speed and quality are bounded by the detection method employed. This is why we focus on the detection task, and leave aside tracking.

Detection methods, either implemented in a stand-alone fashion or for the purpose of tracking initialization, can belong to one of the three following groups: ad-hoc image-processing techniques, data-driven approaches that directly leverage features extracted from the input image (named single-stage), and data-driven approaches requiring an intermediary step (named two-stage).

Among the ad-hoc techniques, Voros *et al.* [17] performed a succession of mathematical morphology operations using the 3d tool insertion position in the abdominal cavity and the shape information to strongly constrain the detection search space. In Haase *et al.* [18], a three-layer approach encompassing clustering and Hough fitting operations has been proposed, assuming rigid tools with cylindrical shaft entering the scene from image boundaries.

Single-stage approaches include the work of Kumar *et al.* [16], in which they propose to model instruments by parts using HOG features, and perform the detection process through Latent Support Vector Machine (LSVM) classification. For retinal microsurgery purposes, Sznitman *et al.* [19] proposed to use a deformable detector where edge features are computed and fed to an AdaBoost algorithm as model learning strategy [20]. Their detector is robust to in-plane rotations but the evaluation was only based on a single point detection without rotation estimate, and a single tool was present in the videos. In another work, Sznitman *et al.* [21] proposed an algorithm to detect needle-shaped objects, by propagating hypotheses starting from the image border. Such an approach seems unable to handle tools occluded around the image boundaries, overlapping tools, or tools without a rectilinear tubular shape.

Lastly, in two-stage approaches, the detector's first stage involves classifying each pixel of the input image as either "instrument" or "background". On top of response scores from this classification, the second stage results in an estimate of the tool's pose, i.e., instrument location, extent, and orientation. In Pezzementi *et al.* [22], a Gaussian mixture model using colour and texture features is used to perform the first stage, while a known by-part 3d model of the tool is iteratively projected (rotation and translation) on the resulting label mask to find the optimal object pose using maximum likelihood estimation. Tackling 3d-pose estimation challenges, Allan *et al.* [23] performed the pixel-wise classification using Random Forests based on a combination of colour, HOG, and SIFT features. Tool positions are retrieved from the semantic labels map using a flooding algorithm to identify the largest connected components. Underlying shapes are analysed using the moment of inertia tensor to retrieve principal orientation axes. The pose is refined within each region using an energy function and prior information of the tool geometry for the 2d to 3d lifting to obtain final 3d pose estimates. Their approach assumes a known number of tools, with known 3d geometry, and expects the tool to be visible at the image borders. In Sznitman *et al.* [24], an instrument-part detector has been proposed, with an early stopping scheme for speed efficiency. The multi-class classifier is combining the gradient boosting framework with edges features to assign an instrument-part or background label to each pixel of an image. Then, the different parts of the instrument are estimated by weighted averaging on the response scores. The overall instrument orientation is retrieved using RANSAC, by fitting the estimated shape of the instrument (i.e., a line) over the resulting labelled image.

Usually only two classes are modelled for the pixel-wise classification, one to represent tool pixels and one for background pixels [22], [25]. However, the possibility exists to represent one instrument with more than one label, which is particularly interesting for part-based detection purposes [11].

Whichever the tool detection strategy, many existing approaches rely on a set of assumptions or prior knowledge to constrain the search space, hence facilitating the task. Such knowledge having different forms and aspects, four groups can be identified for its representation: assumptions on instruments' location in the image, assumptions on instruments' shape, external assistance from a robotic system, or human assistance. Surgical tools were often assumed to be simple tubular shapes [24], [26], solid cylinders with a tip alongside the centre-line [10], [18], [23], or rough estimates such as two parallel side segments [17], [25]. Instruments' location assumptions relate to appearance and disappearance from the field-of-view, as tools must intersect with image boundaries to be visible [13], [21], [23]. When using robotic surgical systems, information provided by internal encoders represents a good estimate of tool positions [10], or can be used to render on-the-fly models with a limited set of joint configurations [27]. Finally, the user can be asked to manually identify the image region to track for online learning methods [15]. Using prior knowledge or an extended set of assumptions, detection methods may or may not transfer well from their design space to other surgical contexts or instruments, and as such can be detrimental for the creation of generic approaches.

The aim of this work is to jointly detect surgical tools and retrieve their pose in 2-dimensional monocular in-vivo images, gathered from operating microscopes. We consider the pose to be described by a limited number of parameters: overall position, orientation, and tip location. Our proposed approach is built upon the strategy of two-stage framework detectors, and attempts to relax assumptions on the number of tools, their shape, and their position in the image. The pixel-wise classification (so-called semantic labelling) is performed using a TextonBoost-like approach [28]. For each surgical tool category, we propose to learn a shape model from training data using a linear SVM integrating a spatial regularisation term. The pose estimation step is evaluating such models in a brute-force manner over the pixel-wise classification results in a sliding-window fashion. Even though being computationally intense, the method is well-suited for GPU parallel computing and is able to perform surgical tool detection in real-time.

### B. Existing Datasets

For the surgical tool detection task, no reference dataset exists, hence none of the previous works has been compared over the same data (nor using the same procedure). In addition, it is not yet common practice to release annotated datasets, and thus

it is difficult to perform comparative studies with published results. A few public datasets containing in-vivo surgical images, eligible for our study, are available:

- A robotic tool dataset focusing on surgery performed using the DaVinci robot [16]. It contains a total of 1 950 frames from 12 different stereoscopic videos (average length of 4 seconds). Only two (fixed) tools are visible at any time and annotations are bounding boxes around the tool. We argue that for a proper evaluation more precise annotations are needed.
- A set of retinal and laparoscopic videos with bounding box and centre point annotations [19]. The retinal surgery set contains 1 500 frames (from 4 videos). Retinal images have very homogeneous backgrounds, and contain one tool at most, rendering the detection problem significantly simpler than our setup. The laparoscopic set contains 1 000 images from a single video, showing two instruments per image. With a single video as source, the dataset lacks diversity and proper training and testing splits.
- A video dataset depicting minimally-invasive surgery (MIS) [23]. It contains ~100 images (from 6 videos) with pixel-wise annotations for the tools. Our proposed dataset is 20 $\times$ larger and contains additional annotations.
- A set of 40 in-vivo video sequences recorded from robotic-assisted MIS procedures, involving scale and rotation changes [29]. No tool annotations are provided with the videos.

Main limitations of these datasets include a lack in data, diversity or precision in annotations, making them unsuitable as reference for comparison amongst methods.

In this paper, we first present a new publicly available surgical images dataset (described in Section II) that is larger than previous datasets and has tight annotations around the tools (bounding polygons). Section III describes our two-stage approach which makes no a-priori assumptions on the number of visible surgical tools, their shape, or their relative positions in the image. In Section IV and Section V, we present the evaluation methodology, a set of baseline methods on this task and show the importance of using methods including feature learning. Finally, sections Sections VI and VII provide a discussion, conclusions and future work directions.

## II. The `NeuroSurgicalTools` Dataset

### A. Dataset Creation

This new dataset is derived from a set of 14 monocular videos captured via "Zeiss OPMI Pentero classic" microscopes (720 $\times$ 576 pixels at 25 fps) during in-vivo surgeries performed at CHU Pontchaillou, Rennes. The videos depict different operations, more specifically brain and spine tumour removal procedures. Illumination and camera parameters differ slightly among the videos.

In order to remove side-effects on still images when extracted from interlaced videos, each sequence has been re-encoded for a final video resolution of 612 $\times$ 460 pixels. After sampling at 1 Hz, a random selection is performed to assemble the proposed `NeuroSurgicalTools` dataset which consists of 2 476 frames. Seven different tool categories are featured for a total of 3 819 different tool appearances (see Table I), detailed statistics are provided in Section II-C. We also suggest a balanced train and test split with 1 221 and 1 255 images respectively.

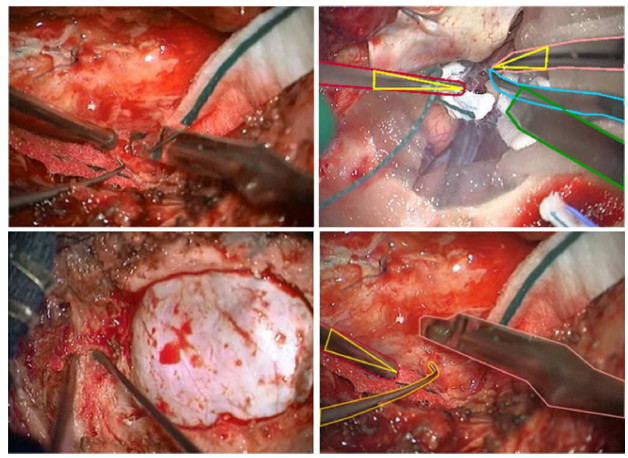


Fig. 2. Example dataset frames (left column) and annotations (right column). In the upper right image, red: suction tube, green: retractor, blue/pink: bipolar forceps, yellow: triangles encoding position and orientation of the tool.

TABLE I
Annotated Tools Distribution

| Surgical tool | Train/Test |
|---|---|
| Suction tube | 900/844 |
| Bipolar forceps | 538/460 |
| Retractors | 157/140 |
| Hook | 163/88 |
| Scalpel | 55/130 |
| Pliers | 81/79 |
| Scissors | 33/30 |
| Others | 0/121 |
| Total = 3 819 | 1 927/1 892 |

The selected images cover a wide range of situations and challenging conditions typically observed during tumour removal procedures: tools overlapping each other, tools occluded by anatomical structures or a surgeon's fingers, tools covered by blood, tools severely blurred from motion, and specular reflections.

The dataset is fully anonymised and available at https://medicis.univ-rennes1.fr/software.

### B. Annotation Protocol

Every tool in each image is annotated with a bounding polygon and a class label. For multiple-part instruments (e.g., bipolar forceps or pliers), each part has a distinctive class label. The suction tube and the upper part of the bipolar forceps are additionally annotated with an isosceles triangle encoding the tool orientation, its width, and its tip position (see Fig. 2). Annotations were done by a domain expert using the LabelMe software [30].

### C. Dataset Statistics

Leaving aside retractors for this calculation as they remain mostly static throughout a surgical procedure, about 27% of the frames contain no surgical tools, while 50% have two and only 12% exhibit three simultaneously. For a deeper description of the dataset, we report below in-plane orientation, scale, and tip

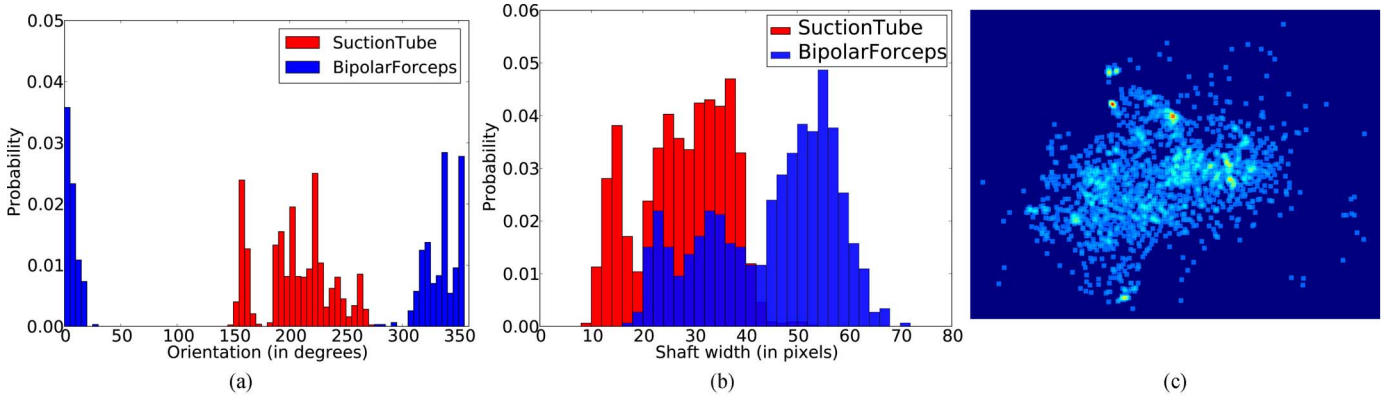



Fig. 3. Suction tube and bipolar forceps statistics computed over the `NeuroSurgicalTools` data-set. (a) In-plane orientation distributions. (b) Scale size distributions (shaft width as reference). (c) Heat map representing tool-tip location.

location distributions for the tools. Since deriving these statistics requires detailed annotations, we exclusively consider the suction tube and the upper part of the bipolar forceps.

*1) Orientation Statistics:* During surgeries, surgical tools undergo in-plane rotations in a range mainly constrained by the surgeon's dexterity. In Fig. 3(a), we report for each of the two aforementioned tool categories their orientation distributions. For reference, we consider orientation 0° to represent a surgical tool horizontally aligned with its tip facing the left image border. Surgeons often using a suction tube and a bipolar forceps concurrently, in addition to both orientation ranges not overlapping, is implying a similar hand dexterity for all the surgeons featured in the dataset. Given orientation ranges for the bipolar forceps between $[0°, 30°]$ and $[320°, 360°]$, we can assume all surgeons to be right-handed as this tool is consistently used by the dominant hand. Similarly, suction tubes in the range $[150°, 270°]$ indicate left hand manipulation. Relative to a vertical image-centred axis, a symmetry can be noticed between instrument positioning, suggesting an optimal placement of surgeons' hands when facing the operating field.

*2) Scale Statistics:* Surgical videos being recorded with different microscope parameters, especially the zoom value, surgical tools appear at different scale sizes. In Fig. 3(b), suction tube and bipolar forceps scale distributions are reported in an histogram fashion using the tool shaft width as reference value. The vast majority of suction tubes, around 75%, appear with a shaft width in-between 20 and 40 pixels. The bipolar forceps is a comparatively bigger tool, mostly with a shaft width in-between 40 and 60 pixels (around 60%).

*3) Position Statistics:* To report position statistics, we compute tool-tip locations over the data-set and plot the resulting heat map in Fig. 3(c), mixing suction tube and bipolar forceps statistics. As can be seen, tool-tips are located within an image-centred circular region that covers a large part of the frame, consistent with surgical microscopes focusing on anatomical structures where the surgeon is operating. Few tool-tips are noticeably close to image borders, representing surgical tools entering or leaving the field-of-view.

## III. Detecting Tools Using Semantic Labelling

Although surgical tools usually do not have a distinctive colour (due to reflections, and grey tissue) or texture (some organs and bones are also untextured), they do exhibit a distinctive local structure. We thus propose a two-stage detection approach. The first stage performs local appearance decisions by classifying each pixel into "tool" or "background" (so called "semantic labelling" task: steps 1 and 2 of Fig. 4). The second stage enforces the global shape by evaluating a tool-specific shape template on top of semantic labelling results (step 3).

### A. Semantic Labelling

In order to classify each pixel as being part of a tool or not, we propose to use the `SquaresChnFtrs` integral channel approach [31], [32]. This classifier is a boosted decision forest over selected feature channels. It was originally proposed for the detection task, however it is suitable for semantic labelling too [28]. The integral channels approach is interesting because of its flexibility in leveraging different feature channels and its strong performance (shown for pedestrian detection [32]).

We consider channels that capture gradient, colour, texture, and position information. `HOG` features are 7 channels, one for gradient magnitude, and six for oriented gradient magnitude. `LUV` are 3 colour channels. `CN` are 11 channels that correspond to common named colours [33]. `FB` are 8 filter bank channels (similar to the ones in [28], [34]) which aim to capture texture information. Finally, `XY` are the normalised vertical and horizontal coordinates.

Our $41 \times 41$ pixel model uses 500 level-2 decision trees, each consisting of three decision stumps, and is trained using AdaBoost. We select each split function per decision stump by minimising the 0–1 loss, which amounts to an exhaustive search over the set of features and corresponding split thresholds. A feature in this case is a sum over a square region in one particular feature channel. Our feature pool consists of all possible square regions inside the model window (see [32] for further details). Our preliminary experiments indicate that a larger model window size or an increased number of weak classifiers has little to no effect on the labelling quality, as reported in Section V-A. Using shallow trees is a form of regularisation.

Since all considered tools have a similar local appearance, we train a single classifier for a "generic tool" class, and a second one for the "background" class. Using two classifiers avoids relying on a single sensitive threshold, thus providing more accurate results. Confidence scores of both classifiers (see

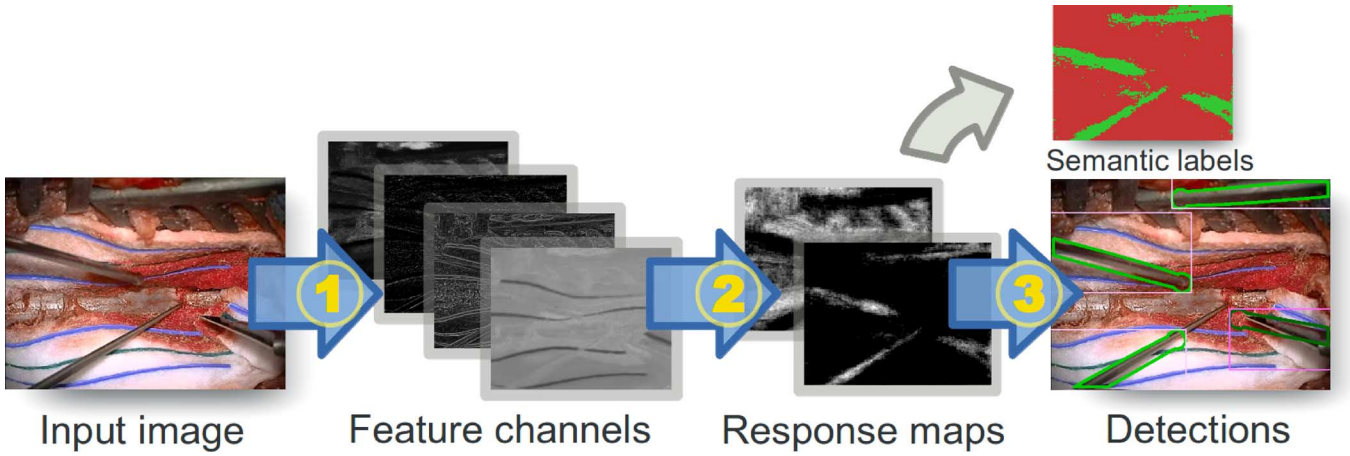



Fig. 4. Overview of the proposed pipeline. Step 1 computes a set of integral feature channels from the input image. Step 2 performs the pixel-wise classification (or semantic labelling) for two classes: tool and background. Step 3 represents the pose estimation process using SVM shape models. Either multiple response maps (i.e., semantic scores) or a single map of semantic labels are eligible as input for the pose estimation.

response maps in Fig. 4) are used as input to the shape-based detection process. The number of response maps is equivalent to the number of classes (i.e., two in this case), but can be extended to any number of classes. We will refer to these multi-class outputs as semantic scores. Alternatively, a single response map can be obtained by computing a pixel-wise $\arg\max$ over each response map. The maximum score across all classes (maps) determines the label of a pixel (see Semantic labels in Fig. 4 or Fig. 5). Shape-based detection methods presented in Section IV-D use either semantic scores or semantic labels as input.

### B. Shape-Based Detection

In our two-stage approach, we propose to capture the global shape of a specific tool using a single rigid template. This template is a linear model that combines the output of the semantic labelling component into a detection score, without using any additional features (see Fig. 10).

*1) SVM Training:* We learn such a template via a linear SVM, with positive training samples normalized for translation, rotation, and scale; negatives are randomly sampled. We also consider regularising the SVM training by adding a 2d spatial smoothness prior. Details of the SVM training are discussed in Section III-C.

*2) SVM Testing:* For each tool category, the SVM model is learned over a set of normalized pose images. To detect objects at different scales and orientations during test time, the SVM template is transformed for each desired scale and orientation (similar to [35]). This speeds-up test time computation, since it avoids the need to recompute the semantic labelling at different scales and orientations. The set of templates are evaluated in a sliding-window fashion.

For further speed-up, each shape template is approximated piecewise via a set of squares (see Fig. 6). To perform this approximation, the SVM model window is sub-divided into 15 $\times$ 15 pixel squares, after addition of extra padding to avoid uneven square size. A new weight is set for each piece, computed by averaging SVM values within the square. For the number of pieces created to be stable across the various scales processed, the model scale coefficient is applied to the square size. This enables using integral images when evaluating the correlation of each scale/orientation specific template over the semantic labelling results. Using integral channels makes the computation cost of the sliding window independent of the template scale. Searching for small tools costs as much as looking for large ones.

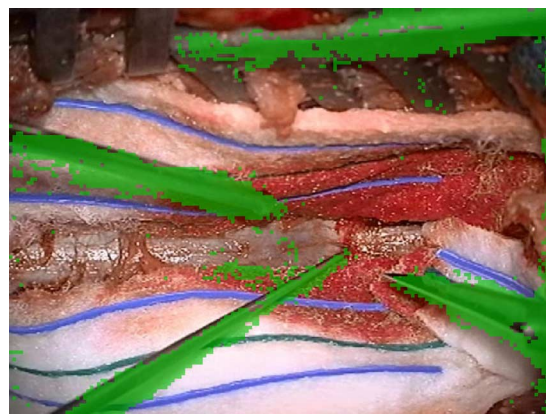


Fig. 5. Semantic labelling results using the `SquaresChnFtrs` integral channels approach. Detected tool pixels are marked in green.

Each candidate detection consists of a score, a bounding polygon on the hypothesized object, a tool-tip position, and an orientation. To eliminate spurious detection hypotheses, we apply a form of greedy non-maximum suppression (NMS) which suppresses multiple nearby detections. The NMS procedure removes the less confident of every pair of detections that overlap sufficiently according to the polygon overlap criterion (as presented in Section IV-A), only if the difference in orientation is lower than a threshold. Our NMS has thus two parameters, the overlap threshold and the orientation difference threshold. By setting the orientation difference threshold to 0, we fall back to the simplified NMS procedure [31]. Using such an orientation difference threshold is meant to allow detections of surgical instruments crossing each other.

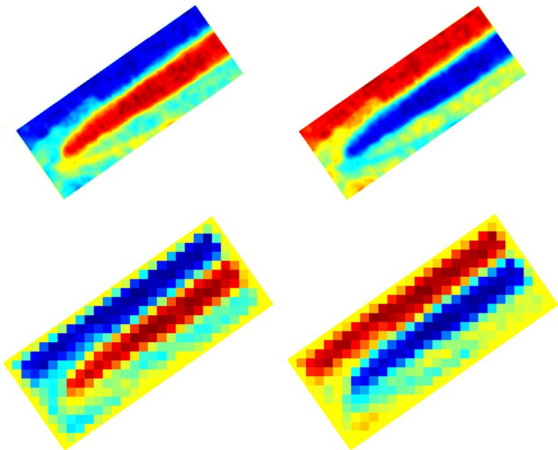


Fig. 6. Original bipolar forceps (upper part) SVM model (top) and its piecewise approximation (bottom). Left column corresponds to the tool class response map and right column to the background class response map. Cool colors represent negative SVM weights, warm color represent positive SVM weights and null weights are colored in yellow.

*3) Benefits:* Our learning-driven approach makes no assumption about the texture or shape of the object. By conducting an exhaustive search we can detect an arbitrary number of tools, at any position and orientation in the image. Finally, the non-maximum suppression allows us to detect tools crossing each other. In Section V, we show a significant improvement over all baselines.

*4) Computational Cost:* Using a two-stage approach is also beneficial speed-wise. Assuming a restricted depth range, semantic labelling can be applied over the image at a single scale, which is common practice in street scene labelling [36]. From the learned model, we prepare an exhaustive set of templates to cover every possible scale and orientation. Using a piecewise approximation of the shape template, integral images over the semantic labelling results can be leveraged to directly detect tools at different scales and orientations without having to recompute features. This is key for high speed detections [35] and it makes the detector eligible for efficient parallel computing.

### C. SVM Training Details

In this subsection, we explain the strategy used to create a tool specific SVM model. We train models of size $125 \times 300$ pixels, with a width/height aspect ratio being kept fixed when preparing the exhaustive set of templates at multiple scales and orientations.

*1) Training Data:* The annotated dataset described in Section II enables us to generate training samples. All positive samples (i.e., showing a tool) are aligned to compensate for translation, scale, and rotation. Compensated training images (as shown in Fig. 7(d)) respect the following parameters: the tool is vertically centred at a 30-pixel distance from the left image border, with a shaft width of 40 pixels (considered to be scale 1), and at orientation 0° . Then, multiple options for generating the training samples exist, of which we consider three:

1) `SquaresChnFtrs` semantic labelling maps (see Section III-A and Fig. 7(b)).

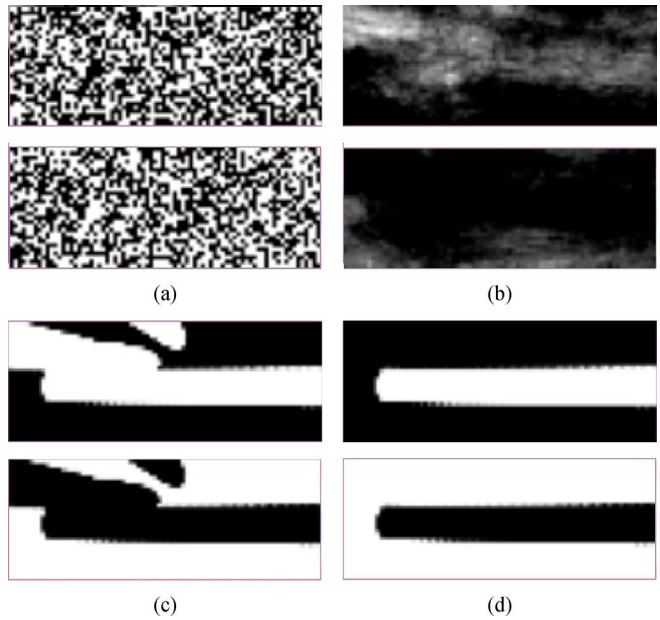



Fig. 7. SVM inputs. Top rows correspond to the tool class response map and bottom rows to the background class response map. (a) Negative. (b) Semantic labelling data. (c) All instruments annotation. (d) Single instrument annotation.

2) Annotations of all surgical instruments (see Fig. 7(c)).

3) Annotation of a single surgical instrument (see Fig. 7(d)).

While semantic labelling maps (alternative (1)) represents best the data the classifier will receive at test time, they are somewhat noisy. This noise makes it difficult to learn the shape of the surgical instrument of interest. To remedy this, we propose to create binary images using surgical instrument annotations. In the "tool" class response map, annotated surgical instruments are in white and the rest of the image in black (the "background" class response map being the reverse case). These binary images can be considered as ideal semantic labelling results (alternative (2)). For the last alternative we also use annotations, but this time only the one surgical instrument of interest is in white and the rest of the image, including neighbour surgical instruments, is in black.

Negative images are randomly sampled from a uniform distribution for the "tool" class response map. The "background" class response map is created as the opposite image (Fig. 7(a)). Using the opposite image is meant to mimic the ideal semantic labelling case. For the uniform distribution, we consider two alternatives: a) a binary distribution where pixels can only have the value 0 or 255, and b) grey-scale distribution [0, 255] to match with semantic labelling data inputs. We report below experimental results on the effect of different training samples.

*2) Regularisation:* Regularisation is an important aspect for SVM training. Since we know that we are operating on a two dimensional domain, we consider modifying the vanilla SVM formulation shown in (1) (see [37]), to include a regularisation term $M$ that promotes a 2d spatial smoothness prior [38].

$$\min_{w \in \mathbb{R}^m} \; w^T w + \frac{C}{|T|} \sum_{t \in T} L\left(y_t, \; \langle x_t, w \rangle\right) \tag{1}$$

$$\min_{w \in \mathbb{R}^m} \; w^T M w + \frac{C}{|T|} \sum_{t \in T} L\left(y_t, \; \langle x_t, w \rangle\right) \tag{2}$$

where $T = \{x_t,\ y_t\}_{t=1}^{|T|}$ are instance-label pairs, $L : \{0,1\} \times \mathbb{R} \rightarrow \mathbb{R}_0^+$ is the loss function and C is a penalty parameter. The matrix $M$ can be decomposed as shown in (3). The regularisation matrix $R$ encodes the 2d spatial structure.

$$M = R^T R. \tag{3}$$

In (4), we develop the link between the standard SVM formulation and the one using regularisation via $R$. It can be seen that the 2d prior can be encoded via a simple transformation of the input data (via $R^{-1}$), allowing the use of unmodified SVM training code. At test time we use the resulting $w$, without needing to change the input data.

$$\begin{aligned}
w^T M w + \frac{C}{|T|}\sum_{t\in T} L\left(y_t,\ \langle x_t, w\rangle\right)\\
w^T (R^T R) w + \frac{C}{|T|}\sum_{t\in T} L\left(y_t,\ \langle x_t, (R^{-1}R) w\rangle\right)\\
(Rw)^T (Rw) + \frac{C}{|T|}\sum_{t\in T} L\left(y_t,\ \langle x_t, R^{-1}(Rw)\rangle\right)\\
\tilde{w}^T \tilde{w} + \frac{C}{|T|}\sum_{t\in T} L\left(y_t,\ \langle x_t, R^{-1}\tilde{w}\rangle\right)\\
w = R^{-1}\tilde{w}.
\end{aligned} \tag{4}$$

The 2d spatial smoothness in the regularisation matrix is performed by enforcing 4-connected pixels to have close values. In case of a 4-pixel image (a, b, c, and d being its four pixels), the regularisation matrix to use is represented in (5). For comparison, (6) and (7) illustrate the regularisation term with and without 2d spatial smoothness.

$$w = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix},\ \tilde{w} = Rw = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \tag{5}$$

$$\begin{aligned}
\tilde{w}^T \tilde{w} &= a^2 + b^2 + c^2 + d^2 \\
&\quad + (a-b)^2 + (a-c)^2 + (b-d)^2 + (c-d)^2
\end{aligned} \tag{6}$$

$$w^T w = a^2 + b^2 + c^2 + d^2. \tag{7}$$

For creating the regularisation matrix, only two parameters are necessary: the number of rows and columns of SVM training samples. $R^{-1}$ is computed once beforehand, and simple transformations of the SVM samples via $R$ and $R^{-1}$ are applied during the training process. In Section V, we evaluate the impact of using such regularisation schema.

## IV. Validation Studies

In this work we aim at detecting tools, and leave aside the problem of tool categorisation. When evaluating the detection of a specific tool we ignore all false positives on other annotated tools. This is similar to the protocol used for pedestrian detection [31], where regions with "crowds" are ignored (related class that triggers false positives for pedestrians). False positives on other tools are considered part of the (fine-grained) tool classification problem, left for future work. Similarly to the scheme laid out by Dollar *et al.* [39], a full image evaluation is performed between the set of candidate detections obtained by the detection method and the corresponding set of references. We use the log-average miss rate to summarise detector performance, computed by averaging miss rate at nine false positives per image rates (FPPI) evenly spaced in log-space in the range $10^{-2}$ to $10^{\circ}$. The minimum miss rate is used for curves that end before reaching the FPPI upper bound.

Train and test image sets have been presented in Section II-A. The train set has been used for every learning process, while detector performances have been evaluated over the test set. In the following, we start by presenting the evaluation metrics considered to obtain performance results. Then, to understand the difficulty of detecting surgical tools from in-vivo surgery images, we consider different baselines for comparison with our proposed method.

### A. Evaluation Metrics

Multiple metrics are of interest depending on the specific applications in mind, and the type of reference available (i.e., manual annotation). In the experimental Sections V-A and V-C, we consider the following four evaluation metrics. The first metric provides overall tool detection performance, the second and third ones further assess the pose estimation quality through orientation and tip position accuracy. The fourth one evaluates the semantic labelling quality.

*1) Polygon Overlap:* Due to surgical tools' elongated shapes, we evaluate detections using bounding polygons instead of bounding boxes aligned to the image border. We use the traditional "intersection over union" criterion to count false positives and false negatives [40]. Since a small difference in orientation between two elongated polygons leads to small overlapping areas, we consider true detections to be those with an overlap $\geqslant 25\%$ with the ground truth annotation. Arguably this evaluation improves over previous work that considered only bounding box overlap [16].

*2) In-Plane Orientation Difference:* Given many in-plane tool rotations during surgeries, we propose for every true detection obtained at a fixed rate of $10^{-1}$ FPPI to observe the error in the orientation estimation. The orientation difference is computed in degrees between a detection and its corresponding reference. To display the results, we plot the percentage of correct detections orientation-wise as a function of the difference in orientation.

*3) Tool-Tip Distance:* In some applications, the tool-tip position is more relevant than the tool-body pose estimation. We can thus measure the Euclidean distance between a detection and its corresponding ground truth tool-tip. To ensure more meaningful results, we compare methods at a fixed rate of $10^{-1}$ false positives per image, and disregard detections deviating by more than 45 degrees from the ground truth. This measure is optimistic given many false positives, but gives an upper-bound on the tool-tip precision when detections are correct.

*4) Segmentation Quality:* Our dataset annotations allow us to generate per-pixel ground-truth label maps. The results of the next section show that using pixel-wise labelling enables better

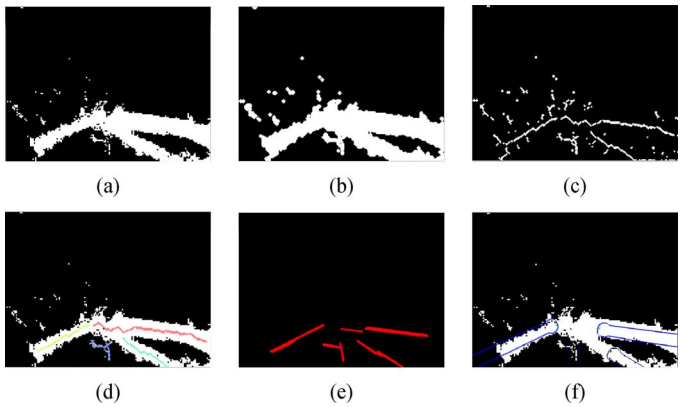



Fig. 8. Illustrated `Skeleton` approach workflow. From an initial image of semantic labels, a set of morphological operation is applied in order to retrieve as many skeletons as surgical instruments. Candidate detections are identified after connected component labelling and Hough line fitting. (a) Input mask. (b) Dilation. (c) Skeletisation. (d) Connected components labelling. (e) Hough transform. (f) Output detections.

detection, thus we are interested in evaluating this intermediary step. Since in the dataset all tools have similar local appearance, we evaluate all tools as a single class. We report the average (per-class) pixel classification accuracy of tools versus background.

### B. Baseline: Semantic Labelling

We use the open source toolbox `Darwin` [34] as a baseline for semantic labelling. It implements a state-of-the-art method (inspired by [28]) based on boosted decision trees built on top of features comprising filter banks, HOG, and RGB colour. The main difference between `Darwin` and `SquaresChnFtrs` is the former's use of more hand-crafted features and pooling regions.

### C. Baseline: Single-Stage Detection

`Linemod` is a real-time detection method for texture-less objects [41]. It is based on fast matching of oriented gradient templates. Surgical tools being mainly texture-less objects, `Linemod` is expected to perform well, and thus serves as a good baseline.

`SquaresChnFtrs` is the classifier we use for semantic labelling (Section III-A). Now we use it for detection, to serve as a (single-stage) baseline. We use the same configuration as in [32], but extending the search space to cover position, scale, and orientation. This detector has shown significantly better results than the classic HOG + linear SVM approach (on pedestrian [31], [32] and face detection [42]). This baseline is a reference point for the performance of a strong generic object detector.

### D. Baseline: Two-Stage Detection

In addition to the single-stage detection baselines, we consider four methods that operate on semantic labelling results (first stage), to produce tool detections (second stage). We first describe a naïve two-stage baseline approach, named `Skeleton`. Then we present multiple variants of our proposed pipeline, each one using a different combination of semantic labelling technique and shape-model creation approach.

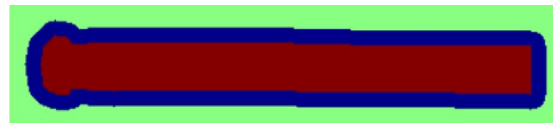


Fig. 9. Fixed shape template illustration for a suction tube. Red pixels are associated with a weight of 1, blue pixels with a weight of $-1$ and green pixels with a weight of 0.

TABLE II
FEATURE CHANNELS IMPACT ON SEMANTIC LABELLING ACCURACY

| Feature channels | Accuracy |
|---|---|
| `HOG` alone | 78.9% |
| `HOG+LUV` | 84.9% |
| `HOG+LUV+XY` | 85.1% |
| `HOG+LUV+XY+FB` | 85.2% |
| **`HOG+CN+XY+FB`** | **85.8%** |
| `Darwin` baseline | 73.4% |
| All background | 50% |

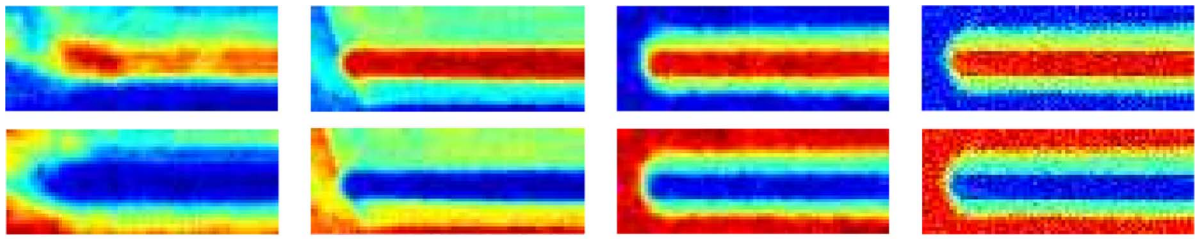
TABLE III
CLASSIFIER PARAMETERS IMPACT ON SEMANTIC LABELLING ACCURACY

| Weak classifiers | Model window size | Accuracy |
|---|---|---|
| 200 | $51 \times 51$ | 85.6% |
| 500 | $31 \times 31$ | 85.4% |
| **500** | $\mathbf{41 \times 41}$ | **85.8%** |
| 500 | $51 \times 51$ | 85.7% |
| 500 | $61 \times 61$ | 84.7% |
| 750 | $51 \times 51$ | 85.4% |
| 1000 | $51 \times 51$ | 85.4% |

`Skeleton` is a naive two-stage approach, performing classic morphological operations on top of the `SquaresChnFtrs` semantic labels. This hand-crafted method exploits the geometry of surgical instruments by searching exclusively for tubular shapes. Fig. 8(a) illustrates semantic labelling results (obtained as described in Section III-A), used as input of this method. To reduce labelling noise we apply a double morphological dilation on the input mask, using a structuring element of size $5 \times 5$ (Fig. 8(b)). Tubular shapes can be reduced to only their barycentre lines (or "skeletons") to be identified and counted, thus we extract topological skeletons [43] to summarise the tool presence evidence (Fig. 8(c)).

Assuming a minimal size for surgical instruments in the images, an additional noise reduction step is performed. After computing connected components, only skeletons with a size larger than an empirical threshold are kept (Fig. 8(d), one colour per connected component). These components are then used to estimate straight lines via Hough transform (Fig. 8(e)). Each line from the Hough transform, longer than a specific threshold, is considered as a candidate detection and enriched with a bounding polygon and a score computed proportionally to the line length. Finally, a greedy non-maximum suppression iteration, as presented earlier in the paper, is performed based on their scores for a final set of detections presented in Fig. 8(f).

`FixedTemplate` uses a linear classifier model, but instead of learning the weights it uses a hand-crafted template. Using the idealised shape of the surgical instrument of interest, a template of $125 \times 300$ pixels is created. Pixels inside the shape have a weight of 1, the ones around the shape boundaries a weight of



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

Fig. 10. Suction tube SVM models for each type of positive images with the spatial regularisation, and without for the last case. Tool class response map in the top row and background class response map in the bottom row. (a) Semantic labelling data. (b) All instruments annotations. (c) Single instrument annotations. (d) Single instruments annotations (no regularisation).

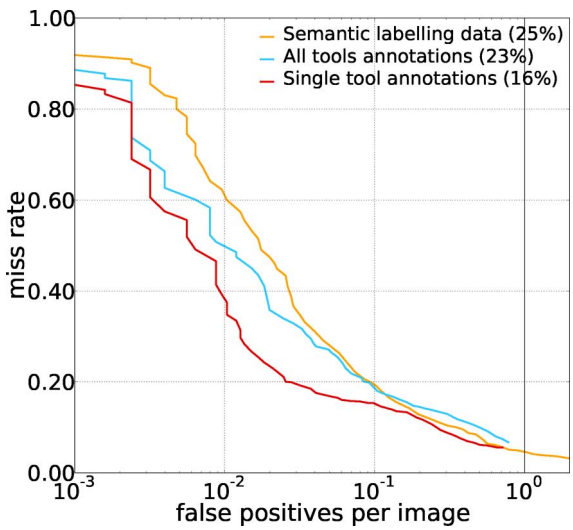



Fig. 11. Detection results for the suction tube (using the polygon overlap metric) vary depending on the type of positive examples used to learn the SVM model. The log-average miss-rate (LAMR) is reported in brackets.

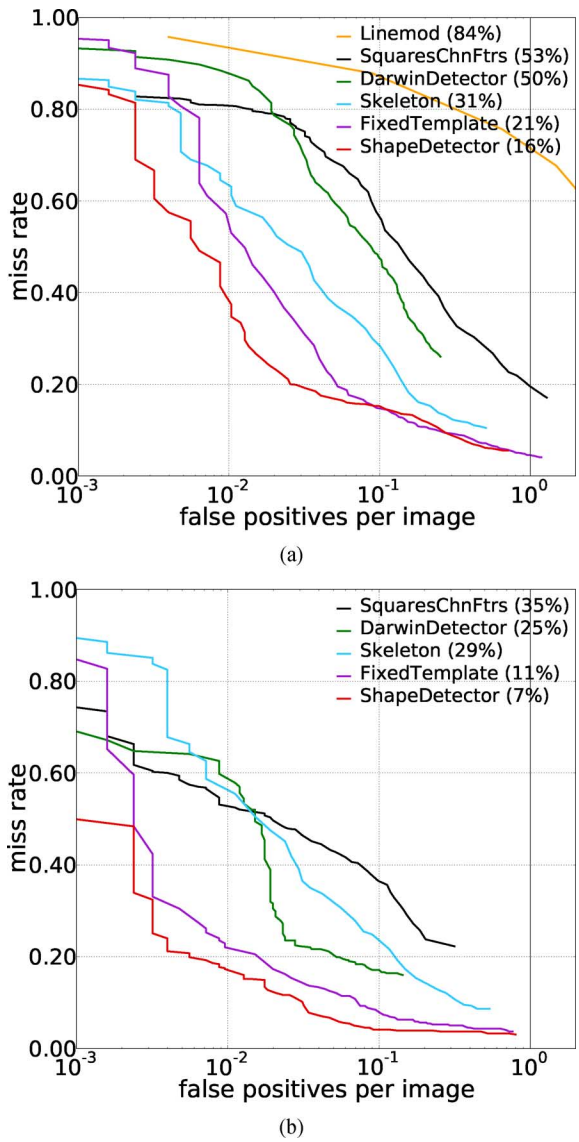



Fig. 12. Detection results over the NeuroSurgicalTools dataset, using the polygon overlap metric. Please refer to section IV for details on the evaluation procedure and compared approaches. The log-average miss-rate (LAMR) is reported in brackets. (a) Suction tube detection performance. (b) Bipolar forceps (upper part) detection performance.

$-1$, and the rest a weight of 0 (see Fig. 9). The linear classifier is applied over the `SquaresChnFtrs` semantic labels.

`DarwinDetector` operates identically to `FixedTemplate`, but detections are obtained on top of the `Darwin` [34] semantic labels instead of the ones obtained from `SquaresChnFtrs` (see Section V-A). This baseline allows us to compare the quality of our semantic labels against an alternative.

`ShapeDetector` is the name of our proposed method, described in Section III, which uses a linear SVM to do detection on top of the `SquaresChnFtrs` semantic scores.

## V. Results

Our approach has been implemented in C++, using CUDA libraries to perform parallel computing. Results were obtained on a DELL Precision T8600, Intel Xeon E5-2620 v2 @2.10 GHz (CPU), NVIDIA Titan Black (GPU). At test time, detectors were evaluated using a 4-pixel stride in both spatial dimensions, and a 5° orientation step (i.e., 72 orientations are evaluated). On a $612 \times 460$ pixel image, between 80 ms and 100 ms are necessary for feature extraction and pixel-wise classification (i.e., first stage), while the pose estimation (i.e., second stage) is performed in around 80–90 ms. The overall system runs at about a speed of 5 Hz, while a complete training requires approximately 2 hours. To ensure a fair comparison, we match the parameters

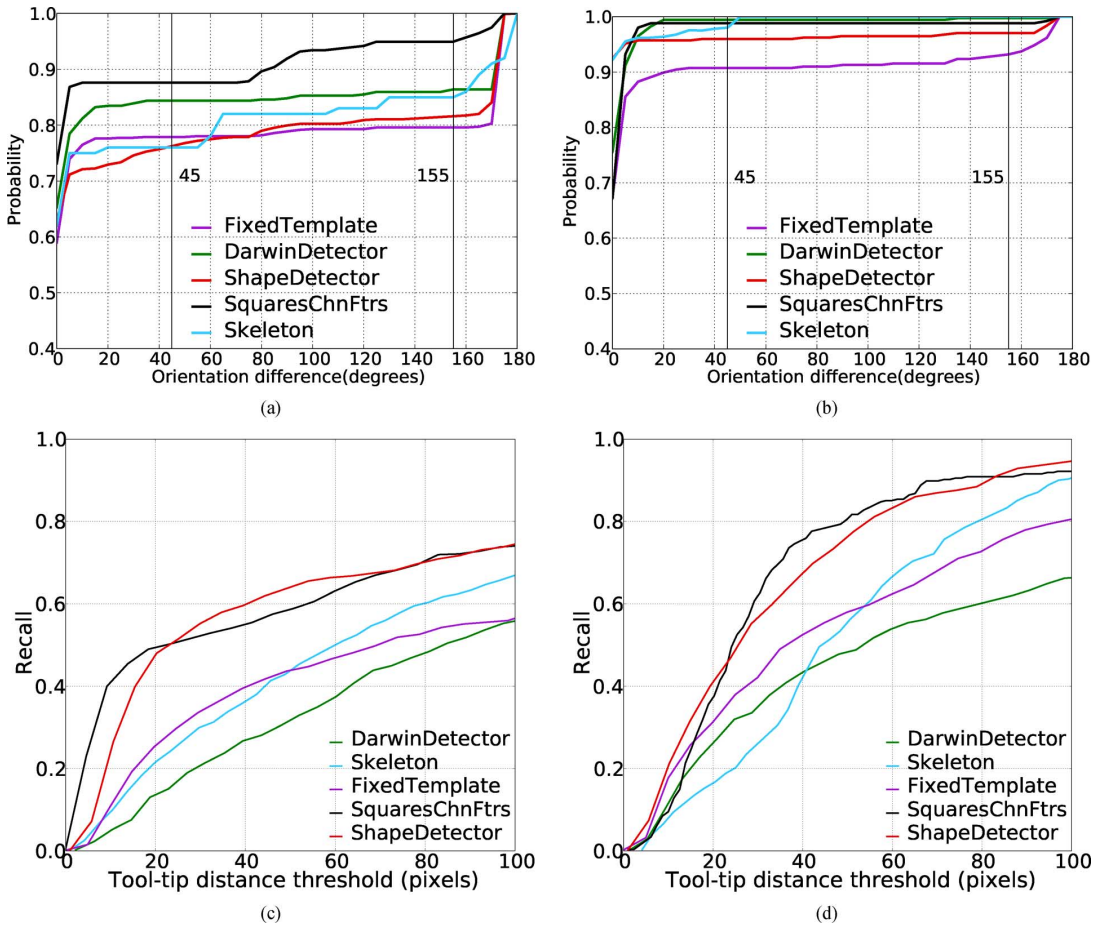



Fig. 13. Orientation and tip position tool pose parameters evaluation at the $10^{-1}$ FPPI rate. Top row corresponds to a comparison based on the in-plane orientation difference metric. Bottom row represents a comparison based on the tool-tip distance metric. (a) Orientation difference metric (Suction tube). (b) Orientation difference metric (Bipolar forceps). (c) Tool-tip distance metric (Suction tube). (d) Tool-tip distance metric (Bipolar forceps).

of each method as closely as possible (i.e., training data, evaluated scales, and orientations).

Section V-A presents the semantic labelling results (input for `ShapeDetector`), Section V-B analyses the design space for the `ShapeDetector` SVM training, and finally Section V-C presents and compares the detection results of the different methods.

### A. Semantic Labelling Results

For completeness, we also include the trivial classifier that considers every pixel as background.

Table II reports the impact of different feature channels on the labelling accuracy. It shows that the proposed `SquaresChnFtrs` meaningfully improves over our strong baseline. As expected, colour (`LUV` channels, `CN` colour names [33]) and texture are strong cues (`FB` filter bank [28], [34]), while position (`XY`) is rather weak. Figs. 5 and 15 provide examples of obtained labellings.

For the `HOG+CN+XY+FB` features combination, Table III reports the impact of the classifier parameters. It indicates that larger model window size or increased number of weak classifiers has very little to no effect on the semantic labelling accuracy. The decision trees depth parameter is not studied as it can not be modified.

All subsequent experiments using the `SquaresChnFtrs` semantic labelling are performed using 500 depth-2 decision trees, a $41 \times 41$ pixel model window, and `HOG+CN+XY+FB` as feature channels.

### B. SVM Model Training

This section illustrates the impact of various design choices and SVM model creation parameters. An accurate surgical instrument model is crucial for high detector performances.

`ShapeDetector` performances are reported in Fig. 11 for the three positive samples alternatives considered (see Section III-C and Fig. 7): (1) `SquaresChnFtrs` semantic labelling scores, (2) annotations of all surgical instruments, (3) annotations of a single surgical instrument.

Neither the value of the regularisation parameter $c$ nor the use of regularisation with a 2d spatial smoothness prior improve the overall quality of the detections, however the learned model is noticeably smoother (see Fig. 10).

All following experiments with SVM models are performed using a $c$ value of 1, the spatial regularisation term, a binary distribution for sampling negative examples, and single instrument annotations as positive examples.

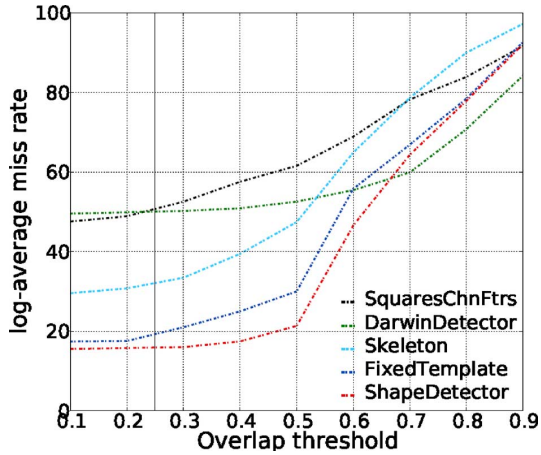



Fig. 14. Log-average miss rate as a function of the overlap threshold for the suction tube (polygons overlap metric).

### C. Detection Results

In the following, pose estimation results are reported for the "suction tube" tool (the most common one in the dataset), and for the "bipolar forceps" tool (the second most common).

Fig. 12 reports global tool position results based on the polygon overlap evaluation metric. Large differences in detection quality amongst the methods are visible. `Linemod` performs quite poorly in this domain, showing that using an off-the-shelf detector is not enough. `SquaresChnFtrs` performs significantly better, most likely due to its more flexible model. Still, generic detectors achieve a rather poor performance, reaching less than 50% recall at $10^{-1}$ false positives per image (for suction tube). On the other hand, the hand-crafted `Skeleton` approach provides better results, indicating that pixel-wise segmentation is a strong cue. Finally, `ShapeDetector`, our two-stage approach, obtains the best results thanks to its data driven learning, instead of hand-crafting features or shape cues. On this metric, at $10^{-1}$ false positives per image, the miss-rate is reduced by a third with respect to the best generic detector.

The poor result of `DarwinDetector` compared to `FixedTemplate` indicates that high quality semantic labels are key for good detection. The good results of our `ShapeDetector` show the utility of the proposed two-stage approach.

For candidate detections obtained by each method, the pose estimation quality is further assessed using orientation and tip position parameters, obtained at a fixed rate of $10^{-1}$ false positives per image (see Fig. 13).

As highlighted in Fig. 13(a), all the compared approaches exhibit a similar behaviour regarding orientation accuracy for suction tube detection. Given models being tested with a 5° orientation step, the best estimation with less than a 5° difference is achieved 70% of the time with our `ShapeDetector`. For the `ShapeDetector`, less than 20% of detections have an orientation deviating by 170°-180°, indicating a well placed detection regarding its global position, only facing the opposite direction. Noisy semantic labelling results around the tool-tip region, heavily focused by the shape model learning strategy, as long as occlusions can induce such a shift in orientation. Regarding the bipolar forceps surgical instrument (illustrated Fig. 13(b)), such a confusion in orientation is far less important, happening only for 5% of obtained detections with the `ShapeDetector`.

At a similar miss-rate of 15% at $10^{-1}$ false positives per image, `FixedTemplate` outperforms `ShapeDetector` at suction tube orientation estimation by a small margin. A 5% gap between the two methods is visible for a difference in orientation lower than 20° . Often the semantic labelling quality is quite noisy and highly irregular along the tool, with background pixels being misclassified as tool pixels more often when closer to the tool-tip region. When combined with high tilt values and partly occluded tips, the hand-crafted shape weights used for `FixedTemplate` are sometimes able to compensate for such adverse conditions. A better orientation estimation is then achieved at the cost of a shift alongside the tool shaft thanks to stronger weight values within the tip region. The same conclusion does not hold for the bipolar forceps, which indicates that small surgical tools, such as the suction tube, are especially susceptible to these problems because of their thinner tip region (i.e., modelled region).

Fig. 13(c) and (d) show results using the tool-tip distance metric (Section IV-A). Both `SquaresChnFtrs` and `ShapeDetector` have similar performances under this metric, with less than a 20-pixel error for a 50% recall (for the suction tube). Between `FixedTemplate` and DarwinDetector, the 10% recall difference for a 40-pixel suction tube tip error indicates the impact of the semantic labelling quality around tool boundaries. A 20% recall improvement for a 20-pixel tip error can be noted between the `FixedTemplate` and the `ShapeDetector` when using a suction tube model, pointing out the benefits from using sophisticated shape modelling towards the tool-tip estimation. With our proposed `ShapeDetector`, the bipolar forceps tip position is overall better estimated than for the suction tube. At a 60-pixel tip error, a recall of 83% is obtained for the bipolar forceps, whereas a recall of 67% only is achieved for the suction tube. Aside from tool tip occlusion, semantic labelling noise appears to be less influential for tools with a large enough tip region, the bipolar forceps being bigger than the suction tube for a similar microscope zoom value.

Fig. 14 shows the effect of selecting different overlap thresholds for the evaluation. The results obtained at the selected value of 25% are similar to the ones at the classic 50%. We can also notice that our proposed `ShapeDetector` obtains low log-average miss-rate for a large range of overlap thresholds.

In the following pages, visual results are displayed, starting with Fig. 15 showing semantic labelling results obtained with the `SquaresChnFtrs` where tool pixels are marked in green. Then, Fig. 16 illustrates detection success and failure modes obtained with our proposed `ShapeDetector` using a suction tube shape model. Finally, Fig. 17 presents side-by-side detection results obtained with the different tool detectors employed.

## VI. Discussion

### A. Two-Stage Approach

Our proposed two-stage approach reaches top performance, however success or failure cases depend critically on the (first) semantic labelling stage. As illustrated by the results between the `DarwinDetector` and the `ShapeDetector`, where only the

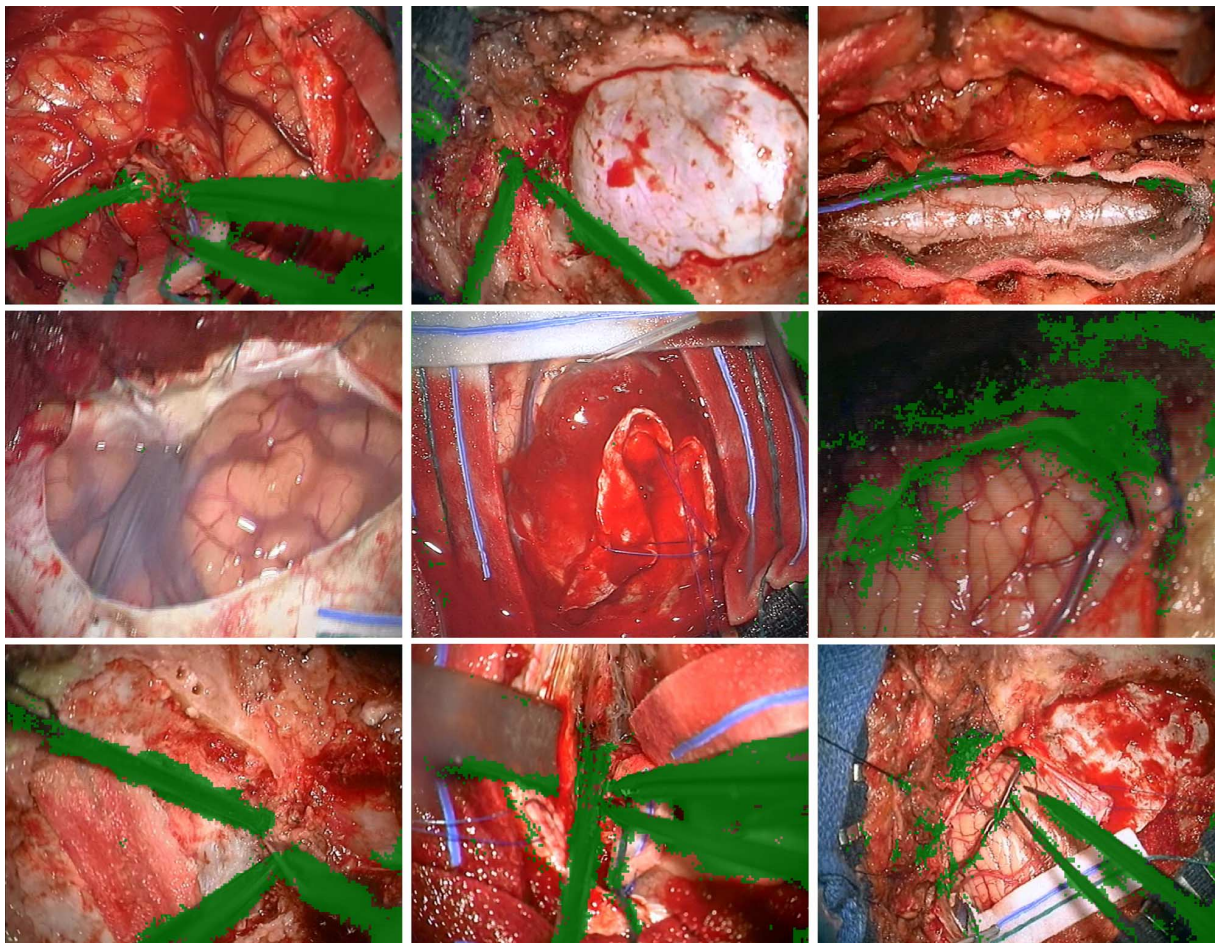


Fig. 15. Semantic labelling results obtained from our method (`HOG` + `CN` + `XY` + `FB` configuration). Detected tool pixels are marked in green. The rightmost column shows some failure modes.

first stage differs, improving the semantic labelling stage drastically improves overall results.

While using the `SquaresChnFtrs` method to perform the semantic labelling task greatly improves over existing baselines, the resulting maps are still noisy. The body of a surgical tool is mainly well labelled whilst its edges and tool tip are not. Such labelling errors lead to detection positioning errors, as reflected in the tool-tip distance metric experiments (Fig. 13(c) and (d)). The polygon overlap metric focuses more on the overall position and orientation of the tool (not only the tip), and seems less sensitive to such noise.

We observe that the semantic labelling struggles with very tilted surgical instruments, making tool-ends go out of focus, hence inducing a lot of blur in the image. In those cases, it happens that only 40% of a surgical instrument is correctly labelled, making the shape template matching harder and more likely to fail. Cases which involve noisy semantic labelling maps, tools with high tilt values, and partly or almost fully occluded tip regions have yet to be addressed properly. By improving the semantic labelling results, or through the use of a tracking layer, we expect tool-tip positions and orientations to be estimated more accurately. The third and fourth columns of Fig. 16 show some additional failure cases.

Our second layer currently assumes that the object shape changes through rotation and scaling only. The remaining degrees of freedom are expected to be handled by the learned SVM template. Finding a way to handle tools that have articulated elements remains to be explored in future work.

### *B. SVM Model*

Even with hundreds of training samples, learning an accurate tool-specific shape template through SVM training might be difficult. With our current implementation choices, enforcing 2d spatial smoothness in the SVM regularisation term has shown to be ineffective to induce any noticeable improvements in the detector performance. However, the resulting SVM models tend to be visibly smoother indicating a proper behaviour of the regularisation term. It might be that the piecewise approximation of shape templates used to gain computational speed (see Section III-B2) already enforces such spatial smoothness (in a brute-force manner).

In our current setup, SVM models are not meant to learn how to differentiate shapes of two similar surgical instruments. As a result, the detection score over a suction tube obtained with a suction tube SVM model can be hardly inferior to the one obtained with a bipolar forceps SVM model. Performing tool

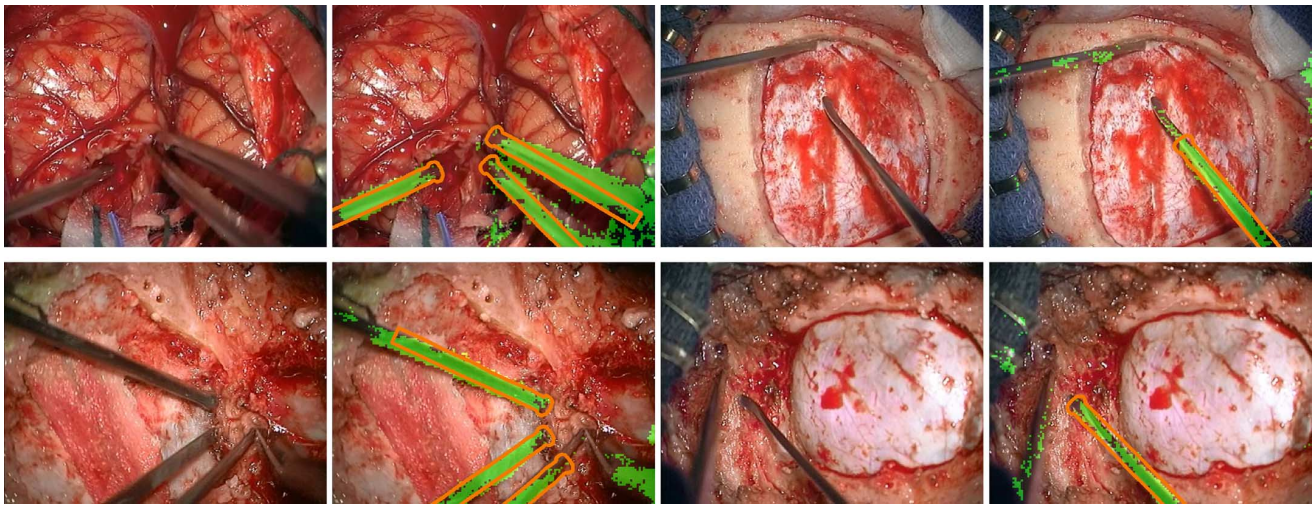


Fig. 16. Success and failure cases using the `ShapeDetector` approach with a suction tube model. Odd columns show original images, even columns show detection results.

classification together with detection is not straightforward in the current architecture. Our initial experiments indicate that only subtle cues enable to distinguish amongst tools (e.g., hook versus suction tube), and thus we believe that more discriminative features are needed to solve this fine-grained classification task.

### C. Evaluation Protocols

Assessing the performances of an object detection approach can be hard as relevant evaluation protocols have to be defined and corresponding evaluation metrics have to be used. Usually, evaluation protocols are built in order to identify strengths and weaknesses of an algorithm designed for a specific application. In this paper, we aim to develop a method with as few assumptions as possible and thus we choose to use standard computer vision metrics for evaluation, also with limited assumptions.

The first metric used, intersection over union criterion, is state-of-the-art and widely used for object detection in computer vision for overall good positioning. Developed to be used with bounding boxes, we consider the intersection over union criterion to also fit well with bounding polygons with an adaptation regarding the overlap area threshold. Instead of a 50% overlap threshold traditionally used, we decrease it to 25% because of the nature of the elongated polygons. Small variations in orientation can substantially lower the overlap area, and the point of this metric is to assess of accurate location not correct pose estimation. In retrospect, the traditional threshold could have been used since we observed performances stability until a 60% area threshold (see Fig. 14).

The second and third metrics used, are relatively straightforward methods used to evaluate accuracy in the pose estimation of the object (i.e., correct orientation and tip position). It has been previously used in similar works when using tracking approaches [19] and in body pose estimation evaluations (e.g., [44]).

We did not evaluate our method within a precise medical application, where potentially specific conditions could be used to optimize models and search ranges, thus obtaining better detection results. Using standard metrics and evaluation protocols, we already show better performances than baseline methods, which supports the idea that our approach will provide high quality in diverse applications.

### D. Applications

Many solutions investigated to solve the surgical instrument pose estimation problem require significant changes to operating rooms setup. Instead of relying exclusively on 2d video signals (as presented here), some methods require additional tags (e.g., RFID technology [45]). Such a technology is in the early stages of use in hospitals, very few are equipped due to installation costs and perceived return over investment. Moreover, studies are not in agreement with each other regarding the threat assessment of this technology on the patient and on other devices of an operating room [46], [47].

Only requiring the video feed from a surgical microscope, which is a standard medical equipment for most hospitals throughout the world, our proposed approach can directly be used in existing operating rooms. Currently running at $\sim 5$ Hz, our method is close to fast enough, and will reach frame-rate processing (25 Hz) in only a couple of hardware generations, or after speed-tuning the implementation (e.g., to use GPUs like in [35]).

## VII. Conclusion

Surgical instrument detection and pose estimation are key components for the next generation of context-aware computer-assisted surgical systems as for many medical applications such as surgical video indexation or surgeons' technique comparison. In order to preserve current operating room setup, we focus our work on 2d videos from existing surgical microscopes instead of using additional sensors. In this paper, we propose a new approach for surgical tool detection in 2d images that makes no assumptions on the number of tools, their shape or position in the image. The first stage of the approach performs a pixel-wise semantic labelling while the second stage matches global shapes. Evaluated on our newly introduced in-vivo dataset, this combined approach achieves better detection results than state-of-the-art baselines.

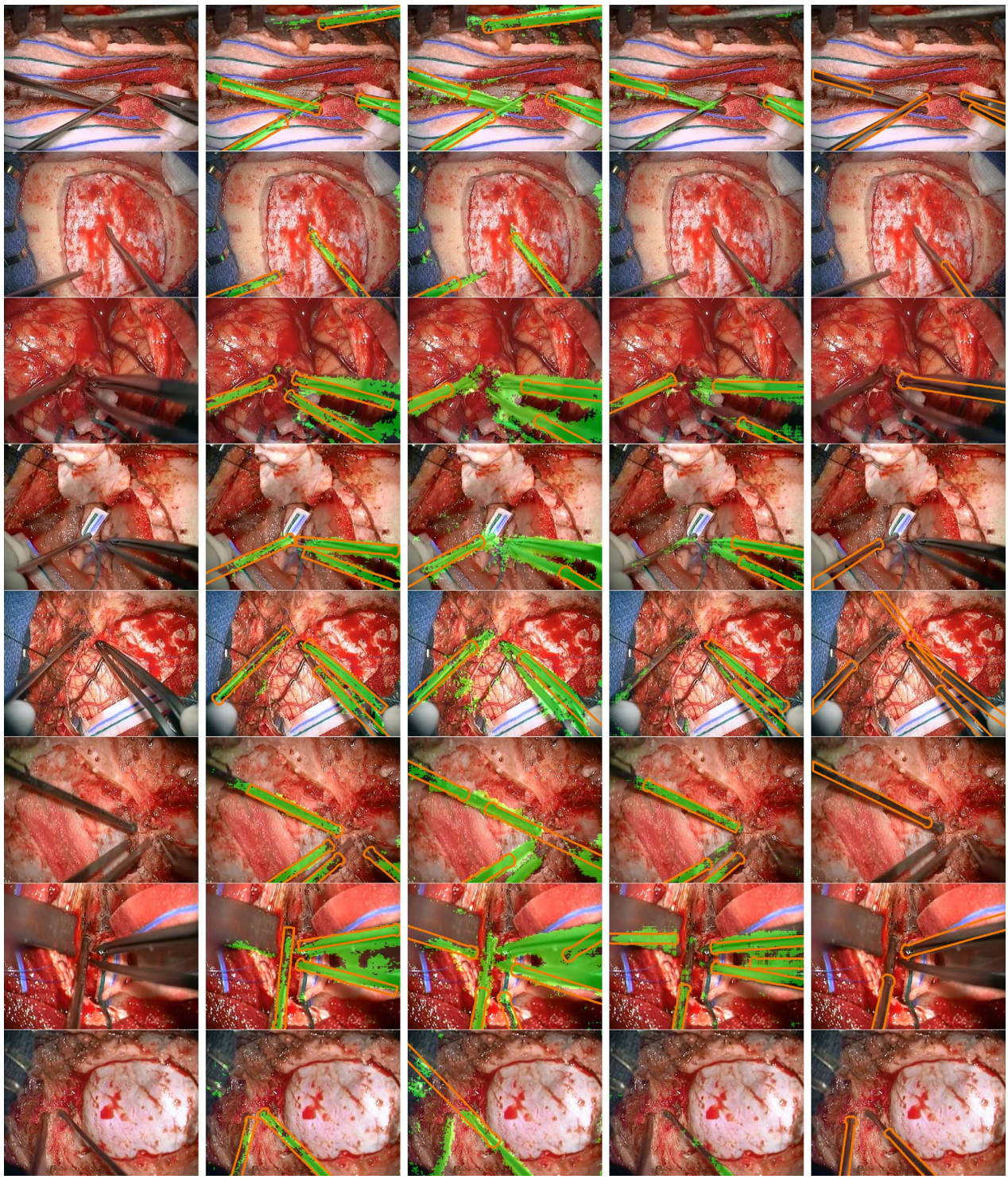


Fig. 17. Detection examples using a suction tube model (with semantic labelling results overlaid in green when used). From left to right: original image, `ShapeDetector`, `Skeleton`, `DarwinDetector`, `SquareChnFtrs`.

Future work will focus on improving the quality of the semantic labelling stage as long as the detection quality. We will also explore coupling detection with classification to be able to distinguish between different surgical instruments. Finally, we plan to improve our dataset by adding more surgical tools' classes and by increasing the diversity within each one.

## REFERENCES

[1] L. T. Kohn *et al., To Err Is Human:: Building a Safer Health System*. Washington, DC: Nat. Acad. Press, 2000, vol. 627.

[2] K. Cleary, H. Y. Chung, and S. K. Mun, "Or2020 workshop overview: Operating room of the future," in *Int. Congr. Ser.*, 2004, vol. 1268, pp. 847–852.

[3] F. Lalys and P. Jannin, "Surgical process modelling: A review," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 9, no. 3, pp. 495–511, 2014.

[4] F. Lalys, D. Bouget, L. Riffaud, and P. Jannin, "Automatic knowledge-based recognition of low-level tasks in ophthalmological procedures," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 8, no. 1, pp. 39–49, 2013.

[5] A. Casals, J. Amat, and E. Laporte, "Automatic guidance of an assistant robot in laparoscopic surgery," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1996, vol. 1, pp. 895–900.

[6] O. Tonet, R. U. Thoranaghatte, G. Megali, and P. Dario, "Tracking endoscopic instruments without a localizer: A shape-analysis-based approach," *Comput. Aid. Surg.*, vol. 12, no. 1, pp. 35–42, 2007.

[7] A. Krupa *et al.*, "Autonomous 3-d positioning of surgical instruments in robotized laparoscopic surgery using visual servoing," *IEEE Trans. Robot. Automat.*, vol. 19, no. 5, pp. 842–853, Oct. 2003.

[8] F. Miyawaki *et al.*, "Development of automatic acquisition system of surgical-instrument informantion in endoscopic and laparoscopic surgey," in *Proc. 4th IEEE Conf. Indust. Electron. Appl.*, 2009, pp. 3058–3063.

[9] S. Speidel *et al.*, "Visual tracking of Da Vinci instruments for laparoscopic surgery," *SPIE Med. Imag.*, pp. 903 608–903 608, 2014.

[10] D. Burschka *et al.*, "Navigating inner space: 3-d assistance for minimally invasive surgery," *Robot. Auton. Syst.*, vol. 52, no. 1, pp. 5–26, 2005.

[11] A. Reiter, P. K. Allen, and T. Zhao, "Feature classification for tracking articulated surgical tools," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012*, 2012, pp. 592–600.

[12] R. Wolf, J. Duchateau, P. Cinquin, and S. Voros, "3d tracking of laparoscopic instruments using statistical and geometric modeling," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2011*, 2011, pp. 203–210.

[13] S. Speidel, M. Delles, C. Gutt, and R. Dillmann, "Tracking of instruments in minimally invasive surgery for surgical skill analysis," in *Medical Imaging and Augmented Reality*. New York: Springer, 2006, pp. 148–155.

[14] R. Richa, M. Balicki, E. Meisner, R. Sznitman, R. Taylor, and G. Hager, "Visual tracking of surgical tools for proximity detection in retinal surgery," in *Inf. Process. Comput.-Assist. Intervent.*, 2011, pp. 55–66.

[15] A. Reiter and P. K. Allen, "An online learning approach to in-vivo tracking using synergistic features," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 3441–3446.

[16] S. Kumar *et al.*, "Video-based framework for safer and smarter computer aided surgery," in *Hamlyn Symp. Med. Robot.*, 2013, pp. 107–108.

[17] S. Voros, J.-A. Long, and P. Cinquin, "Automatic detection of instruments in laparoscopic images: A first step towards high-level command of robotic endoscopic holders," *Int. J. Robot. Res.*, vol. 26, no. 11–12, pp. 1173–1190, 2007.

[18] S. Haase, J. Wasza, T. Kilgus, and J. Hornegger, "Laparoscopic instrument localization using a 3-d time-of-flight/RGB endoscope," in *Proc. IEEE Workshop Appl. Comput. Vis.*, 2013, pp. 449–454.

[19] R. Sznitman *et al.*, "Data-driven visual tracking in retinal microsurgery," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012*, 2012, pp. 568–575.

[20] K. Ali, F. Fleuret, D. Hasler, and P. Fua, "A real-time deformable detector," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 2, pp. 225–239, 2012.

[21] R. Sznitman, R. Richa, R. H. Taylor, B. Jedynak, and G. D. Hager, "Unified detection and tracking of instruments during retinal microsurgery," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 5, pp. 1263–1273, May 2013.

[22] Z. Pezzementi, S. Voros, and G. D. Hager, "Articulated object tracking by rendering consistent appearance parts," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2009, pp. 3940–3947.

[23] M. Allan *et al.*, "Toward detection and localization of instruments in minimally invasive surgery," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 4, pp. 1050–1058, Apr. 2013.

[24] R. Sznitman, C. Becker, and P. Fua, "Fast part-based classification for instrument detection in minimally invasive surgery," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2014*, 2014, pp. 692–699.

[25] S. McKenna, H. N. Charif, and T. Frank, "Towards video understanding of laparoscopic surgery: Instrument tracking," in *Proc. Image Vis. Comput.*, New Zealand, 2005.

[26] S. Speidel *et al.*, "Recognition of risk situations based on endoscopic instrument tracking and knowledge based situation modeling," in *Proc. SPIE Med. Imag.*, 2008, pp. 69 180X–69 180X.

[27] A. Reiter, P. K. Allen, and T. Zhao, "Marker-less articulated surgical tool detection," *Proc. Comput. Assist. Radiol. Surg.*, vol. 7, pp. 175–176, 2012.

[28] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation," in *Computer Vision–ECCV 2006*. New York: Springer, 2006, pp. 1–15.

[29] S. Giannarou, M. Visentini-Scarzanella, and G.-Z. Yang, "Probabilistic tracking of affine-invariant anisotropic regions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 130–143, Jan. 2013.

[30] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: A database and web-based tool for image annotation," *Int. J. Comput. Vis.*, vol. 77, no. 1–3, pp. 157–173, 2008.

[31] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," in *Proc. Br. Mach. Vis. Conf.*, 2009, pp. 91.1–91.11.

[32] R. Benenson, M. Mathias, T. Tuytelaars, and L. Van Gool, "Seeking the strongest rigid detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 3666–3673.

[33] F. Shahbaz Khan *et al.*, "Color attributes for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 3306–3313.

[34] S. Gould, "Darwin: A framework for Machine learning and computer vision research and development," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 3533–3537, 2012.

[35] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool, "Pedestrian detection at 100 frames per second," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2903–2910.

[36] A. Ess, T. Müller, H. Grabner, and L. Van Gool, "Segmentation-based urban traffic scene understanding," in *Proc. Br. Mach. Vis. Conf.*, 2009, pp. 84.1–84.11.

[37] C. J. Burges, "A tutorial on Support Vector machines for pattern recognition," *Data Mining Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, 1998.

[38] A. Lehmann, B. Leibe, and L. Van Gool, "Fast prism: Branch and bound Hough transform for object class detection," *Int. J. Comput. Vis.*, vol. 94, no. 2, pp. 175–197, 2011.

[39] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 743–761, Apr. 2012.

[40] M. Everingham *et al.*, "The pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, pp. 1–39, 2014.

[41] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab, "Dominant orientation templates for real-time detection of texture-less objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2257–2264.

[42] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool, "Face detection without bells and whistles," in *Comput. Vis.*, 2014, pp. 720–735.

[43] D. Zhao and D. G. Daut, "Morphological hit-or-miss transformation for shape recognition," *J. Vis. Commun. Image Represent.*, vol. 2, no. 3, pp. 230–243, 1991.

[44] M. Dantone, J. Gall, C. Leistner, and L. van Gool, "Body parts dependent joint regressors for human pose estimation in still images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2131–2143, Nov. 2014.

[45] J. E. Bardram *et al.*, "Phase recognition during surgical procedures using embedded and body-worn sensors," in *Proc. IEEE Internat. Conf. Pervasive Comput. Commun.*, 2011, pp. 45–53.

[46] R. Van Der Togt *et al.*, "Electromagnetic interference from radio frequency identification inducing potentially hazardous incidents in critical care medical equipment," *JAMA*, vol. 299, no. 24, pp. 2884–2890, 2008.

[47] B. Christe *et al.*, "Testing potential interference with RFID usage in the patient care environment," *Biomed. Instrum. Technol.*, vol. 42, no. 6, pp. 479–484, 2008.