Raport Badawczy
Research Report

RB/23/2017

Instrument detection and pose estimation with rigid part mixtures model in video-assisted surgeries

D. Węsierski, A. Jezierska

Instytut Badań Systemowych
Polska Akademia Nauk

Systems Research Institute
Polish Academy of Sciences

POLSKA AKADEMIA NAUK

Instytut Badań Systemowych

ul. Newelska 6

01-447 Warszawa

tel.: (+48) (22) 3810100

fax: (+48) (22) 3810105

Kierownik Zakładu zgłaszający pracę:
Prof. dr hab. inż. Antoni Żochowski

Warszawa 2017

Instrument detection and pose estimation with rigid part mixtures model in video-assisted surgeries

Daniel Wesierski[a,b,*], Anna Jezierska[b,a]

[a] *Multimedia Systems Department, Faculty of Electronics, Telecommunications, and Informatics, Gdansk University of Technology, ul. Narutowicza 11/12, 80-233 Gdansk, Poland*

[b] *Systems Research Institute of the Polish Academy of Sciences, ul. Newelska 6, 01-447 Warsaw, Poland*

Abstract

Localizing instrument parts in video-assisted surgeries is an attractive and open computer vision problem. A working algorithm would immediately find applications in computer-aided interventions in the operating theater. Knowing the location of tool parts could help virtually augment visual faculty of surgeons, assess skills of novice surgeons, and increase autonomy of surgical robots. A surgical tool varies in appearance due to articulation, viewpoint changes, and noise. We introduce a new method for detection and pose estimation of multiple non-rigid and robotic tools in surgical videos. The method uses a rigidly structured, bipartite model of end-effector and shaft parts that consistently encode diverse, pose-specific appearance mixtures of the tool. This rigid part mixtures model then jointly explains the evolving tool structure by switching between mixture components. Rigidly capturing end-effector appearance allows explicit transfer of keypoint meta-data of the detected components for full 2D pose estimation. The detector can as well delineate precise skeleton of the end-effector by transferring additional keypoints. To this end, we propose effective procedure for learning such rigid mixtures from videos and for pooling the modeled shaft part that undergoes frequent truncation at the border of the imaged scene. Notably, extensive diagnostic experiments inform that feature regularization is a key to

*Corresponding author

Email address: daniel.wesierski@pg.gda.pl (Daniel Wesierski)

fine-tune the model in the presence of inherent appearance bias in videos. Experiments further illustrate that estimation of end-effector pose improves upon including the shaft part in the model. We then evaluate our approach on publicly available datasets of *in-vivo* sequences of non-rigid tools and demonstrate state-of-the-art results.

Keywords: surgical instrument detection, surgical instrument tracking, video-assisted minimally invasive surgery, robotic surgery, part-based models

1. Introduction

Facilitating video-assisted surgeries belongs to main objectives for developing next-generation operating theaters. During the surgery, a surgeon controls surgical instruments either robotically or manually. Minimally invasive surgeries and microsurgeries involve vision sensors that help surgeons correctly position the instruments onto operated tissue areas. Carrying out the surgeries is not easy though. In minimally invasive surgeries the surgeons insert elongated surgical instruments through keyhole incisions in the body thereby compromising the dexterity of maneuvers within the body. On the other hand, delicate, retinal microsurgery requires high precision in placing the instruments over retina after eye surface penetration. Furthermore, the surgeons struggle to perceive depth well and lack tactile feedback using the available surgical vision technology. Augmenting the surgeon's vision with helpful yet unobstructive metadata and robotized support thus appear as attractive, potential improvements to existing surgical workflow.

In view of this, locating surgical instruments with the help of vision sensors has recently received much attention Bouget et al. (2017). Knowing the location and pose of articulated surgical tools in videos could enable virtual measurements and overlays Reiter et al. (2012a), Kumar et al. (2014) for better guidance and recognition of risk situations Speidel et al. (2008), detailed motion analysis for surgical skills assessment Chmarra et al. (2007), Speidel et al. (2006), Oropesa et al. (2013), Ahmidi et al. (2017), and surgical process modeling Lalys

et al. (2013) for better understanding of surgical workflow. Furthermore, precisely locating tool pose is essential for retinal microsurgery Balicki et al. (2009), Rieke et al. (2016). In further stages of maturity, the technology could enable visual servoing Lee et al. (1994), Voros et al. (2006) for automatically maintaining the operated tissue areas in view. Suturing tasks Nageotte et al. (2004), Padoy & Hager (2012), Sen et al. (2016) could be executed with greater ease and effectiveness by automating the gripping process of surgical suture and needle. Likewise, even grater autonomy of surgical robots has already been speculated Yang et al. (2017).

Instrument localization has been approached with the help of other sensors as well. Robotic manipulators can control the instruments with high flexibility and stability. Their encoders accumulate errors in forward kinematics, though, leading to inaccurate estimations of the absolute instrument location Reiter et al. (2013). Other external tracking systems suffer from lower accuracy as well and require extensive hardware integration Allan et al. (2014) thereby cumbersomely integrating to multiple operating rooms. While depth-only sensing devices would be hardly interpretable for humans, combined RGB-D sensors Haase et al. (2013) provide an interesting alternative. However, stereovision remains the primary modality in video-assisted surgeries Maier-Hein et al. (2014). By transmitting 3D-hallucinated videos, widespread color cameras offer an intuitive, visual feedback to surgeons. Amenable to easy transfer between operating rooms and motivated by steady progress of computer vision, vision-based instrument tracking thus constitutes an encouraging approach to improving the guidance and navigation of manual and robotic surgeries. As 2D pose enables 3D pose estimation from stereoscopic images **?**, this work addresses the problem of tool 2D pose estimation in a single image frame.

Ideally, one would like to have an algorithm for instrument pose estimation that generalizes to all kinds of tool shapes at test time. A human visual cortex effortlessly and rapidly localizes key parts of an instrument in video sequences even when it sees particular tool type for the very first time. In current practice though, when could we say that a tool part detector indeed generalizes shapes

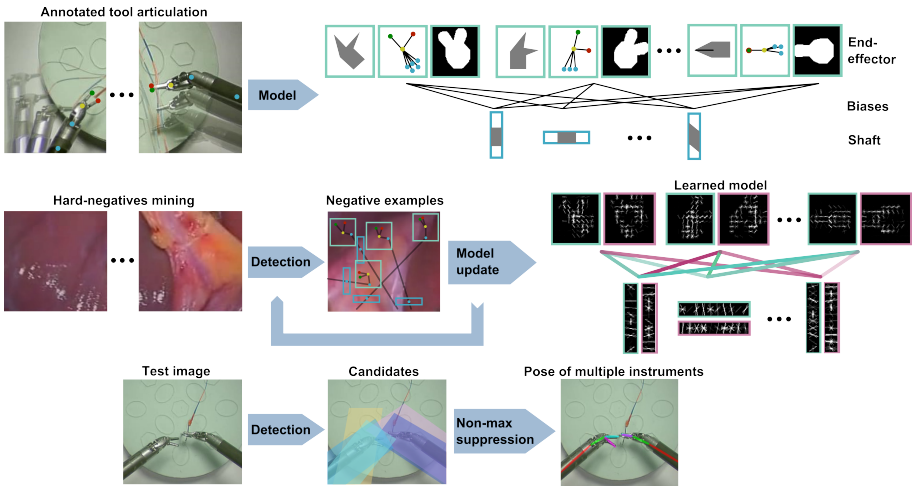

Figure 1: Flowchart of our method. **Top:** We capture diverse poses of surgical instruments with a rigid part mixtures model. The model uses (i) a set of appearance templates (i.e., part mixture) that represent various articulations of the end-effector part (e.g., rotated, open, closed gripper), (ii) a set of appearance templates that represent a single sub-part of the shaft at multiple orientations, and (iii) a set of biases, sandwiched between template parts, that promote or discourage rigidly composed appearance from mixture components of the parts. Each type of end-effector articulation is associated with pose-specific keypoints to retrieve instrument pose as well as binary regularization masks that facilitate learning from annotated videos. **Midlle:** We hard-mine negative examples on arbitrary images without the instruments and iteratively refine the model. **Bottom:** At test time, we match the model to an image. Candidate detections are non-maximum suppressed. Well separated, consistent compositions of end-effector and shaft type, which best explain the image, translate to particular poses of articulated instruments.

well enough to be commissioned in surgical intervention? Undoubtedly, building a visual intelligence system with such a generalization power is currently challenging. Surgical instruments are rigid, non-rigid, and robotic, with a variety of shapes and functions. This might suggest progressive transfer of the surgical vision technology to operating theaters in order to continuously deliver best outcome to inpatients. Given that we are at the beginning of the process, this work proposes an approach that strongly relies on prior knowledge. Recognition of previously seen tool poses is central to our method

1.1. Contributions

We describe a rigid part mixtures model of a surgical instrument along with dedicated model matching and learning procedures for articulated pose estimation. We evaluate the training-time configurability of our approach on localizing keypoints of non-rigid and robotic tool parts at test time. The proposed model is a spatial assembly of instrument parts that encode mixtures of tailored pose appearances.

Our contribution is three-fold. Firstly, we develop a structured, springs-free model of non-rigid and robotic instruments (Fig. 1). An initial version of the parts-based model, which originally appeared in Wesierski et al. (2015), did not generalize to articulated, robotic tools. Models with deforming springs Yang & Ramanan (2013) can synthesize appearance changes that result from previously unseen deformations of the underlying object structure by flexing the model to regions with putative objects. Such models are useful especially when structural deformations of an object follow long-tail distribution, such as in human pose estimation. However, pose variation of a tool is evidently more constrained. We propose, therefore, to represent the appearance changes of the instrument explicitly with rich pose-dedicated appearance mixtures of object parts. Our approach imposes a rigid structure on spatially distributed local features of the tool shaft. Hence, it can discard putative tool regions near the oriented shaft that might prompt a model with springs to flexed structure detection. It is tempting to build-in rotational invariance at the local level of model parts either through features Liu et al. (2014) or non-linear classifiers Ramakrishna et al. (2014) in order to reduce model complexity. On the other hand, one would like to help the model enforce repetitive orientation patterns along the shaft, and in general, grant consistent appearance compositions of end-effector and shaft parts. Notably, one would not like to accumulate implausible, zigzag evidence along the shaft. To this end, we represent object articulations through rigidly interrelated part mixtures. In effect, our white-box representation explicitly explains structured image evidence.

Secondly, we demonstrate that a structured part-based model can be suc-

cessfully applied to detection and pose estimation of surgical instruments. Our method is on par with or exceeds state-of-the-art results in localizing tool parts on publicly available datasets of in-vivo sequences. We surpass our preliminary work Wesierski et al. (2015) in the task of tool localization mainly because of improved learning and detection procedures that exploit feature regularization and local shaft pooling, respectively. Estimating instrument pose is typically approached in a disjoint manner by first detecting individual parts and then fusing detections . By exploiting rigidly structured relations between instrument parts, our method detects the end-effector and shaft parts jointly thereby recovering the full 2D pose of the instrument. Applying a structured model, though, is challenging as this requires frequent updates of its underlying structure. Object appearance can vary significantly between frames, especially due to its frequent truncations. Specifically, the rigid, straightly elongated shaft has often been used as a discriminative visual cue in detecting the tool and in estimating its 3D pose Doignon et al. (2006),Wolf et al. (2011). However, observing that surgeons often prefer to work in close proximity to tissue, Reiter et al. (2012b) ignore the shaft and focus on tracking the articulating end-effector with thousands of efficiently matched templates. This leads to a dilemma. On the one hand, one would like to take advantage of the shaft part when it is visible. On the other hand, one has to take into consideration the varying, truncated tool structure. Our detector exploits the rigid shaft while adapting to its changing length by pooling locally selected sub-parts of the shaft. To our knowledge, we are the first to evaluate the impact of various downsampling procedures that account for shafts of different lengths at test time.

Thirdly, our structured, white-box representation of a tool admits comprehensive evaluation on four datasets. Notably, the empirical evidence shows that including the shaft part into the model helps estimate articulation of the end-effector part. As the shaft part improves localization of end-effector parts, we derive a new metric in the tool pose estimation setting for this part that jointly evaluates the shaft location and orientation. Moreover, with proper training, we show on phantom video sequences of robotic instruments that our model

can encode a large number of rigid, pose-dedicated appearances and explain well diverse, end-effector articulations using only hundreds of training images. While object detectors typically are trained on datasets of temporarily unrelated images, in surgical tool localization settings object detectors and trackers usually are trained on videos. We show that pose-specific feature regularization is a key to learning effective tool models from videos. To our knowledge, we are the first to quantitatively evaluate a model-based approach in the task of pose estimation of non-rigid and multiple robotic tools.

2. Related work

Visual data, which are registered during the surgery, may suffer from degraded quality due to motion blur, gas inflating the abdomen, and smoke Vogt et al. (2003). Clarity of vision can also be compromised by adverse lighting conditions in the form of globally varying illumination of the scene, reflections on the tool and tissue regions Saint-Pierre et al. (2011), as well as shadows left by the tool. In addition, surgical forceps can leave little image evidence over low-contrasted and cluttered backgrounds. Requiring the algorithms to recognize the appearance of the grippers in this setting makes instrument detection and pose estimation a challenging task in practice.

In recent history, markers have been used to simplify detection of the instruments in surgical videos West & Maurer Jr (2004). Several marker design techniques have been proposed, e.g. blue and green colour tags Wei et al. (1997), Tonet et al. (2007), Groeger et al. (2008) for estimation of tool location, several stripes along the tool for achieving robustness against occluded markers and for measuring tool-to-camera distance Casals et al. (1996), Zhang & Payandeh (2002), light-emitting diodes on tool tips, accompanied by a laser that projects patterns for measuring tool-to-organ distance Krupa et al. (2003), and fiducial markers Zhao et al. (2010). However, marker-based methods have practical limitations Mckenna et al. (2005). As the main disadvantage of marker-based approaches remains the invasiveness into surgical workflow Bouget et al. (2015),

we propose a marker-less detection method.

Marker-free analysis of videos in retinal microsurgery and minimally invasive surgery has been carried out by leveraging a combination of tool location and motion priors as well as appearance and shape priors. The priors are either specified manually or learned from data and help segment the pose in 2D-t and 3D-t spaces. Although retrospective tracking such as batchwise processing of video frames is justified in off-line surgical process analysis, other applications require a tracker to process video frames ideally in ultra real-time.

Visual tracking algorithms take advantage of temporal evolution of object pose and appearance. Constrained search space allows increasing computational efficiency Lee & Soatto (2011), favoring neighbor locations or appearances, and learning object appearance on-line Henriques et al. (2015). In Burschka et al. (2005), a Kalman filter selected the most likely template to match to current frame from predetermined mixture of out-of-plane orientation-encoded appearance templates. Particle filter sampled multiple, coarse shape templates in Mckenna et al. (2005) that were fit to binary images. Learning tool appearance online from an initial track was realized as spatio-temporal cooperation of image features in Reiter & Allen (2010). Bootstrapping object appearance from the initial frame has recently been applied to tracking instrument center with state-of-the-art performance Li et al. (2014a). Precise, gradient-based tracking in Richa et al. (2011) optimized mutual information objective function for tool-tissue proximity detection. Product-of-Tracking-Experts Kumar et al. (2013) algorithm merged outputs of multiple trackers under a probabilistic framework. The trackers used DPM detectors Felzenszwalb et al. (2010a) that were trained on particular end-effector shapes. Tool tips detector trained with regression forests in Rieke et al. (2016) was merged with a tracker to find tool location and articulation during retinal microsurgery.

External and prior information also facilitates the task of estimating tool location and pose. In Reiter et al. (2012b), da Vinci robot encoders allowed generating only a subset of proximal end-effector CAD models of articulated visual templates. The encoders also served as good initialization in search for

tool pose in Reiter et al. (2013). Locating tool insertion point in 3D granted near real-time visual servoing in Voros et al. (2006) and real-time 3D pose estimation in Wolf et al. (2011) by restricting the search for tubular shaped tool model around the insertion point. In Sznitman et al. (2011), pose search commenced at the image border for efficiency. Also, matching of rigid contour template to tool end started at 2D pose that was specified by encoder readings in Staub et al. (2010).

Our method can take advantage of ad hoc, external information and temporal tracking procedures. Motion constraints and robotic encoder readings could help select at any time a specific, small subset of our huge set of pose-dedicated appearance templates. Complementary to such auxiliary heuristics, an object detector remains an inevitable component of pose tracking algorithms. To spawn a new track, a tracker can either be initialized manually by a user Reiter & Allen (2010), or automatically by a pretrained Reiter et al. (2012a) or coarse Ramanan et al. (2007) detector. However, when the target reappears after occlusion or after leaving the scene, the detector can reinitialize the old track. A tracker itself can be fused with the detector for object localization Sznitman et al. (2013) and identification Andriluka et al. (2008).

Hence, in this work, we focus on developing a detector and, opportunistically, track the target tool structure within a tracking-by-detection framework. While motion models filter instrument location and size, our model-based method can successfully detect instrument pose in each video frame independently from neighbor frames. The proposed model is a spatial assembly of instrument parts that encode mixtures of tailored pose appearances. The model can explain instrument pose variations by switching between mixture components that explicitly transfer pose-related keypoints. By expressing various tool poses with corresponding appearances of an object part, our approach relates to López-Sastre et al. (2011). The method reasons about holistic object poses by conditioning DPM root part mixtures on specific camera viewpoint categories. In contrast, we find fine-grained object pose in the form a skeleton by conditioning geometric compositions of part mixtures on object articulations. By transferring metadata

over detected object locations, our method also shares similarities with Hejrati & Ramanan (2014) that uses a synthesis engine over provided 3D object models. It explicitly estimates 3D shape of rigid objects in a reconstructive, bottom-up manner from object keypoints. Our approach explicitly transfers the shape of the articulated objects, as illustrated on the task of 2D shape estimation of non-rigid tools in retinal microsurgery setting.

Pipeline algorithms for tool detection and tracking have traditionally been initialized with early-decision, pixel-in–pixel-out classifiers. The background is mostly reddish and whitish and the tool has greyish color. One of the earliest works Lee et al. (1994), Uecker et al. (1995) learned a simple bayesian classifier to discern tool from organ pixels by capturing `RGB` color statistics of training data under a parametric model. Color distributions of both classes overlap to some extent. Though tools are not reddish, organs are often locally greyish. Hence, the authors apply median filtering and spatial relaxation within each frame and temporal coherence constraints between neighbor frames to post-process erroneous classifications. Finally, they find segments in binary images with tool-like shape that satisfy predefined image moments. Non-parametric classification of pixels was realized in Doignon et al. (2005) by thresholding an image histogram of non-linearly transformed `RGB` color channels. Segmentation masks were obtained in Speidel et al. (2006) by region growing to allow fitting tool models using image moments. Bayes classifier assigned binary labels to `HS` images followed by a Condensation tracker of weighted tool pixel locations.

Descriptors of image regions play significant role in tool detection setting. Past work has explored region descriptors in region-in–pixel-out classification systems where each pixel gains contextual support of center surrounding region. Such classifiers can make more aware class predictions than pixel-level classifiers in the presence of local, noisy observations, e.g. blur and spots of reflected light. In Sznitman et al. (2012), AdaBoost binary classifier of sums of oriented edges scored every pixel in a patch. The patch was indicated by a gradient-based tracker. A gaussian kernel, anchored at the tracked tool center, weighted and aggregated the classifier scores. In another work Sznitman et al. (2014),

multiclass gradient boosted regression trees learned to classify oriented edge features of the tool. Assigned labels of tool parts to each pixel created semantic clouds that then allowed retrieving tool center location and orientation using Ransac algorithm in the second stage. In Bouget et al. (2015), binary Adaboost classifier was trained with channel features that combined HOG gradients, LUV color attributes, filter banks, and spatial position to create foreground clouds in the first stage of the test pipeline. In the second stage, a linear SVM classifier learned tool shape templates over the clouds at various orientations in the spirit of TextonBoost features. The algorithm reported tip location and orientation of the tool. In Allan et al. (2013), color histograms in multiple color spaces together with SIFT and HOG features were evaluated using random forests for binary pixel classification. The most discriminative color channels were hue, saturation, opponent 2 and opponent 3. The 2D shape of the obtained connected regions was represented by the moment of inertia tensor for initialization of segmentation and 3D pose (location+orientation) recovery within level-set framework.

Tool articulation was estimated in Reiter et al. (2013) with the help of region covariance descriptor. The descriptor represented multiple landmark features on the da Vinci end-effector. Randomized trees classifier assigned landmark labels to pixels for robust fusion of shaft and end-effector parts with the help of Ransac sampling and robot kinematics. In Kumar et al. (2015), gaussian process regression of HOG and LBP features estimated 3D angular pose of tools. The conducted evaluation suggested equal performance of both features. In Rieke et al. (2016), first a tracker detected a non-rigid tool, and then random forest regressed a combination of HOG and color features of tool center and tips within the tracked window.

Multiscale image representations have recently been built with deep learning network architectures, in which object part localization is cast as a 2D heatmap prediction task. Heatmaps of tool keypoints were obtained in Kurmann et al. (2017) using a CNN-based U-net deep learning network architecture under a probabilistic scene model for tool pose estimation. Similarily, a deep residual network from Laina et al. (2017) was trained for joint segmentation and lo-

calization of tools parts under keypoint heatmap regression. Optic flow and color images jointly fed a CNN in Sarikaya et al. (2017) to obtain tool region proposals. The algorithm retrieved the 2D center location of robotic tools in challenging phantom data.

Our detection-based method uses reduced HOG descriptors, being insensitive to specific image contrast. The algorithm exploits no temporal constraints. It achieves state-of-the-art performance on public datasets by training structured tool models with pose-specific regularization and by jointly detecting end-effector and shaft keypoints. It outputs full 2D pose, i.e. the location of tool tips, tool center, shaft ending, and shaft orientation, of non-rigid and robotic instruments. Using additional annotations, it further outputs precise skeleton of tool parts.

3. Problem formulation

The structure of surgical instruments, e.g. for laparoscopy and retinal microsurgery, can operationally be represented in image I as a composition of two parts: (i) a rigid, straight, elongated shaft $\mathcal{S}$ and (ii) a non-rigid or robotic end-effector $\mathcal{E}$, as depicted in Fig. 1. In practice, both parts slightly rotate during a surgery while instrument pose admits non-circumvolving motion. In general, though, the shaft is oriented at an arbitrary angle as the locations of the incisions vary between surgical scenarios. Moreover, the grippers of the end-effector articulate and take various forms, i.e. the length and shape of the grippers varies. In view of this, we approach the problem of surgical instrument detection and pose estimation by capturing the appearance variation of the tool with a structured model of rigid mixtures of parts that jointly encodes pose-specific tool appearance.

In this section we describe our model. We will commence by introducing the terminology and setup that will guide us through the rest of the paper.

3.1. Rigid Part Mixtures Model

Let G_I denote a two-dimensional regular tessellation of the pixel grid of image I. Let $l \in \mathbb{N}^{2\times 1}$ denote discrete 2D locations in image domain L over the whole grid G_I, and $L_b \subset L$ denote a discrete set of locations on arbitrarily shaped (e.g., rectangular, circular) one-dimensional border stripe of this grid, denoting tool entry points in the image.

The end-effector part $\mathcal{E}$ is enclosed in a single window in the grid with the center location $l_{\mathcal{E}} \in L \setminus L_b$. It has a set of anchors at which the shaft can be attached, where each anchor $l_{\mathcal{A}} \in L \setminus L_b$ denotes an offset location from $l_{\mathcal{E}}$. The shaft part $\mathcal{S}$ is a collection of N_e sub-parts that are outlined by adjacent windows. We restrict possible locations of these windows $l_{\mathcal{S}(k)} \in L$, where $1 \leqslant k \leqslant N_e$, to an oriented raster line segment that ranges from the border stripe $l_{\mathcal{S}(1)} \in L_b$ to the shaft ending $l_{\mathcal{A}}$, as shown in Fig. 2(middle).

Then, let $l_{\mathcal{AS}} = (l_{\mathcal{A}}, l_{\mathcal{S}(1)})_{2\times 2}$ denote the line segment. As the length of the shaft varies in a video sequence, we downsample the number of shaft subparts by representing the $\mathcal{S}$-part as a subcollection of $N \leqslant N_e$ sub-parts for each new image frame I. In this way, the accumulated evidence for this part is qualitatively comparable among hypothesized shaft locations in the 2D-t space. As a result, in our model the location of the $\mathcal{S}$-part $l_{\mathcal{S}}(l_{\mathcal{AS}}) = \left(l_{\mathcal{S}(k_1)}, l_{\mathcal{S}(k_2)}, \ldots, l_{\mathcal{S}(k_{N-1})}, l_{\mathcal{S}(k_N)}\right)_{2\times N}$ determines some ordering of these sub-parts on the line segment $l_{\mathcal{AS}}$ according to the strength of the evidence. We then restrict this ordering by dividing the line segment into subsegments to require coherent appearance along the whole shaft at test time, such that for example $l_{\mathcal{S}(1)} \leq l_{\mathcal{S}(k_2)} < l_{\mathcal{S}(k_1)} < \ldots < l_{\mathcal{S}(k_{N-1})} < l_{\mathcal{S}(k_N)} \leq l_{\mathcal{A}}$ where $l_{\mathcal{S}(k_1)}$ and $l_{\mathcal{S}(k_2)}$ belong to the first and $l_{\mathcal{S}(k_{N-1})}$ and $l_{\mathcal{S}(k_N)}$ belong to the last subsegment.

We represent the appearance and structure of the instruments under a bipartite graph $\mathcal{M} = \{V, E\}$. The appearance mixtures of the end-effector part are chained with the appearance mixtures of the shaft parts (Fig. 1). The nodes V of the graph $\mathcal{M}$:

$$V = \{w^i_{\mathcal{E}}, l^i_{\mathcal{A}}, l_{\mathcal{E}}\}_{i=1}^{n_{\mathcal{E}}} \cup \{w^j_{\mathcal{S}}, l_{\mathcal{S}}\}_{j=1}^{n_{\mathcal{S}}} \tag{1}$$

denote particular appearances of the $n_{\mathcal{E}}$ end-effector and $n_{\mathcal{S}}$ shaft mixtures, respectively. The i-th component of the appearance mixture of the end-effector part at location $l_{\mathcal{E}}$ is specified by template $w^i_{\mathcal{E}}$ that rigidly encodes specific articulation of this part. It is equipped with a set of anchors $l^i_{\mathcal{A}}$ that link to the shaft part. The j-th component of the appearance mixture of the shaft part at location $l_{\mathcal{S}}$ is specified by template $w^j_{\mathcal{S}}$ that can capture specific perspective and orientation of the part, e.g. an outwards slanted shaft. The edges E of the graph $\mathcal{M}$:

$$E = \{b^{ij}_{\mathcal{ES}}\}^{n_{\mathcal{E}} \times n_{\mathcal{S}}}_{ij=1} \tag{2}$$

model rigid compositions of the end-effector mixture with the shaft mixture. Specifically, the co-occurrences $b^{ij}_{\mathcal{ES}} \in \mathbb{R}$ bias configurations of mixtures such that certain, rigidly encoded articulations $w^i_{\mathcal{E}}$ may form more consistent compositions with certain orientations $w^j_{\mathcal{S}}$. In effect, our model encodes rigid structure.

We define the mixture of the shaft part as orientation templates. On the other hand, the $\mathcal{S}$-part lies on the oriented line segment $l_{\mathcal{AS}}$. Hence, the mapping $j : l_{\mathcal{AS}} \rightarrow \{1, 2, ..., n_{\mathcal{S}}\}$ that depends on a given instance of this oriented line uniquely determines the j-th mixture component of the shaft. For notational convenience, we define the mixture label j that depends on $l_{\mathcal{AS}}$ as $j(l_{\mathcal{AS}}) = \bar{j}$.

The varying length of the shaft notwithstanding, our model allows taking advantage of the discriminative evidence for this part in each image during tracking-by-detection. As we assume the elongated shaft roughly admits consistent appearance along the image plane, we deem all sub-parts of its j-th mixture component to be alike and dedicate a single, canonical template $w^j_{\mathcal{S}(k_p)} = w^j_{\mathcal{S}(k_1)}$, $p = 1, \ldots, N$, for representing their appearance. In effect, replicating the canonical template for the N sub-parts as:

$$w^j_{\mathcal{S}} = \left[w^j_{\mathcal{S}(k_1)}, \ldots, w^j_{\mathcal{S}(k_1)}\right] \tag{3}$$

yields the representation of the template of the shaft part.

Then, instantiating the composition of a pair of particular mixture components of the $\mathcal{ES}$-parts in image I at location $l_{\mathcal{ES}} = (l_{\mathcal{E}}, l_{\mathcal{AS}})$ is scored with our

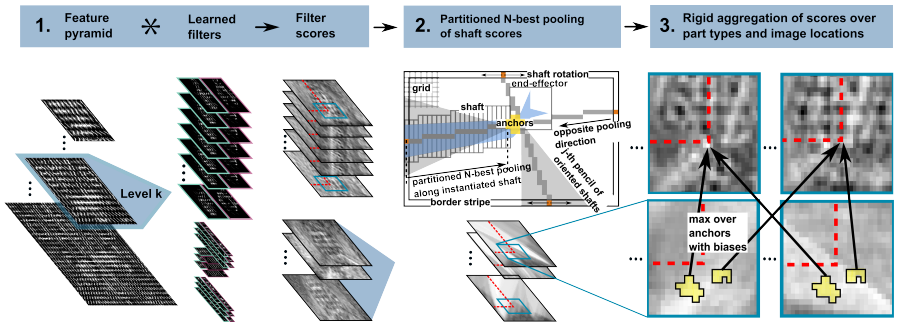

Figure 2: Instrument detection. Matching rigid part mixtures model to an image consists of three steps. **Step 1:** An image pyramid is converted into HOG pyramid. Learned appearance templates of the end-effector and shaft part are convolved with HOG image at given pyramid level yielding filter response maps. **Step 2:** N-best scores in each shaft response are locally pooled along instantiated line segments (Algorithm 1). We instantiate the segments at endpoints lying on the border stripe. Sweeping the endpoints through the whole border stripe generates all possible locations and orientations of the shaft part. **Step 3:** N-best pooled shaft scores are max-pooled over anchor locations and aggregated with end-effector score. This is repeated using dynamic programming across all shaft and end-effector mixture components for each location of the end-effector. Each composition comes with certain negative or positive bias.

model as:

$$S(I, l_{\mathcal{ES}}, i) = \langle w^i_{\mathcal{E}}, \phi^i_{\mathcal{E}}(I, l_{\mathcal{E}}) \rangle + \tag{4}$$
$$1/N \sum_{p=1}^{N} \langle w^{\bar{j}}_{\mathcal{S}(k_1)}, \phi^{\bar{j}}_{\mathcal{S}}(I, l_{\mathcal{S}(k_p)}(l^i_{\mathcal{AS}})) \rangle + b^{i\bar{j}}_{\mathcal{ES}}$$

where $\langle \cdot, \cdot \rangle$ is the inner product of vectorized arguments, $\phi^i_{\mathcal{E}}(I, l_{\mathcal{E}})$ and $\phi^{\bar{j}}_{\mathcal{S}}(I, l_{\mathcal{S}(k_p)}(l^i_{\mathcal{AS}}))$ are image descriptors (e.g., HOG Dalal & Triggs (2005a), color histogram). The function $\phi^i_{\mathcal{E}}(I, l_{\mathcal{E}})$ describes image region in the window of the i-th $\mathcal{E}$-part component at location $l_{\mathcal{E}}$. The function $\phi^{\bar{j}}_{\mathcal{S}}(I, l_{\mathcal{S}(k_p)}(l^i_{\mathcal{AS}}))$ describes image region in the window of the sub-part of the j-th $\mathcal{S}$-part component at location $l_{\mathcal{S}(k_p)}(l^i_{\mathcal{AS}})$. The location $l_{\mathcal{S}(k_p)}(l^i_{\mathcal{AS}})$ returns k_p-th element of $l_{\mathcal{S}}(l^i_{\mathcal{AS}})$ segment, anchored at one of the offsets from the anchor set $l^i_{\mathcal{A}}$ of the i-th $\mathcal{E}$-part component.

4. Method

In this section we describe the proposed method for detecting articulated surgical tools. We also describe a procedure for discriminative learning of dedicated tool models directly from annotated video sequences.

4.1. Detection

We find the rigid composition of mixture components of the $\mathcal{ES}$-parts at location $l_{\mathcal{ES}}$ that best explains image I by solving:

$$(l, i) = \text{argmax } S(I, l_{\mathcal{ES}}, i) \tag{5}$$

as depicted in Fig. 2. We solve (5) in three steps at each level of image pyramid in order to find correct scale of the instruments.

Step 1 Matching the appearance templates $\{w^i_{\mathcal{E}}\}^{n_{\mathcal{E}}}_{i=1}$ and $\{w^j_{\mathcal{S}}\}^{n_{\mathcal{S}}}_{j=1}$ to corresponding image descriptors at each location in L amounts to filter convolution in the feature space. The convolution results in $n_{\mathcal{E}}$ and $n_{\mathcal{S}}$ tables of appearance scores for each component of end-effector and shaft mixture, respectively.

Step 2 As the $\mathcal{S}$-part is represented by N sub-parts, the score of hypothesized shaft orientation depends on finding such a configuration of image descriptors that best match to $w^j_{\mathcal{S}}$ template. However, selecting N-best scoring sub-parts of the shaft from locations within a line segment without partitioning it into subsegments, as in Wesierski et al. (2015), would bias the detector to favor longer shaft segments, as we show in section 5.4. Hence, we propose a heuristic, partitioned N-best pooling procedure that selects N sub-part scores roughly equally from the instantiated segment. The procedure first partitions the segment into M subsegments with equal number of sub-parts and then selects N_*-best sub-part scores from each of M subsegments, where $N = MN_*$ and $N \geq N_*$.

Furthermore, there are many candidate shaft parts that lie on a line segment $l_{\mathcal{AS}}$. Unlike Wesierski et al. (2015), that select sub-part scores independently at each shaft orientation and location, we observe that the procedure can reuse

previously selected sub-part scores of the hypothesized, shorter shaft part to select the scores of the longer shaft part that both lie on the same line segment. With this in hand, the shaft score tables are pooled more efficiently.

The pseudocode of the procedure is given in Alg. (1). We generate line segments W at all possible orientations by sweeping their endpoints through border stripe L_b, hence $|W| = |L_b|^2$. Then, partitioned N-best pooling procedure traverses instantiated line segment W^p (Line 1), which determines shaft orientation label $\bar{j}$ (Line 2), to select N sub-part scores from array S^p of the $\bar{j}$-th table $I^{\bar{j}}$ (Line 3). Notably, as the number M of subsegments is constant, the length N_*^p of subsegments varies (Line 4) to total N scores at any given location q. To this end, at each location q in the segment W^p (Line 5), we update the number N_*^p of pooled local sub-part scores according to the number of already traversed subsegments (Line 6). Then, we efficiently retrieve N_*-best local scores from the first to the current subsegment at given location in the segment (Line 7) by maintaining an auxiliary array of sorted scores in each subsegment. Finally, the array Q of downsampled and summed scores B (Line 8) is stored in-place at $Y^{\bar{j}}$ (Line 12) together with the shaft beginning and ending $X^{\bar{j}}$ (Line 13) at respective locations $W^p(q)$ of the instantiated segment.

Step 3 Our graph $\mathcal{M}$ is a mixture of chains in which $\mathcal{E}$-part mixture components are parents and $\mathcal{S}$-part mixture components are children. We employ dynamic programming routine to exhaustively search over the state space (l, i). It combines $n_{\mathcal{E}}$ end-effector appearance scores (Step 1) with $n_{\mathcal{S}}$ pooled shaft sub-part appearance scores (Step 2) across their plausible locations and mixture components.

The search enumerates all possible compositions of mixture components of the $\mathcal{ES}$-parts. After aggregating the score $b_{\mathcal{ES}}^{ij}$ of ij-th composition with the N-best scores of the j-th component of shaft part mixture, the best segment of the shaft $l_{\mathcal{AS}}$ is selected across anchor set $l_{\mathcal{A}}^i$ for each i-th mixture component of the end-effector. We then retrieve the best i-th mixture component at location $l_{\mathcal{E}}$. Repeating this search procedure for each $l_{\mathcal{E}}$ amounts to $|L|$ end-effector candidates. We select the most plausible end-effector at $l_{\mathcal{E}}$ that has the highest

score (4), then backtrack to the best i-th component stored at that location, and terminate at the best $l_{\mathcal{AS}}$ pointed by this component.

Algorithm 1 Partitioned N-best pooling (Step 2)

Input:

N ▷ total number of best scores from all subsegments

N_* ▷ minimal number of best scores from one subsegment

I^j ▷ score tables of shaft mixture $j = 1, \ldots, n_{\mathcal{S}}$

W ▷ line segments with opposite endpoints (l_{b_1}, l_{b_2}) on L_b

Output:

Y^j ▷ N-best pooled score tables of shaft mixture

X^j ▷ endpoints of shaft segments

for $p = 1, \ldots, |W|$ **do**
  $\bar{j} \leftarrow$ shaft type based on orientation of p-th segment
  $S^p \leftarrow I^{\bar{j}}(W^p)$ ▷ array of shaft sub-part scores
  $N_*^p \leftarrow \emptyset$ ▷ array of N_* numbers for M subseg.
  for $q = N, \ldots, |W^p|$ **do**
    $N_*^p \leftarrow \texttt{update}(N_*^p, q)$
    $B \leftarrow \texttt{retrieveNbest}(S^p, N_*^p, q)$
    $Q \leftarrow \texttt{push}(\texttt{sum}(B))$
  end for
  for $q = N, \ldots, |W^p|$ **do**
    if $Q(q - N + 1) > Y^{\bar{j}}(W^p(q))$ **then**
      $Y^{\bar{j}}(W^p(q)) \leftarrow Q(q - N + 1)$
      $X^{\bar{j}}(W^p(q)) \leftarrow (W^p(q), l_{b_1}^p)$
    end if
  end for
end for

Non-maximum suppression When more than one tool is present in the image at test time, we use a greedy NMS method to select best, separated detection candidates above some scoring threshold. We then prune all candidates

that overlap with the current best candidate. We repeat this procedure until there are no candidates left.

Pose and skeleton transfer Surgical tools vary in shape. Capturing tool appearance with pose-specific templates allows augmenting the templates with additional annotations that correspond to pose-specific and shape-specific keypoints of the tool. At test time, our method can retrieve the actual tool pose and tool skeleton by transferring the assigned keypoints at detected locations, as illustrated in Fig. (**??**).

4.2. Learning

We learn the parameters of the model in a supervised manner (Fig. 3). Our model of surgical instruments uses a mixture of appearance templates per part, where only a single template of this part is present in a given positive training image. It is inspired by the deformable part models Felzenszwalb et al. (2010a); Yang & Ramanan (2013). Its array of model parameters is learned jointly and takes the form:

$$\begin{aligned}\beta = \Big[& b^{11}_{\mathcal{ES}}, \ldots, b^{ij}_{\mathcal{ES}}, \ldots, b^{n_\mathcal{E} n_\mathcal{S}}_{\mathcal{ES}}, \\ & w^1_\mathcal{E}, \ldots, w^i_\mathcal{E}, \ldots, w^{n_\mathcal{E}}_\mathcal{E}, \\ & w^1_\mathcal{S}, \ldots, w^j_\mathcal{S}, \ldots, w^{n_\mathcal{S}}_\mathcal{S} \Big]\end{aligned} \tag{6}$$

Since β uses a canonical appearance template $w^j_\mathcal{S}$ of a single sub-part to generalize the appearance of all shaft sub-parts for j-th mixture component, the function (4) scoring a training feature vector x_n yields the following dot-product form:

$$S(I_n, l_{\mathcal{ES}}, i) = \langle \beta, x_n \rangle \tag{7}$$

where

$$x_n = (0 \ldots 1 \ldots 0 \ldots \phi^i_\mathcal{E}(I_n, l_\mathcal{E}) \ldots 0 \ldots \phi^{\bar{j}}_\mathcal{S}(I_n, l_{\mathcal{S}(k)}) \ldots 0) \tag{8}$$

It induces a sparse structure on x_n that depends on particular pre-assignment of mixture labels to respective parts in a given training image I_n.

We learn model parameters β under linear SVM regime:

$$\begin{aligned} \text{argmin}_{\beta,\xi} \quad & \frac{1}{2}\|\beta R\|^2 + C^+ \sum_{n=1}^{m^+} \xi_n + C^- \sum_{n=1}^{m^-} \xi_n \\ \text{s.t.} \quad & \beta\ x_n^+ \geq 1 - \xi_n\ , \quad \forall x_n^+ \\ & \beta\ x_n^- \leq -1 + \xi_n\ , \quad \forall x_n^- \end{aligned} \tag{9}$$

where the regularization matrix R is diagonal. The elements of R are positive, such that $R_{ii} = \{1, \tau\}$ and $\tau < 1$ attenuates background features. The objective funtion (9) can be optimized with, e.g., a dual coordinate-decent solver Yang & Ramanan (2013). The above formulation states that our model β should learn to assign scores higher than 1 to positive examples x_n^+ of rigid compositions of respective mixture components and assign scores lower than -1 to negative examples x_n^-. The objective function penalizes violations of these constraints with slack variables $\xi_n \geq 0$, weighted by constants C^+ and C^-. The negative examples x_n^- come from incorrect detections. They are mined as hard-negatives on images without surgical instruments. An example illustration of learned parameters of a da Vinci tool model is in Fig. 4.

Mixture labels We assume that a given collection of positive training images contains only keypoint annotations. Hence, we have to retrieve the missing ij-labels of mixture components for each image. As shown in Fig. 3, we obtain the i-th label of the $\mathcal{E}$ part by first (i) binning the end-effector keypoints in a coarse, polar grid and then (ii) grouping the bins features into $n_\mathcal{E}$ disjoint sets across all training images. In effect, a given, unique spatial arrangement of bins captures a particular, i-th articulation of the end-effector. This allows us to estimate tool articulation at test time by prior averaging of all keypoints sharing the i-th label. The j-th label of the $\mathcal{S}$-part is obtained similarly, by slicing the image plane into $n_\mathcal{S}$ angular intervals. The angular resolution of the polar grids determines the maximal number of possible mixture components for both parts.

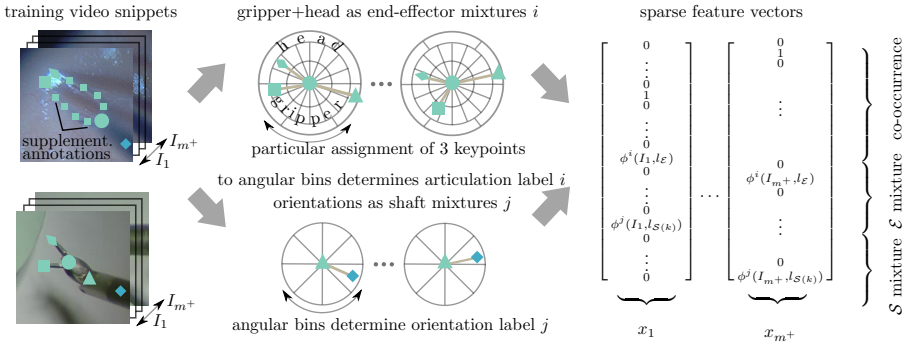

Figure 3: Learning mixture labels from annotated training videos. The mixture labels of shaft and end-effector parts determine sparse structure of feature vectors x_n in SVM classification. The annotations of positive training examples indicate keypoint locations of shaft beginning, shaft ending, two end-effector tips, and optionally of da Vinci head. For non-rigid tools, the shaft ending is the tool center $l_{\mathcal{A}}$ (top row, left). For robotic tools (bottom row, left), the intersection of forceps and head denotes the tool center. The number of mixture components of both parts, $n_{\mathcal{E}}$ and $n_{\mathcal{S}}$, is obtained automatically and depends on the resolution of respective coarse grids. We use polar grids to retrieve types of end-effector and shaft parts. Innermost circle of the end-effector part is the boundary of multiresolution models (Sec. 5.2). Finally, compositions of $\mathcal{ES}$ mixture components serve to store their co-occurrence indicator as well as their feature descriptors at respective locations of the sparse feature vectors. In addition, our model allows retrieving exact skeleton of the grippers at test time by providing it with supplementary annotations (top row, left). The skeleton, which is averaged over all keypoints with i-th label, is transferred onto detected end-effector type thereby yielding its precise shape in the image.

5. Experiments and results

In this section we extensively evaluate our method in the task of estimating articulated structure of a single and multiple surgical tools. To this end, firstly we use two public datasets to compare the performance of our and state-of-the-art approaches. Unlike most other approaches that are initialized manually and use temporal trackers to constrain the search space of putative tool locations, our method densely searches for tool center locations over the entire image, detects surgical tools in each frame individually, and uses no temporal constraints. We show that it still yields competitive results to the state-of-the-art. Secondly, we show that the proposed rigid parts mixture model is comfortably configurable

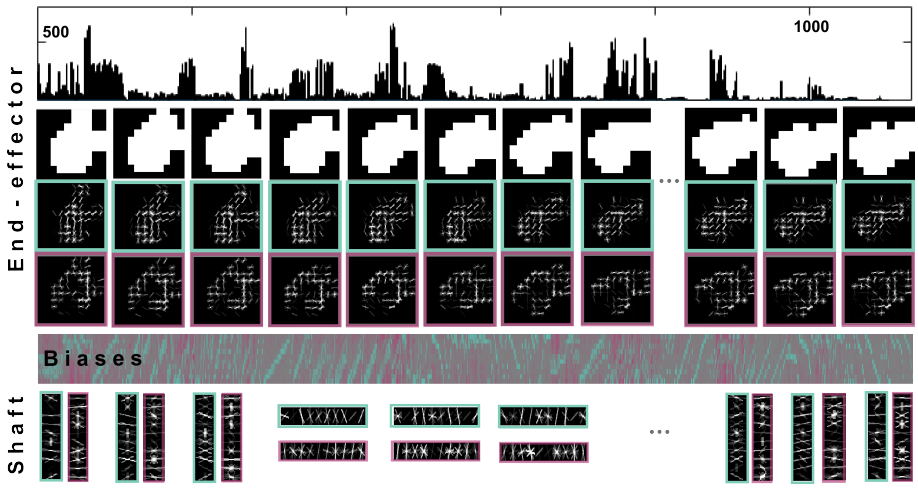

Figure 4: Example of learned end-effector, bias, and shaft parameters of robotic tool model. Top of the figure shows dispersity of the number of examples across end-effector types. Binary regularization masks, which are binary sums of dilated exemplar skeletons, facilitate learning of tailored appearance templates. The algorithm can then focus on discriminative areas of training images to fine-tune positive and negative HOG templates mainly along tool contours. Point clouds of annotated keypoints determine the articulated skeletons.

to more complex tools. It can detect tools of variable level of articulation, with the most prominent example of the da Vinci, robotic surgical instruments.

This section is organized as follows. First, we review two public datasets, describe an additional, phantom dataset of robotic instruments (sec. 5.1), and fourth, public dataset from which we collect images of one tool type. We give implementation details regarding our algorithm (sec. 5.2). Next, we provide a thorough review of metrics that allow evaluating the performance of the algorithms for detection and pose estimation of one and multiple surgical instruments (sec. 5.3). In addition, we propose an extended form of one metric to evaluate the performance of our algorithm in detecting the shaft part. Training and testing phases of the proposed model allow multiple design configurations but we select only particular ones for comparison with the state-of-the-art. To this end, we perform multiple diagnostic tests in sec. 5.4. Notably, we evaluate each training configuration with a fixed testing configuration. After having se-

lected the best performing training configuration, we evaluate multiple testing configurations. In each case, a particular model is trained and tested according to the dataset train and test splits, as shown in Tab. 1. Finally, we report qualitative and quantitative results of the selected configuration with respect to the state-of-the-art in sec. 5.5. Notably, by transferring metadata of the detected end-effector type, we demonstrate that our model can precisely estimate the skeleton of the end-effector. Furthermore, it can be applied to detect and estimate the pose of non-rigid and robotic surgical instruments. Finally, comparison of our method to state-of-the-art approaches shows that our method often improves upon them in terms of model versatility and precision.

5.1. Datasets

We use four datasets to evaluate our method and compare it to the state-of-the-art. Two datasets contain *in-vivo* videos, acquired during (i) retinal microsurgery and (ii) laparoscopy, the third dataset (iii) contains *phantom* videos, acquired during da Vinci suture manipulation, and the fourth one (iv) contains images of non-rigid scissor tools from cholecystectomy surgeries. The datasets are summarized in Tab. 1. Notably, the table presents train and test splits for training tool models in each dataset.

Retinal Microsurgery (`REMI`) The dataset `REMI` Sznitman et al. (2012) contains three video sequences of *in-vivo* vitreoretinal surgeries that were recorded through a microscope. Each sequence shows a single surgical tool, which is always present in the video. Annotations include four landmarks: shaft beginning, shaft ending, and both tool tips. The dataset features specular reflections, tool-like shadows, frequent low contrast, blur and shaft truncation, but little variation in scale of a tool. Moreover, the shaft beginning is located away from the image border. In the third sequence `REMI 3`, we skipped 16 images that were left unannotated by Sznitman et al. (2012) because of much blurred tool. In addition, we gathered further annotations for the training sets as metadata that represented precise skeleton of the forceps. Given the detected end-effector type at test time, our model allows transferring its skeleton's shape to each image,

Table 1: Summary of four datasets for our experiments, learned model configurations, and test configurations. The table presents train and test splits for each dataset sequence. We train on the first half and test on the second half of each sequence from REMI and LAPA datasets. For the PHDV dataset, we train on the whole first sequence and test on the other two sequences, as shown by the number of images that are attributed to the splits. For the CH80SCI dataset, the train images come from the first 40 videos, and the test images come from the remaining 40 videos. The datasets are augmented by image rotation within respective angular ranges and, additionally, the images from the PHDV dataset are flipped. The number of trained end-effector and shaft templates depend on the articulation of the tool. Our learning procedures indicate that the robotic tools require an order of magnitude more templates and model parameters than the non-rigid tools. Finally, NMS is required only for multiple tools in the PHDV dataset. For each tool model, the test times (in sec.) per image are indicated at the bottom of the table.

Video Sequence	REMI 1	REMI 2	REMI 3	LAPA	PHDV 1	PHDV 2	PHDV 3	CH80SCI
Image size w×h (pix.)	640×480	640×480	640×480	384×288	640×360	640×360	640×360	856×481
#Images train / test	198 / 201	111 / 111	267 / 264	500 / 501	613 / 0	0 / 904	0 / 505	370 / 612
Annotations ours / suppl.	✗ / ✓	✗ / ✓	✗ / ✓	✓ / ✗	✓ / ✗	✓ / ✗	✓ / ✗	✓ / ✗
Training im. rot. / flipp.	$0^\circ - 90^\circ$ ✓/ ✗	$0^\circ - 90^\circ$ ✓/ ✗	$0^\circ - 90^\circ$ ✓/ ✗	$270^\circ - 360^\circ$ ✓/ ✗	$0^\circ - 180^\circ$ ✓/ ✓	$0^\circ - 180^\circ$ ✓/ ✓	$0^\circ - 180^\circ$ ✓/ ✓	$270^\circ - 90^\circ$ ✓/ ✗
#End-eff. templ.	111	79	102	86	1122	–	–	455
#Shaft templ.	6	6	6	5	10	–	–	10
End-eff. HOG size	6×6	10×10	14×14	20×20	10×10	–	–	12×12
Shaft width	20	15	25	35	–	25	25	–
#model param.	$\sim 4.1 \times 10^4$	$\sim 8.0 \times 10^4$	$\sim 2.0 \times 10^5$	$\sim 3.5 \times 10^5$	$\sim 1.1 \times 10^6$	–	–	$\sim 6.6 \times 10^5$
NMS	✗	✗	✗	✗	✓	✓	✓	✗
Time (sec)	0.5	0.8	1.5	3.8	–	17.9	17.9	6.6

as shown in Fig. **??**. In effect, at the cost of additional annotations, it produces more precise shape estimates of the end-effectors than state-of-the-art methods, such as Rieke et al. (2016), that estimate only the tips of the forceps.

Laparoscopy (`LAPA`) The dataset `LAPA` Sznitman et al. (2012) contains one video sequence of *in-vivo* laparoscopic surgery. It features two surgical tools with annotated tool center. In this work, we evaluate our algorithm on one tool, as in Rieke et al. (2016), which is a more interesting case. The other tool is mostly static throughout the sequence. For training and testing, we use our annotations of tool tips and shaft beginning.

Phantom da Vinci (`PHDV`) The dataset `PHDV` Padoy & Hager (2012) contains three video sequences of surgical suture manipulation over phantom background. The sequences feature two da Vinci needle drivers, which are always in view. Despite the phantom background and its low variation across the three sequences, the dataset is challenging due to rich articulations of cooperating robotic end-effectors, significant variation in length of both shafts, shadows left by the shafts, and the presence of the needle that is similar to robotic forceps.

Cholec80–Scissors (`CH80SCI`) The dataset Cholec80 Twinanda et al. (2017) contains 80 videos of cholecystectomy surgeries performed by 13 surgeons. Each video frame from the dataset is labelled with the presense of six types of tools. From this dataset we collected and annotated temporarily unrelated images of articulated, surgical scissors that depict the scissors with visible shaft. The collected dataset `CH80SCI` depicts scissors under challenging, working conditions, including blurry images of the tools, low contrasted and occluded end-effectors, significant changes of the shaft appearance under affine and projective transformations, blood and moisture on the tools, and several types of scissors.

5.2. Implementation

For all datasets, we equally configure our method and use fixed parameter settings. The appearance templates are defined in HOG feature space. We set HOG cell resolution to sbin=8 for `REMI`, `LAPA`, `PHDV` and to sbin=16 for `CH80SCI`, truncating histogram magnitudes to 0.2. Our image descriptors are the reduced

version of HOG Felzenszwalb et al. (2010b), without orientation-sensitive and texture features. We only use soft-binned, 9 absolute orientations of image gradients. Hence, its depth is 9+1, where additional feature captures occluded image boundaries. The width and height dimensions of image descriptors are given in Tab. 1.

Original HOG features were designed for recognizing diverse object categories. We focus on capturing the appearance of surgical tools. We regard these objects as a composition of elongated and scissorlike contours and model them monolithically. Importantly, our features have limited 'air' to overfit to training sequences. For instance, our model does not assume that a tool darker than the background during training will continue to be darker at test time. In this section, we show that such simple features achieve good results in practice. Clearly though, edge directions along the contours are usually correlated. We leave exploiting local co-occurrence of gradient orientations as an interesting future work.

We set the size of angular bins to 10°, 45°, and 20° to specify the labels for the end-effector forceps, end-effector head, and shaft, respectively. The SVM hyperparameters are asymmetrically set to $C^+ = 2.0$ and $C^- = 0.2$ thereby accounting for $m^+ \ll m^-$ imbalance between positive and negative training sets. The appearance regularization parameter is set to $\tau = 0.1$. At test time, we search over $K = 13$ pyramid levels, so that the image at the last level is two times smaller than the original image at the first level. At each level, we call Algorithm (1) with $N = 6$ and $N_* = 3$. Having obtained candidate detections with oriented bounding boxes, we non-maximum suppress them with the overlap threshold of 10%.

Computational complexity Let $|L|$ denote the size of the image domain, let $|\widehat{E}|$ be the average number of edges in graph $\mathcal{M}$ from an $\mathcal{E}$-part template to all $\mathcal{S}$-part templates, let $|\widehat{A}|$ be the average number of anchors per edge, and $|\widehat{S}|$ be the average length of instantiated line segments in the image. Then, the computational cost of convolution in Step 1 is $\mathcal{O}(|L|(n_\mathcal{E} + n_\mathcal{S}))$, of shaft pooling in Step 2 is $\mathcal{O}(|\widehat{S}||L_b|^2)$, and of inference in Step 3 is $\mathcal{O}(|\widehat{E}||\widehat{A}||L|n_\mathcal{E})$.

The end-effector templates rigidly model the appearance changes that arise from end-effector articulations. Hence, we have $n_{\mathcal{E}} \gg n_{\mathcal{S}}$, as shown in Tab. 1. Moreover, one end-effector type links to only few shaft types, and $|\widehat{E}| \ll n_{\mathcal{S}}$. Then, the number of assigned anchors to an edge $|\widehat{A}|$ is low and depends on aligning the annotated, end-effector and shaft keypoints in the learning phase.

Negative examples Some approaches to tool detection, e.g. Sznitman et al. (2012), Sznitman et al. (2014), retrieve negative samples from the background of the surgical sequence to discriminatively train their tool appearance models. The samples are mined either online during tracking Li et al. (2014b) or offline around the tool Wesierski et al. (2015). In our experiments, we train our models offline on a generic dataset Dalal & Triggs (2005b). It contains a large collection of diverse, outdoor scenes that were originally used for pedestrian detection. In effect, the negative parameters of our discriminative models are not biased to background feature statistics of individual surgical sequences.

Positive examples To make our comparison fair, we follow other methods Sznitman et al. (2012), Sznitman et al. (2014), Rieke et al. (2016), and use training sequences of respective datasets (Tab. 1) to train individual models of the instruments. Our learning procedure first computes window sizes of the end-effector and shaft parts from keypoint annotations of training images. In order to tightly align the examples per type, though, the procedure has to account for inaccurate annotations that frequently occur due to ambiguous locations of characteristic landmarks. The training examples should vary smoothly in scale across a video sequence. To this end, the procedure passes the window radii of the end-effector part through IIR filter $d^{(t)} = 0.1d^{(t)} + 0.9d^{(t-1)}$, where d is the length either of (i) the longer forcep of non-rigid tools or (ii) the longest distance of end-effector keypoints to the tool center for robotic tools. Including the head keypoints in (ii) accounts for possible foreshortening of the end-effector forceps.

The image examples are warped to canonical window size such that window radius is the mean of both extreme window radii. Positive examples in low resolution, which have radii that fall below the canonical radius, are assigned another type, as shown in Fig. 3. Their window sizes are set to the canonical

window size. In this way, our method can conveniently learn multiresolution appearance of the end-effector types. As our model uses a single shaft sub-part to represent the whole shaft, the sub-part is selected randomly from the set of shaft sub-parts in each training image. Finally, the preprocessed training set is augmented to form new end-effector articulations. Datasets are rotated every 5 degrees within specified angular range (Tab. 1). Additionally for the `PHDV` dataset, the images are flipped horizontally. The model is then trained on each non-rigid or robotic tool as shown in Fig. 3 and explained in Sec. 4.2.

5.3. Evaluation criteria

Following traditional evaluation protocols in human pose estimation, we use metrics: (i) strict percentage of correct pose (strictPCP) Ferrari et al. (2008) and (ii) recall-precision curves with average precision Everingham et al. (2010) of keypoints (APK) Yang & Ramanan (2013). In addition, we use metrics (iii) keypoint threshold bounding box (KBB) Rieke et al. (2016), being analogous to the PCK metric Yang & Ramanan (2013), and (iv) keypoint threshold (KT) Sznitman et al. (2012) to compare our detector with state-of-the-art trackers. The metrics (i), (iii), (iv) evaluate algorithm performance in localizing tool center and tool tips provided the correspondence between the ground truth and detections is given or unambiguous, as in the case of a single tool in the image (`REMI` and `LAPA` datasets). Here, we additionally describe the APK metric (ii) to address the performance of our method in the general scenario of articulated 2D pose estimation of multiple tools (`PHDV` dataset) without knowing the number of tools and without correspondence given a priori. We then derive a metric that we call Augmented Keypoint (K+) to evaluate the performance of our method in detecting the truncated shaft part. Without loss of generality, we assume that surgical tools have at most four parts, such that the shaft ending connects to the end-effector, which consists of two grippers and, optionally, the head (Fig. 3). In the diagnostic experiments (sec. 5.4) we arithmetically average the KBB metric scores of the respective end-effector keypoints, which comprise both tool tips of non-rigid tools and additionally the tool center for the robotic tools, in

order to collectively illustrate the performance of the method in estimating the pose of the end-effector part.

Ground truth Let $\widehat{k}_{i,j} \in \mathbb{R}^2_+$ denote ground truth x, y-location of i-th keypoint (also called landmark or joint) of j-th tool, which was manually annotated in a test image, where $j = 1 \dots J$. In the `REMI` and `LAPA` datasets, we have three keypoints $|i| = 3$ of the end-effector and one instrument $J = 1$. In the `PHDV` dataset, we have four keypoints $|i| = 4$ of the end-effector and two instruments $J = 2$. We intentionally omitted the keypoint of the beginning of the shaft part as we evaluate the shaft part jointly by its orientation and location (K+ metric), as described later in this section.

Test time Respectively, let $k_{i,m} \in \mathbb{R}^2_+$ denote the detected i-th keypoint of m-th tool in the test image, where $m = 1, \dots, M$. When a video sequence shows a single tool and an algorithm returns only the highest scoring candidate pose, the correspondence of keypoints is one–to–one $j = m = 1$. NMS is not required in this case. When multiple tools are present though, an evaluation protocol has to address the problem of ambiguous correspondences between candidates and the ground truth. The PCK metric uses ground truth bounding boxes of objects to find best associations. It first selects candidate boxes that sufficiently overlap a referenced ground truth box. Then, it finds highest scoring candidate in the selected set and matches the candidate to ground truth. In effect, it performs NMS but *locally* in windows that are anchored at ground truth bounding boxes of whole objects. On the other hand, the APK metric first non-maximum suppresses lower scoring candidates in the *whole* image, irrespectively of the ground truth. Then, the protocol selects candidate keypoints that are in sufficient proximity to the ground truth keypoint and picks the highest scoring one.

KT Given the resolved correspondence between detected and ground truth tools, the KT metric evaluates the closeness of detected and ground truth keypoints. The detected keypoint k_i is true positive when it lies within a circle of

radius T, anchored in the image at location $\widehat{k}_i$:

$$\|k_i - \widehat{k}_i\| < T \tag{10}$$

where $\|\cdot\|$ denotes euclidean distance. Evaluation protocol with the KT metric sets the proximity threshold T to an array of fixed, increasing pixel values. Although the array spans from-precise-to-loose regimes of part localization, it depends on image size and accounts for no changes in scale of the object across a dataset.

strictPCP On the other hand, the strictPCP metric evaluates the closeness of detected and ground truth sticks, normalized by the length of the latter. A stick is defined by its two endpoints k_{i_1} and k_{i_2} and indicates part segment. The detected part is true positive when it satisfies the following two inequalities of normalized distances:

$$\|k_{i_1} - \widehat{k}_{i_1}\|/D < \alpha \quad \wedge \quad \|k_{i_2} - \widehat{k}_{i_2}\|/D < \alpha \tag{11}$$

where normalization $D = \|\widehat{k}_{i_1} - \widehat{k}_{i_2}\|$ is the distance between two ground truth endpoints of the part. It accounts for potential variation in scale of the part and makes the metric invariant to image size. Then, similarly to the KT metric, evaluation protocol sets proximity threshold $\alpha > 0$ to some fixed array. However, the strictPCP metric may be sensitive to the amount of foreshortening of the part, as warned by Yang & Ramanan (2013). Arguably, in the tool localization setting, the strictPCP metric is too loose for the truncated shaft part, which varies in length significantly and is often much longer than other tool parts.

KBB The KBB metric, which is de facto analogous to the PCK metric, evaluates the closeness of the detected and ground truth keypoints, normalized by object size:

$$\|k_i - \widehat{k}_i\|/D < \alpha \tag{12}$$

where $D = \max(w, h)$, and w and h are width and height of the ground truth bounding box of the tool end-effector, respectively. As the bounding box is axis-aligned and tightly cropped to contain all keypoints $\widehat{k}_{i,j}$ of the end-effector, the

max(·) operator ensures that $D > 0$ in case the gripper was closed and oriented either horizontally or vertically.

APK Unlike the PCK protocol, that locally resolves the correspondence problem of the whole tool, the APK protocol globally resolves ambiguities separately for each i-th keypoint. In the test image, APK assigns subsets of M candidate keypoints $k_{i,m}$, that satisfy (12), to J ground truth circles. The circles are anchored at $\widehat{k}_{i,j}$ and have radii αD_j, where $D_j = \max(w_j, h_j)$. Notably, across candidate keypoints from the subset that compete for the j-th tool, the one with the highest score claims the ground truth keypoint. It is deemed true positive. The remaining ones, including the ones with distance smaller to the ground truth, are false positive. Though, this can be accounted for by varying the radii through α threshold. Conversely, across ground truth keypoints that compete for the m-th candidate, the one with the longest radius claims the candidate. Hence, the APK metric depends on NMS approach and can be biased towards larger scale candidates. Finally, when the tool is absent, any candidate detection is false positive while unmatched tools are false negatives. Counting true positive, false positive, and false negative detections allows obtaining precision-recall curves together with average precision (AP) Everingham et al. (2010) per keypoint.

K+ The length of the shaft part can vary significantly during surgery. In view of this, we derive another metric from KT and KBB metrics that evaluates the candidate detections of the truncated shaft part. We can use this metric when single or multiple tools are present. Notably, one can parameterize the shaft part by location of its ending and by its orientation. The metric jointly computes (i) euclidean distance between the endings of the m-th detected and j-th ground truth shafts $r = \|k_{i,m} - \widehat{k}_{i,j}\|$ and (ii) absolute angular difference between their orientations $\varphi = |\varphi_m - \widehat{\varphi}_j|$, which are obtained from two shaft endpoints. Thus, the 2D location of the candidate shaft part (r, φ) is represented in the polar coordinate system that is anchored at the ground truth shaft part.

Normalized distance to origin then yields:

$$\|k_{i,m} - \widehat{k}_{i,j}\|/D_j + |\varphi_m - \widehat{\varphi}_j|/D_\varphi < \alpha \tag{13}$$

where D_φ is some predetermined angular resolution step. When $D_j = \max(w_j, h_j)$, and w_j and h_j are width and height of the ground truth bounding box of the j-th tool end-effector, the euclidean distance is normalized according to the KBB metric. Alternatively, one can normalize the euclidean distance by a fixed value, after the KT metric. As the normalized distance (12) additionally includes normalized angular offset in (13), we call this metric Augmented Keypoint (referred to as K+). The angular part of this metric is insensitive to changes of part length, foreshortening, scale, and by image size.

5.4. Diagnostic experiments

In this section we analyze and extensively evaluate several, crucial aspects regarding training and testing phases of our method. Notably, we are interested whether and to what extent (i) feature regularization and multiresolution, (ii) shaft score pooling, and (iii) including the shaft part in the model have influence on the performance of our method. The three experiments differ in terms of dataset constraints: one or two tools, and model structures: only end-effector part, only shaft part, and end-effector+shaft parts detectors. Coping with specific constraints of each diagnostic test would require tweaking SVM threshold and non-maximum suppression. Hence, in order to judiciously evaluate the tests (i)–(iii) and standardize testbed settings, in each test for the `PHDV` dataset we first non-maximum suppress candidate detections and then select top-2 scoring detections. We then report recall results of the diagnostics experiments in Fig. 5.

5.4.1. Training phase

Natural data sources often have strong bias of pose configurations Tran & Forsyth (2010). Hand-selecting an image dataset from recorded video sequences, which is representative of the pose configurations in the appearance

space, is cumbersome. Learning tool models directly from the annotated, uncut sequences of tool movement, is more favorable. A straightforward way then to train a model from video snippets is to collect positive examples of mixture types from all, consecutive image frames. In effect, some tool articulations could receive less diverse training images than other ones. This would lead, in turn, to adversarial, repetitive background features of positive examples across mixture types. We show that this negatively affects the performance of the model at test time. In this case, the SVM model, which is a linear combination of training examples, could be biased to the background features. To account for frequently low background diversity of video training data, we attenuate model features that correspond to background features of positive examples, as discussed in Sec. 4.2. This has critical effect on our model. It has fewer parameters to learn (about 50% fewer on average) thereby being less prone to overfitting. Additionally, it can focus on visually more important tool-background features during training by assigning positive and negative weights particularly at tool contours (Fig. 4).

Our experiments illustrate that feature regularization R in (9) significantly improves model performance at test time. We trained individual tool models (a) with (`reg`) and (b) without (`noreg`) feature regularization of end-effector and shaft part mixture on all three datasets (Sec. 5.1). In Fig. 5 (left column) we depict summarized results of recall performance for the datasets across several proximity thresholds. Additionally, we trained the models (c) with and (d) without multiresolution features (Sec. 5.2). Models without multiresolution features have approximately half the number of the end-effector templates with respect to the multiresolution models. They are also roughly two times faster to evaluate. For the keypoints of the end-effector part, the regularized model without multiresolution features performs slightly better than the regularized, multiresolution model. On the other hand, the regularized, multiresolution model evidently outperforms its counterpart for the shaft part.

The collective performance of these training configurations is dominated by the `PHDV` dataset, which has the largest collection of test images. However, the

results from experiments on individual datasets are analogous. In particular, regularized features significantly improve performance for end-effector keypoints (by $15\% - 20\%$) in each dataset. Then, the tool center is best detected with regularized, multiresolution features (by $5\% - 10\%$) on `REMI` and `LAPA` datasets. In the remaining experiments, we present the results for tool models with regularized, multiresolution features.

5.4.2. Testing phase

Shaft part leaves characteristic, visual cues that discriminate the whole tool from the background. In view of this, one would like to take advantage of the shaft part to help locate surgical tools. On the other hand, one has to consider the varying length of the shaft during surgery due to truncation at the image border. We approach this problem with several pooling procedures that differently aggregate the shaft scores along an instantiated line segment. The procedures are evaluated on the `PHDV` dataset. Present significant variation in length and orientation of both shafts makes this testbed an informative choice. The datasets `REMI` and `LAPA` are not well suited for this experiment. Former violates our assumption that the shaft starts at the image border while the latter has only short shaft instances.

Pooling procedures Our detection pipeline (Fig. 2) requires an effective mechanism for aggregation of shaft filter scores. Yet, choosing a specific pooling layer that copes with varying shaft length affects model performance at test time. We experiment with five pooling methods for aggregating N scores over the interval $(l_{\mathcal{S}(1)}, l_{\mathcal{S}(k)})$ of k elements on instantiated line segment $(l_{\mathcal{S}(1)}, l_{\mathcal{S}(K)})$:

1. `Nrandom` – selects N scores at random locations within the interval,
2. `Nbest` – selects N best scores of the interval,
3. `partNbest` – selects N best scores over M subintervals of the segment, with local N_* best scores per subinterval, as described in Sec. 4.1
4. `mean` – computes arithmetic average of all scores over the interval,
5. `Nmedian` – selects N scores from the middle of the array of scores, sorted within the interval.

The performance of the pooling methods is quantitatively assessed under two scenarios, using (i) full detector, composed of end-effector and shaft part (Fig. 4), as well as (ii) shaft detector, composed of only the shaft part. The results for (i)–(ii) are depicted in Fig. 5 (middle column).

Full detector scenario (i) assesses the impact of each pooling procedure on the overall performance of our method in tool detection and pose estimation. Averaging over all shaft sub-part scores (`mean`-pooling) slightly deteriorates performance (by $1\% - 1.5\%$) in locating end-effector keypoints with respect to other pooling methods, but the other methods perform evidently better in locating the shaft part (by $5\% - 10\%$). We argue the consistently worse performance of `mean` owes to the fact that it has to average over all scores, which are obtained from the simple, shaft sub-part detector. On the other hand, two variants of N-best pooling, which select only most reliable scores of the shaft detector, consistently outperform other methods. The global N-best pooling (`Nbest`) is on par with local, partitioned N-best pooling (`partNbest`) in locating end-effector keypoints but performs worse in locating the shaft part. In the former case, we posit this negligible difference between both methods is dictated by the influence of the strong end-effector detector that dominates the shaft detector.

In scenario (ii), we isolate line segment detectors (1)–(5) from the strong end-effector detector in order to assess the impact solely of the pooling procedures. To this end, we first obtain candidate detections of the shaft part. Given $n_{\mathcal{S}}$ tables of shaft filter scores for an image frame, we pool the scores using methods (1)–(5) according to Algorithm (1). As the algorithm describes method (3), Lines 4–9 have to be replaced accordingly for each method. We obtain a table of result scores Y^j and associated table X^j that stores the first and last bounding box of shaft sub-parts. Then, we select in total 6000 highest scoring candidates from the score tables across pyramid levels, which are best at given image locations. This amounts to $\sim 20\%$ of all possible candidates. Finally, we non-maximum suppress the candidate shafts, where each one is spanned by the bounding boxes of the two shaft sub-parts. We set K^+ metric with $D_j = 25$ pixels, $D_\varphi = 3°$, and thresholds $\alpha = [1.0, \ldots, 3.5]$. The `partNbest` pooling

procedure ranks first, followed by `Nrandom`, `Nmedian`, `Nbest`, and `mean`.

We also evaluate the five pooling procedures qualitatively over the whole euclidean and angular distance ranges for scenario (ii). Polar plots in Fig. **??**(top) show candidate detections for each pooling method with respect to the ground truth. The ground truth was centered in the origin of polar coordinate system. Method `partNbest` generates candidate detections that are most concentrated around the origin. Interestingly, the `random` procedure ranks second in overall, ranking even first at the strictest threshold $\alpha = 1.0$ (Fig. 5 (middle top)). The shaft has typically uniform appearance. Hence, given the scores along the shaft are comparable after filter convolutions, the manner of selection of the scores should ideally be invariant with respect to their location on the shaft part. Moreover, when the interval surpasses the true length of the shaft, the randomized selection method can select some scores not lying on the shaft. This is the desired behavior. The total score at the surpassed end location will be reduced. Though, we posit that `partNbest` is essentially better than `random` because it selects the scores in a systematic manner, enforcing step-wise structure on the instantiated line segment. On the other hand, method `Nbest` favors longer shafts, with many detections far from ground truth but with well estimated orientation. This is intuitive because the longer the interval for selection of N-best scores, the more likely it is for the method to aggregate higher scores, as shown Fig. **??** (bottom). Methods `mean` and `Nmedian` generally show weak performance.

Influence of shaft part Our model is a structured composition of end-effector part and shaft sub-part. In Fig. 5 (right) we investigate the influence of the shaft part on the performance of our full detector by pooling with `partNbest` an increasing number of sub-part scores. We test 8 detectors, with the number of sub-parts from 0 to 14. We start with end-effector mixture model, which uses no shaft sub-parts, as our baseline. End-effector detector performs quite well achieving collective recall of 55% for end-effector keypoints at threshold $\alpha = 0.1$. The threshold is rigorous but plausible as it converts, on average, to 5 pixels in the `PHDV` dataset, being one-fifth of forcep length and 10° of angular

deviation from true orientation of the forcep. Then, already a detector with two shaft sub-parts enjoys apparent increase in performance. The recall for the keypoints increases with the number of pooled sub-part scores and saturates at 6 sub-parts, suggesting that our method is insensitive from some configuration onward to tuning the testing parameters. A similar behavior can be observed for the shaft part. Consequently, our model can take advantage of the image evidence that is left by the shaft. Including the shaft part in the model improves overall performance.

5.5. Benchmark results

The proposed approach is inspired by the flexible mixtures of parts model (FMP) from Yang & Ramanan (2013). FMP uses structural features of deformation constraints and cooccurrences between pairs of part mixture components. It can successfully generalize to previously unseen object configurations by searching through exponentially large, synthesized mixtures of trees. However, one of its main limitations is double counting image evidence due to employed tree-graph structure. On the other hand, the graph enables efficient inference. In this work, we remove the structural features in modeling object deformations and transfer the appearance changes of deforming tools directly into the appearance templates. In effect, we monolithically model the appearance of the end-effector, which can be thought of as a 3– and 4–clique of parts for non-rigid and robotic tool, respectively. The distance between the keypoints is captured rigidly in the mixture components of our model.

We train and test the FMP model on the same data as our model using FMP code, which is available online. We use the same HOG features as for our method and run tests on 13 pyramid levels as well. Similarly to our method, by modeling the shaft part, we wish to examine its influence on the overall tool pose estimation. To this end, we train 5– and 6–part FMP model categories for the non-rigid and robotic tools, respectively, employing a fork-like graph structure. Specifically, for non-rigid tools, we start with a 3–part FMP model that connects tool tips to tool center, without the edge between the tips, and

	Shaft			
	REMI1	**REMI2**	**REMI3**	**LAPA**
POSE	–	–	–	–
FMP3	–	–	–	–
FMP6	**61.2**	18.0	**73.9**	–
Ours	55.2	75.7	48.5	36.3
OursFxd	–	**94.6**	–	–

	Center			
	REMI1	**REMI2**	**REMI3**	**LAPA**
POSE	**97.0**	**100**	**95.0**	58.0
FMP3	92.5	17.1	78.0	57.9
FMP6	92.5	18.0	68.2	–
Ours	92.5	73.7	85.2	**66.7**
OursFxd	–	96.4	–	–

	Left			
	REMI1	**REMI2**	**REMI3**	**LAPA**
POSE	64.7	24.6	**82.4**	–
FMP3	62.2	2.7	53.8	37.3
FMP6	60.7	0.0	36.4	–
Ours	**74.1**	3.6	75.8	**60.1**
OursFxd	–	**39.6**	–	–

	Right			
	REMI1	**REMI2**	**REMI3**	**LAPA**
POSE	**86.4**	**48.2**	74.7	–
FMP3	41.8	0.9	63.3	36.9
FMP6	38.8	0.0	53.0	–
Ours	62.7	0.9	**78.4**	**52.7**
OursFxd	–	21.6	–	–

Table 2: Consolidation of the results from Fig. 6 of POSE tracker Rieke et al. (2016), 3–part and 6–part FMP models Yang & Ramanan (2013), and our method. The scores were extracted at thresholds $\alpha = 2.0$ for the shaft part, at $KT = 20$ pixels for the tool center, and $\alpha = 0.2$ for the left and right forcep. Best scores are in bold.

	Shaft		
	LAPA	**CH80SCI**	**PHDV**
POSE	–	–	–
FMP3	–	–	–
FMP6	–	**36.6**	72.4
Ours	39.1	28.0	**74.4**

	Center		
	LAPA	**CH80SCI**	**PHDV**
POSE	52.6	–	–
FMP3	58.7	11.3	63.0
FMP6	–	17.5	80.0
Ours	**64.3**	**19.9**	**85.5**

	Left		
	LAPA	**CH80SCI**	**PHDV**
POSE	72.3	–	–
FMP3	45.1	19.0	52.5
FMP6	–	28.1	57.5
Ours	**80.3**	**31.7**	**76.9**

	Left		
	LAPA	**CH80SCI**	**PHDV**
POSE	**77.4**	–	–
FMP3	44.9	14.4	59.4
FMP6	–	19.8	68.9
Ours	74.9	**24.3**	**80.0**

Table 3: Consolidation of the results from Fig. 7 of POSE tracker Rieke et al. (2016), 3–part and 6–part FMP models Yang & Ramanan (2013), and our method. The scores were extracted at thresholds $\alpha = 2.0$ for the shaft part and at $\alpha = 0.2$ for the tool center and end-effector tips. Best scores are in bold.

finish with a 5–part FMP model, where the last three parts in the graph capture the appearance of the shaft. For both tool types, we intentionally attribute the part that joins the shaft part with the end-effector part (i.e., shaft ending) to the shaft in order to separate the evaluation of the influence of the models on both parts. Hence, the end-effector of non-rigid and robotic tools consist of 2 and 3 parts, respectively. Each FMP model category was trained with the following configurations: 3 to 6 mixture components per part for the `REMI` and `LAPA` datasets, and 6 to 12 components per part for the `PHDV` and `CH80SCI` datasets, as in the latter case the angular motion of the shaft is 2X larger. We trained first without and then with latent update of mixture components. In the evaluation, we computed the shaft orientation as the arithmetic mean of shaft part locations. In each model category, models with more components and with the latent update yielded consistently better results than other training configurations. In further analysis we report the results for FMP models with maximal number of components per part.

As shown in Fig. 5(right), the proposed rigid part mixtures model localizes the shaft part and the end-effector keypoints consistently better than the FMP model, equipped with the growing number of parts. Interestingly, while FMP models also benefit from including the shaft part in estimating the articulation of the end-effector part, their performance slightly decreases with more shaft parts. Furthermore, detailed results for each dataset in Fig. 6 confirm that our method is better especially in estimating the end-effector articulation. At the same time, matching a 6–part FMP model required, on average, about 1.2 sec per frame. Our method has higher computation cost as it requires more templates to capture tool articulations.

We now discuss quantitative results from Fig. 6 and in Tab. **??**, where we compare our method to state-of-the-art methods: POSE Rieke et al. (2016), ITOL Li et al. (2014b), DDVT Sznitman et al. (2014), including our method OURS15 from Wesierski et al. (2015) on two public datasets `REMI` and `LAPA` of non-rigid tools. Moreover, we evaluate our method on `PHDV` dataset of two robotic tools.

Our detector with the proposed rigid part mixtures model achieves competitive state-of-the-art results on public benchmarks without tracking and manual initialization. Except for OURS15, the other methods are trackers that use temporal information to constrain the search space of target objects. Similarly to OURS15, the proposed detector processes every video frame individually. Owing to improved learning, pooling, and efficient detection procedures, it outperforms our previous work in detection and computation performance on the `REMI` dataset, which required tens of seconds to process each image.

As in Rieke et al. (2016), Sznitman et al. (2014), Wesierski et al. (2015), we trained individual tool models for video sequences. We used configuration settings from Tab. 1. We mostly compare our method to POSE that estimates the location of tool center, left and right forceps (PCP) and their tips. We additionally estimate the location and orientation of the shaft part using the K+ metric. We use the shaft width and the detected window sizes for the KT-based and KBB-based K+ evaluation, respectively. Both evaluation protocols are correlated though, as shown for `LAPA` sequence.

The KT-based evaluation shows that our method is mostly on par with or better than other methods. We score better than ITOL on `REMI 1` and worse on `LAPA` but we output full pose of the instrument, do not require manual initialization, and our method relies only on the detector. On the other hand, the KBB-based evaluation shows that our method is better than POSE on `LAPA` sequence. However, the POSE method is evidently better on `REMI 2` sequence under the strictPCP metric. Our method uses the same train and test settings for all sequences (Sec. 5.2). When we manually decrease the size of our appearance templates and decrease the range of scales in the image pyramid to $K = 3$, our method (Ours-fixed) effectively approaches the performance of POSE on `REMI 2` sequence. Tweaking appearance resolution helps because the training data of `REMI 2` are unrepresentative. The tool has high resolution in the training set while it has low resolution at test time. When our method is supervised to learn appearance templates in lower resolution, its localization precision significantly increases.

Our method extends to pose estimation of multiple, robotic tools. We evaluated its performance using the APK metric that asks for locations of tool keypoints without giving the number of tools a priori. The results in Fig. 6 (bottom row) indicate that the algorithm achieves high performance in locating tool center and tool tips, outperforming the FMP method in its all model categories (i.e., 3–part to 6–part models with 12 mixture components per part). Consequently, our approach, which captures rich, pose-specific variations of end-effector and shaft appearance, can successfully estimate articulated pose of non-rigid and robotic tools.

6. Discussion

In overall, we give competitive results to the state-of-the-art. We argue, though, that the strength of our approach lies in its demonstrated applicability to pose estimation of non-rigid and robotic tools. We showed, qualitatively and quantitatively, that explicit transfer of pose and skeleton keypoints is precise using our model-based approach. We attribute its good performance primarily to jointly trained model of end-effector and shaft parts. The model uses the shaft part and regularizes pose-specific features, being a key to effective detection in and learning from biased video data. While the versatile applicability comes at the increased computational cost, our method allows multiple optimization variants.

This section locates our method in broader context of instrument recognition. It highlights and gives rationale for future work in several aspects that include computational efficiency and more complex model learning procedures from videos.

6.1. Pose estimation as pose recognition

Proposed tool-configurable, rigid parts mixtures model estimates tool pose essentially by recognizing particular types of tool parts at test time. Correctly selected appearance templates precisely transfer the keypoints, which are as-

signed to the templates at training time. Hence, in our approach, correct recognition of previously seen tool poses is prerequisite for correct pose estimation.

We regard formulating the problem of pose estimation as pose recognition as a crucial advantage of our approach. First, surgical instruments are rigid, non-rigid, and articulated with a variety of shapes of the end-effector but often the instruments share repetitive, image gradient features that are attributed to straight, elongated shaft. In this work, we extensively study several shaft detection procedures and show that shaft part is an important visual cue in instrument pose estimation. Potential applicability across many tool types thus motivates further development of shaft detection methods. Second, the variety of pose-specific shapes is a subcategory of the general class of intrinsic shapes of different end-effector tool types in the chosen appearance space. We showed on four datasets that our spatial, bipartite detector is quite successful at sub-model selection. Hence, in our future work, we will go beyond model configurability and investigate the problem of generalization at training time – jointly learning multiple tool models Twinanda et al. (2017) for spatio-temporal tool detection.

6.2. Computation

Test-time efficiency of our algorithm depends on the size of template mixtures that capture pose-specific tool appearance and on the partitioned N-best pooling procedure. The number of learned appearance templates ranges from ~ 80 to ~ 1100 (Tab. 1). Moreover, our detector exhaustively searches for the end-effector in the whole image and in constrained, angular space of orientations for the shaft across many scales. Processing times of our mexed Matlab/C++ CPU implementation are given in Tab. 1.

Generally, our large sets of templates are amenable to multiple, existing CPU acceleration variants, such as Sadeghi & Forsyth (2014), Dean et al. (2013), Kokkinos (2013). The first one efficiently computes and evaluates HOG features across scale levels. The second one enjoys constant time computational complexity in matching millions of filters that capture hundreds of thousands of object categories and depends linearly on filter locations. The third one finds

and shares clustered appearance across many templates through sparse coding and substantially accelerates model evaluation without loss in performance. Our spatial pooling procedure lends itself to parallelization as well. Further optimization of the algorithm can then be achieved by reducing the number of templates and reducing the location search space based on (i) online, robot encoder readings Ye et al. (2016), (ii) tracking Henriques et al. (2015), and (iii) coarse-to-fine feature hierarchies Salakhutdinov et al. (2011). In addition, our algorithm consumes less than 50 MB of memory at test time for e.g. da Vinci tool model (Tab. 1).

Assuming that we are given constraints that apply to specific surgeries, such as robot encoder readings to limit the number of templates, we can predict potential computational gains through the following ablative analysis. The model Ours-fixed for `REMI 2` sequence, that operates at 3 scales and uses 79 end-effector filters, has running time of 1.1 sec per frame. When it additionally uses only 1 orientation filter of the shaft sub-part, the time drops further to 0.5 sec. On the other hand, when the robotic tool model for the `PHDV` sequences is matched at 3 scales, the processing time reduces to 7.0 sec per frame with respect to 17.9 sec in the original, fixed setting of 13 scales. When it uses 100 instead of 1122 end-effector filters, the time further decreases to 1.8 sec, reaching 0.6 sec with 2 instead of 10 shaft filters.

6.3. Foreground vs Background and Foreground vs Foreground

Let us begin this section by introducing the following example that will be our point of reference. The example considers two related pose types of robotic end-effector. Let articulation token  denote Y-open gripper and token  denote Y-closed gripper. Without loss of generality, let us assume there are two error sources that can occur at test-time: (i) given Y-closed gripper and its location, the algorithm finds Y-open gripper, and vice versa, (ii) given Y-open gripper and its location, the algorithm finds Y-closed gripper.

The former case stems from incorrect foreground-background classification. Apart from finding features in the test image that are relevant to the tool head

and right forcep, the algorithm also finds background features that it considered relevant for the left forcep. Such a type of errors suggest shortcomings in the observation function or in the model. Although our algorithm sometimes localizes robotic tools in the background, we found that it almost never (i.e. only one time) made this type of mistakes – the algorithm is good at detecting closed end-effectors.

The latter case is the result of incorrect foreground-foreground classification, usually called double counting error, where particular configuration sub-space is double counted. Errors of double counting at test time can occur in object pose estimation when object configuration or camera viewpoint allow two similarly looking parts, such as end-effector forceps, occlude each other at training time. In order to allow tractable inference, tree-graph models usually put no edge between such parts and suffer from double-counting. Our method partly deals with double counting by monolithically modeling the appearance of the end-effector, which can be thought of as a clique of three parts. The distance between forceps is captured rigidly in the appearance templates. However, the algorithm sometimes selects a wrong template. This happens especially when the correct and incorrect template are both plausible given the evidence. Then, the algorithm double-counts the right forcep, as shown for the right tool in Fig. **??**F. In this case, both filters are far in the configuration space but share most of the appearance space.

On the one hand, we attribute the ability of the model to correctly estimate poses of closed end-effectors to their strongly trained appearance templates. Given our cost-sensistive SVM formulation (9), such types of end-effectors should score above the same margin as open end-effectors. Conversely, double counting errors occur because of the same reason. Filters of open and closed end-effectors are scaled to score above the same threshold. Most likely then, only slight differences in their rigidly captured appearance influence the decision of the SVM predictor. Indeed, we found that correct Y-open end-effector types belonged up to top 5 detections (out of more than 1000 types) in most cases of double counting, where all types before the correct type were close

neighbors (in angular sense) of the incorrect, top scoring Y-closed end-effector. We use slack rescaling to train our model effectively discriminate foreground from the background. However, additional, local margin rescaling is one attractive avenue to be explored in order to reduce double counting errors.

Moreover, the SVM formulation jointly trains multiple, pose-specific appearance templates. The objective states that the model has to discriminate between foreground and background features but not between the multiple foreground features. Discrimination especially between often confused pose-specific sub-models, which lead for instance to double-counting errors, might help our approach.

One workaround might be to learn the model in the classical one-vs-all and one-vs-one settings where given sub-model has to be well separated from the remaining sub-models in the appearance space. The former is attractive with respect to the latter setting because it yields linear instead of quadratic number of sub-models. However, it can be problematic. In one-vs-all, when two end-effector types differ by one forcep, as in the discussed example from Fig. **??**F, the classic structured prediction SVM will generally learn that the only features that discriminate between the two poses belong to the left forcep. At the same time though, the SVM would also have to discern the reverse articulation, where double-counting occurs for the left forcep. Although this can be avoided with one-vs-one learning, such classification would still be weak if the SVM had only one layer – there are many forcep-like contours in the image. A more complex, two-layer SVM architecture, which first discriminates foreground from background and then one foreground mask from another foreground mask, might also mitigate the double-counting problem.

6.4. Video data

Most methods develop data-driven appearance models of surgical tools on in-vivo, ex-vivo, and phantom data. Clearly, testing tool models in in-vivo setting is a crucial validation step. However, collecting ex-vivo and phantom data is still easier. Although our approach performs quite well on in-vivo and

phantom videos at test time, our future work will focus on the problem of modeled data transfer – training a given tool model on phantom and ex-vivo data and then transferring and testing the model in in-vivo settings. In this way, it might be easier to automatically generate and explicitly learn all possible tool poses, especially those with end-effector foreshortening and self-occlusions. At the same time though, one would like to account for background bias in phantom and ex-vivo videos. Our coarse segmentation masks help significantly improve pose estimation by attenuating background features in our regularized SVM formulation. As instrument pose estimation is an important, exciting, and increasingly popular computer vision problem, we anticipate more democratized and diverse video collections in the nearest future.

7. Conclusions

We proposed a configurable, rigid part mixtures model for structurally representing the appearance of non-rigid and robotic instruments in video-assisted surgeries. We described joint model matching and learning procedures for articulated tool pose estimation. The model jointly explains the evolving object structure in videos by switching between part mixture components that rigidly encode pose-specific appearances of the tool. Rigidly capturing end-effector appearance allows explicit transfer of keypoint meta-data of the detected components. In effect, our versatile approach reaches state-of-the-art results in pose estimation on two public benchmarks and can precisely delineate end-effector skeleton. Conducted diagnostic experiments show that including the shaft part into our model improves estimation of end-effector articulation. We also demonstrated that proper regularization of model features significantly improves pose estimation when the model is trained on videos.

Acknowledgment

This work was supported in part by the National Science Centre in the framework of the SONATA 7 project UMO-2014/13/D/ST7/03358.

References

Ahmidi, N., Tao, L., Sefati, S., Gao, Y., Lea, C., Bejar, B., Zappella, L., Khudanpur, S., Vidal, R., & Hager, G. D. (2017). A dataset and benchmarks for segmentation and recognition of gestures in robotic surgery. *IEEE Trans. Biomed. Engineering*, *64*, 2025 –2041.

Allan, M., Ourselin, S., Thompson, S., Hawkes, D. J., Kelly, J., & Stoyanov, D. (2013). Toward detection and localization of instruments in minimally invasive surgery. *IEEE Trans. Biomed. Engineering*, *60*, 1050–1058.

Allan, M., Thompson, S., Clarkson, M. J., Ourselin, S., Hawkes, D., Kelly, J., & Stoyanov, D. (2014). 2D-3D pose tracking of rigid instruments in minimally invasive surgery. In D. Stoyanov, D. Collins, I. Sakuma, P. Abolmaesumi, & P. Jannin (Eds.), *Information Processing in Computer-Assisted Interventions (IPCAI)* (pp. 1–10). Springer volume 8498 of *Lecture Notes in Computer Science*.

Andriluka, M., Roth, S., & Schiele, B. (2008). People-tracking-by-detection and people-detection-by-tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1–8). IEEE.

Balicki, M., Han, J.-H., Iordachita, I., Gehlbach, P., Handa, J., Taylor, R. H., & Kang, J. U. (2009). Single fiber optical coherence tomography microsurgical instruments for computer and robot-assisted retinal surgery. In G.-Z. Yang, D. J. Hawkes, D. Rueckert, J. A. Noble, & C. J. Taylor (Eds.), *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (pp. 108–115). Springer volume 5761 of *Lecture Notes in Computer Science*.

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 Trans. Med. Imaging*, *34*, 2603–2617.

Burschka, D., Corso, J., Dewan, M., Lau, W., M., L., Lin, H., P., M., Ramey, N., Hager, G., Hoffman, B., Larkin, D., & Hasser, C. (2005). Navigating inner space: 3-D assistance for minimally invasive surgery. *Robotics and Autonomous Systems*, *52*, 5 – 26. Advances in Robot Vision.

Casals, A., Amat, J., & Laporte, E. (1996). Automatic guidance of an assistant robot in laparoscopic surgery. In *IEEE Int. Conf. Robot. Automat.* (pp. 895–900). volume 1.

Chmarra, M., Kolkman, W., Jansen, F., Grimbergen, C., & Dankelman, J. (2007). The influence of experience and camera holding on laparoscopic instrument movements measured with the trendo tracking system. *Surgical endoscopy*, *21*, 2069–2075.

Dalal, N., & Triggs, B. (2005a). Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 886–893). New York: IEEE Press.

Dalal, N., & Triggs, B. (2005b). Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 886–893). IEEE volume 1.

Dean, T., Ruzon, M. A., Segal, M., Shlens, J., Vijayanarasimhan, S., & Yagnik, J. (2013). Fast, accurate detection of 100,000 object classes on a single machine. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1814–1821).

Doignon, C., Graebling, P., & de Mathelin, M. (2005). Real-time segmentation of surgical instruments inside the abdominal cavity using a joint hue saturation color feature. *Real-Time Imaging*, *11*, 429–442.

Doignon, C., Nageotte, F., & de Mathelin, M. (2006). Segmentation and guidance of multiple rigid objects for intra-operative endoscopic vision. In *European Conference on Computer Vision (ECCV)* (pp. 314–327).

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, *88*, 303–338.

Felzenszwalb, P., Girshick, R., McAllester, D., & Ramanan, D. (2010a). Object detection with discriminatively trained part based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, *32*.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010b). Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, *32*, 1627–1645.

Ferrari, V., Marin-Jimenez, M., & Zisserman, A. (2008). Progressive search space reduction for human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1–8). IEEE.

Groeger, M., Arbter, K., & Hirzinger, G. (2008). Motion tracking for minimally invasive robotic surgery. In V. Bozovic (Ed.), *Medical Robotics* chapter 10. InTech, Education and Publishing. (978th ed.).

Haase, S., Wasza, J., Kilgus, T., & Hornegger, J. (2013). Laparoscopic instrument localization using a 3-D Time-of-Flight/RGB endoscope. In *IEEE Workshop on Applications of Computer Vision (WACV)* (pp. 449–454). IEEE Computer Society.

Hejrati, M., & Ramanan, D. (2014). Analysis by synthesis: 3d object recognition by object reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2449–2456).

Henriques, J. F., Caseiro, R., Martins, P., & Batista, J. (2015). High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, *37*, 583–596.

Kokkinos, I. (2013). Shufflets: Shared mid-level parts for fast object detection. In *IEEE International Conference on Computer Vision (ICCV)* (pp. 1393–1400).

Krupa, A., Gangloff, J., Doignon, C., de Mathelin, M., Morel, G., Leroy, J., Soler, L., & Marescaux, J. (2003). Autonomous 3-d positioning of surgical instruments in robotized laparoscopic surgery using visual servoing. *IEEE Trans. Robot. Autom.*, *19*, 842–853.

Kumar, S., Narayanan, M. S., Singhal, P., Corso, J. J., & Krovi, V. (2013). Product of tracking experts for visual tracking of surgical tools. In *IEEE International Conference on Automation Science and Engineering (CASE)* (pp. 480–485). IEEE.

Kumar, S., Narayanan, M. S., Singhal, P., Corso, J. J., & Krovi, V. (2014). Surgical tool attributes from monocular video. In *IEEE International Conference on Robotics and Automation (ICRA)* (pp. 4887–4892). IEEE.

Kumar, S., Sovizi, J., Narayanan, M. S., & Krovi, V. (2015). Surgical tool pose estimation from monocular endoscopic videos. In *IEEE International Conference on Robotics and Automation (ICRA)* (pp. 598–603). IEEE.

Kurmann, T., Neila, P. M., Du, X., Fua, P., Stoyanov, D., Wolf, S., & Sznitman, R. (2017). Simultaneous recognition and pose estimation of instruments in minimally invasive surgery. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 505–513).

Laina, I., Rieke, N., Rupprecht, C., Vizcaíno, J. P., Eslami, A., Tombari, F., & Navab, N. (2017). Concurrent segmentation and localization for tracking of surgical instruments. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 664–672).

Lalys, F., Bouget, D., Riffaud, L., & Jannin, P. (2013). Automatic knowledge-based recognition of low-level tasks in ophthalmological procedures. *Int. J. Computer Assisted Radiology and Surgery*, *8*, 39–49.

Lee, C., Wang, Y., Uecker, D., & Y., W. (1994). Image analysis for automated tracking in robot-assisted endoscopic surgery. In *International Conference on Pattern Recognition (IAPR)* (pp. 88–92). volume 1.

Lee, T., & Soatto, S. (2011). Learning and matching multiscale template descriptors for real-time detection, localization and tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1457–1464). IEEE.

Li, Y., Chen, C., Huang, X., & Huang, J. (2014a). Instrument tracking via online learning in retinal microsurgery. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* Lecture Notes in Computer Science (pp. 464–471). Springer.

Li, Y., Chen, C., Huang, X., & Huang, J. (2014b). Instrument tracking via online learning in retinal microsurgery. In P. Golland, N. Hata, C. Barillot, J. Hornegger, & R. Howe (Eds.), *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (pp. 464–471). Springer volume 8673 of *Lecture Notes in Computer Science*.

Liu, K., Skibbe, H., Schmidt, T., Blein, T., Palme, K., Brox, T., & Ronneberger, O. (2014). Rotation-invariant hog descriptors using fourier analysis in polar and spherical coordinates. *International Journal of Computer Vision*, *106*, 342–364.

López-Sastre, R. J., Tuytelaars, T., & Savarese, S. (2011). Deformable part models revisited: A performance evaluation for object category pose estimation. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)* (pp. 1052–1059).

Maier-Hein, L., Groch, A., Bartoli, A., Bodenstedt, S., Boissonnat, G., Chang, P.-L., Clancy, N., Elson, D. S., Haase, S., Heim, E. et al. (2014). Comparative validation of single-shot optical techniques for laparoscopic 3-d surface reconstruction. *IEEE Trans. Med. Imaging*, *33*, 1913–1930.

Mckenna, S. J., Charif, H. N., & Frank, T. (2005). Towards Video Understanding of Laparoscopic Surgery: Instrument Tracking. In *Image and Vision Computing New Zealand*.

Nageotte, F., de Mathelin, M., Doignon, C., Soler, L., Leroy, J., & Marescaux, J. (2004). Computer-aided suturing in laparoscopic surgery. In *International Congress Series* (pp. 781–786). Elsevier volume 1268.

Oropesa, I., Sánchez-González, P., Chmarra, M. K., Lamata, P., Fernández, Á., Sánchez-Margallo, J. A., Jansen, F. W., Dankelman, J., Sánchez-Margallo, F. M., & Gómez, E. J. (2013). Eva: laparoscopic instrument tracking based on endoscopic video analysis for psychomotor skills assessment. *Surgical endoscopy*, *27*, 1029–1039.

Padoy, N., & Hager, G. D. (2012). Deformable tracking of textured curvilinear objects. In *British Machine Vision Conference (BMVC)* (pp. 1–11).

Ramakrishna, V., Munoz, D., Hebert, M., Bagnell, J., & Sheikh, Y. (2014). Pose machines: Articulated pose estimation via inference machines. In *European Conference on Computer Vision (ECCV)* (pp. 33–47). Springer.

Ramanan, D., Forsyth, D. A., & Zisserman, A. (2007). Tracking people by learning their appearance. *IEEE Trans. Pattern Anal. Mach. Intell.*, *29*, 65–81.

Reiter, A., Allen, P., & Zhao, T. (2012a). Feature classification for tracking articulated surgical tools. In N. Ayache, H. Delingette, P. Golland, & K. Mori (Eds.), *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (pp. 592–600). Springer volume 7511 of *Lecture Notes in Computer Science*.

Reiter, A., & Allen, P. K. (2010). An online learning approach to in-vivo tracking using synergistic features. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 3441–3446).

Reiter, A., Allen, P. K., & Zhao, T. (2012b). Marker-less articulated surgical tool detection. *Computer Assisted Radiology and Surgery*, .

Reiter, A., Allen, P. K., & Zhao, T. (2013). Appearance learning for 3d tracking of robotic surgical tools. *The International Journal of Robotics Research*, (pp. 1–15).

Richa, R., Balicki, M., Meisner, E., Sznitman, R., Taylor, R., & Hager, G. (2011). Visual tracking of surgical tools for proximity detection in retinal surgery. In R. H. Taylor, & G.-Z. Yang (Eds.), *Information Processing in Computer-Assisted Interventions* (pp. 55–66). Springer Berlin Heidelberg volume 6689 of *Lecture Notes in Computer Science*.

Rieke, N., Tan, D. J., di San Filippo, C. A., Tombari, F., Alsheakhali, M., Belagiannis, V., Eslami, A., & Navab, N. (2016). Real-time localization of articulated surgical instruments in retinal microsurgery. *Medical Image Analysis*, *34*, 82–100.

Sadeghi, M. A., & Forsyth, D. (2014). 30hz object detection with dpm v5. In *European Conference on Computer Vision (ECCV)* (pp. 65–79). Springer.

Saint-Pierre, C.-A., Boisvert, J., Grimard, G., & Cheriet, F. (2011). Detection and correction of specular reflections for automatic surgical tool segmentation in thoracoscopic images. *Machine Vision and Applications*, *22*, 171–180.

Salakhutdinov, R., Torralba, A., & Tenenbaum, J. (2011). Learning to share visual appearance for multiclass object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1481–1488). IEEE.

Sarikaya, D., Corso, J., & Guru, K. (2017). Detection and localization of robotic tools in robot-assisted surgery videos using deep neural networks for region proposal and detection. *IEEE Trans. Med. Imaging*, .

Sen, S., Garg, A., Gealy, D. V., McKinley, S., Jen, Y., & Goldberg, K. (2016). Automating multi-throw multilateral surgical suturing with a mechanical nee-

dle guide and sequential convex optimization. In *IEEE International Conference on Robotics and Automation (ICRA)* (pp. 4178–4185). IEEE.

Speidel, S., Delles, M., Gutt, C. N., & Dillmann, R. (2006). Tracking of instruments in minimally invasive surgery for surgical skill analysis. In G.-Z. Yang, T. Jiang, D. Shen, L. Gu, & J. Yang (Eds.), *MIAR* (pp. 148–155). Springer volume 4091 of *Lecture Notes in Computer Science*.

Speidel, S., Sudra, G., Senemaud, J., Drentschew, M., Müller-Stich, B. P., Gutt, C., & Dillmann, R. (2008). Recognition of risk situations based on endoscopic instrument tracking and knowledge based situation modeling. In *Medical Imaging* (pp. 69180X–69180X). International Society for Optics and Photonics.

Staub, C., Panin, G., Knoll, A., Bauernschmitt, R., & Munich, G. H. C. (2010). Visual instrument guidance in minimally invasive robot surgery. *International Journal on Advances in Life Sciences*, *2*.

Sznitman, R., Ali, K., Richa, R., Taylor, R., Hager, G., & Fua, P. (2012). Data-driven visual tracking in retinal microsurgery. In N. Ayache, H. Delingette, P. Golland, & K. Mori (Eds.), *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (pp. 568–575). Springer volume 7511 of *Lecture Notes in Computer Science*.

Sznitman, R., Basu, A., Richa, R., Handa, J., Gehlbach, P., Taylor, R., & Jedynak, G., B.and Hager (2011). Unified detection and tracking in retinal microsurgery. In G. Fichtinger, & T. Martel, A.and Peters (Eds.), *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (pp. 1–8). Springer, Heidelberg volume 6891 of *Lecture Notes in Computer Science*.

Sznitman, R., Becker, C., & Fua, P. (2014). Fast part-based classification for instrument detection in minimally invasive surgery. In P. Golland, N. Hata, C. Barillot, J. Hornegger, & R. Howe (Eds.), *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (pp. 692–699). Springer volume 8674 of *Lecture Notes in Computer Science*.

Sznitman, R., Richa, R., Taylor, R. H., Jedynak, B., & Hager, G. D. (2013). Unified detection and tracking of instruments during retinal microsurgery. *IEEE Trans. Pattern Anal. Mach. Intell.*, *35*, 1263–1273.

Tonet, O., Thoranaghatte, R. U., Megali, G., & Dario, P. (2007). Tracking endoscopic instruments without a localizer: A shape-analysis-based approach. *Comput. Aid. Surg.*, *12*, 35–42.

Tran, D., & Forsyth, D. (2010). Improved human parsing with a full relational model. In *European Conference on Computer Vision (ECCV)* (pp. 227–240). Springer.

Twinanda, A. P., Shehata, S., Mutter, D., Marescaux, J., de Mathelin, M., & Padoy, N. (2017). Endonet: A deep architecture for recognition tasks on laparoscopic videos. *IEEE Trans. Med. Imaging*, *36*, 86–97.

Uecker, D. R., Lee, C., Wang, Y. F., & Wang, Y. (1995). Automated instrument tracking in robotically assisted laparoscopic surgery. *Journal of Image Guided Surgery*, *1*, 308–325.

Vogt, F., Krüger, S., Niemann, H., & Schick, C. (2003). A system for real-time endoscopic image enhancement. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* Lecture Notes in Computer Science (pp. 356–363). Springer.

Voros, S., Orvain, E., Cinquin, P., & Long, J.-A. (2006). Automatic detection of instruments in laparoscopic images: a first step towards high level command of robotized endoscopic holders. In *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006.* (pp. 1107–1112).

Wei, G.-Q., Arbter, K., & Hirzinger, G. (1997). Automatic tracking of laparoscopic instruments by color coding. In J. Troccaz, E. Grimson, & R. Masges (Eds.), *CVRMed-MRCAS'97* (pp. 357–366). Springer Berlin Heidelberg volume 1205 of *Lecture Notes in Computer Science*.

Wesierski, D., Wojdyga, G., & Jezierska, A. (2015). Instrument tracking with rigid part mixtures model. In *CARE*. Springer.

West, J. B., & Maurer Jr, C. R. (2004). Designing optically tracked instruments for image-guided surgery. *IEEE Trans. Med. Imaging,*, *23*, 533–545.

Wolf, R., Duchateau, J., Cinquin, P., & Voros, S. (2011). 3D tracking of laparoscopic instruments using statistical and geometric modeling. In G. Fichtinger, A. Martel, & T. Peters (Eds.), *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (pp. 203–210). Springer volume 6891 of *Lecture Notes in Computer Science*.

Yang, G.-Z., Cambias, J., Cleary, K., Daimler, E., Drake, J., Dupont, P. E., Hata, N., Kazanzides, P., Martel, S., Patel, R. V. et al. (2017). Medical robotics - regulatory, ethical, and legal considerations for increasing levels of autonomy. *Science Robotics*, *2*.

Yang, Y., & Ramanan, D. (2013). Articulated human detection with flexible mixtures of parts. *IEEE Trans. Pattern Anal. Mach. Intell.*, *35*, 2878–2890.

Ye, M., Zhang, L., Giannarou, S., & Yang, G.-Z. (2016). Real-time 3D tracking of articulated tools for robotic surgery. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (pp. 386–394). Springer.

Zhang, X., & Payandeh, S. (2002). Application of visual tracking for robot-assisted laparoscopic surgery. *J. Field Robotics*, *19*, 315–328.

Zhao, T., Zhao, W., Halabe, D., Hoffman, B., & Nowlin, W. (2010). Fiducial marker design and detection for locating surgical instrument in images. US Patent App. 12/428,657.

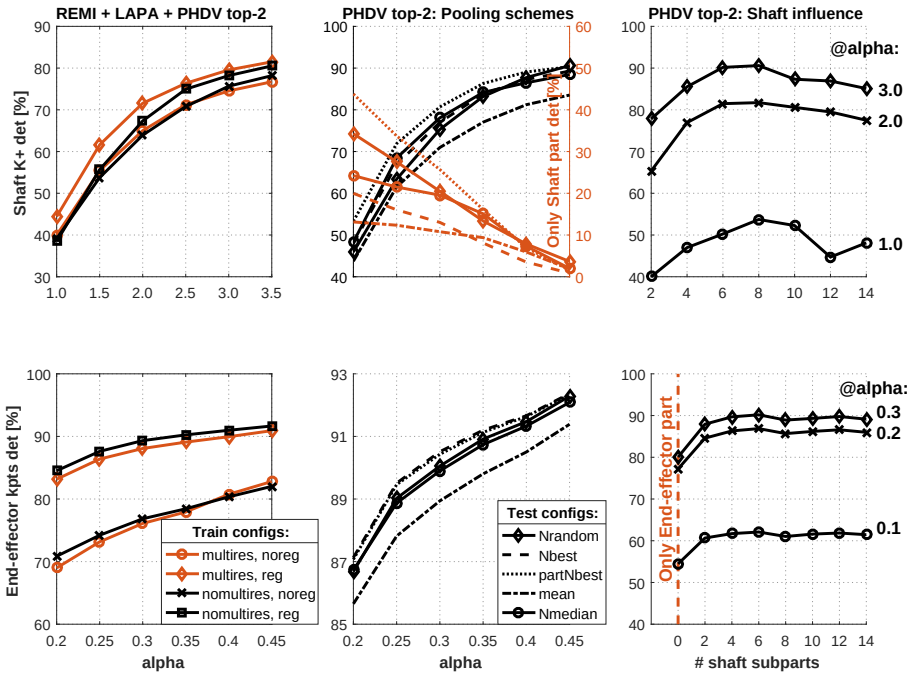

Figure 5: Diagnostic results in terms of recall. We use KBB metric for the end-effector keypoints (center, left/right forcep tips) and K+/KBB metric for the shaft (Sec. 5.3). For the `PHDV` dataset with two instruments, we select top 2 detections, after non-maximum suppression. **Left:** Collective evaluation of four training configurations on four datasets indicates that our model performs best with regularized features. **Middle:** At test time, our method best estimates the pose of robotic instruments when it uses `partNbest` procedure, which pools N best scoring sub-parts of the shaft with length constraints. Further validation shows that `partNbest` clearly outperforms other four pooling procedures as a line segment detector. **Right:** Influence of the shaft part on tool pose estimation grows with the number of pooled subparts, where end-effector detectors without the shaft part are the baseline. Saturation at 6 subparts suggests our method is insensitive to tuning the testing parameters. Moreover, our full instrument model can take advantage of the image evidence that is left by the shaft. Finally, extensive evaluation on three datasets shows that our model performs favorably wrt the FMP models Yang & Ramanan (2013) that represent the tools with 1– to 3– shaft parts (FMP4, FMP5, FMP6, respectively).

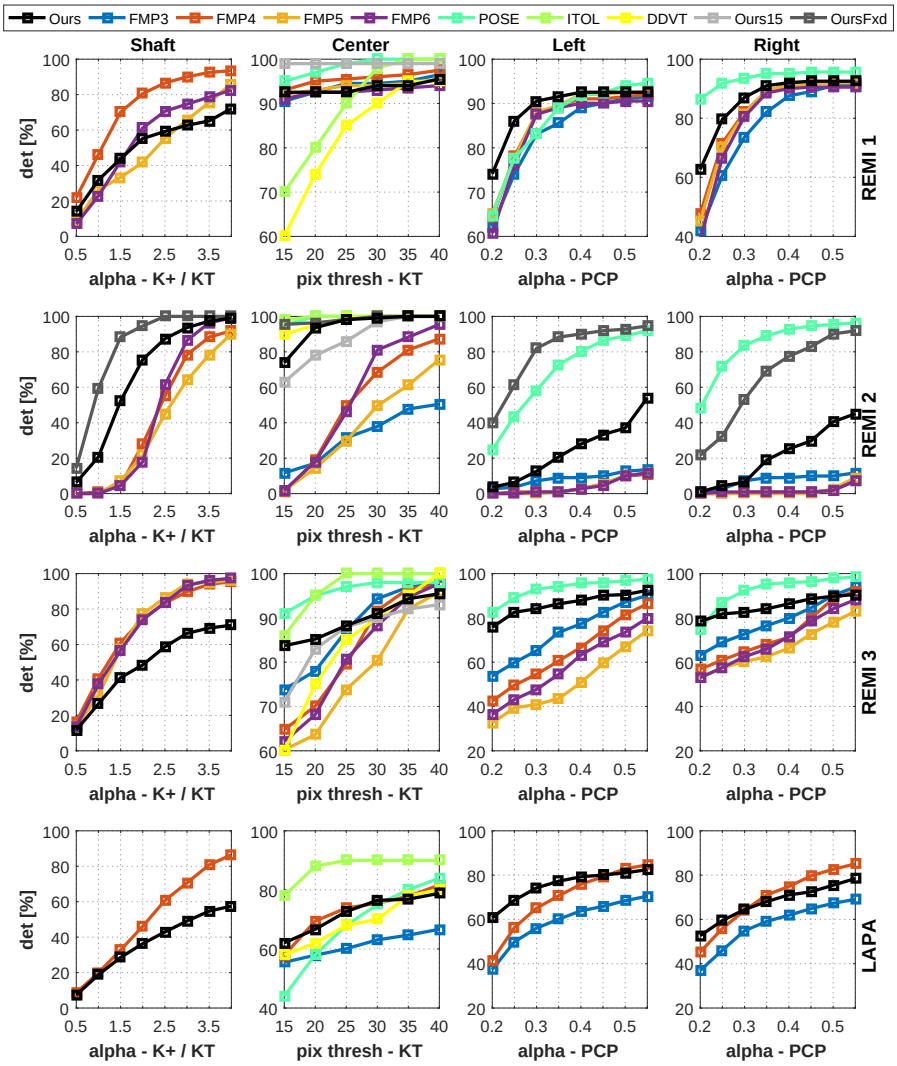

Figure 6: Recall performance of our method and of the state-of-the-art methods: POSE Rieke et al. (2016), ITOL Li et al. (2014b), DDVT Sznitman et al. (2014), including our method OURS15 from Wesierski et al. (2015) and general object pose estimation method FMP Yang & Ramanan (2013) (best viewed in color). We used metrics KT and strictPCP (here abbreviated to PCP) from Section 5.3 and datasets REMI and LAPA. We tested our method in the task of detection and pose estimation of non-rigid tools, which are composed of shaft, center, and left and right forcep. The quantitative evaluation indicates that our method achieves state-of-the-art results. The FMP models perform well in locating the shaft part but struggle with finding the pose of the end-effector.

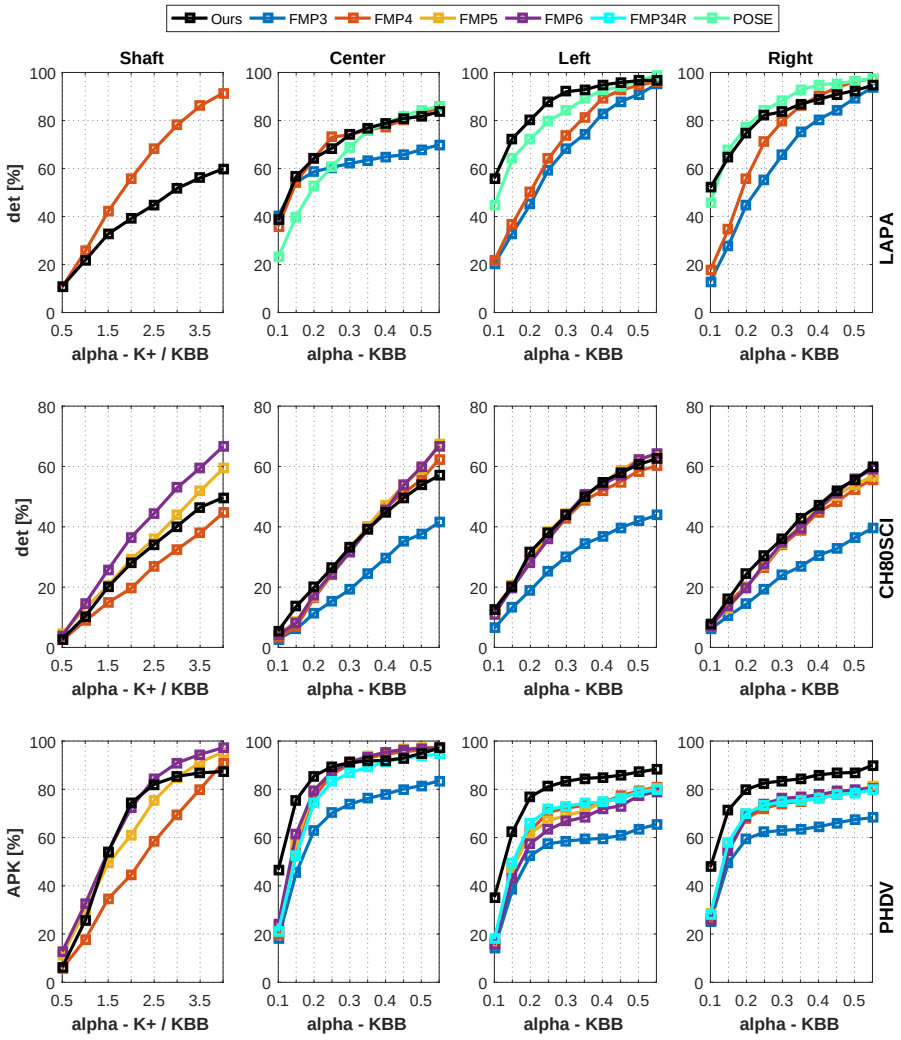

Figure 7: Recall performance of our method and of the state-of-the-art methods: POSE Rieke et al. (2016) and general object pose estimation method FMP Yang & Ramanan (2013) (best viewed in color). We used metrics KBB and APK from Section 5.3 and datasets LAPA, PHDV, and CH80SCI. The quantitative evaluation indicates that our method performs favorably wrt to POSE on the LAPA dataset. Furthermore, it can estimate the pose of non-rigid and multiple robotic instruments consistently better than the FMP models, as shown for the CH80SCI and PHDV datasets.