Real-Time Segmentation of Non-Rigid Surgical Tools based on Deep Learning and Tracking

Luis C. García-Peraza-Herrera[1], Wenqi Li[1], Caspar Gruijthuijsen[4], Alain Devreker[4], George Attilakos[3], Jan Deprest[5], Emmanuel Vander Poorten[4], Danail Stoyanov[2], Tom Vercauteren[1], and Sébastien Ourselin[1]

[1] Translational Imaging Group, CMIC, University College London, UK
[2] Surgical Robot Vision Group, CMIC, University College London, UK
[3] University College London Hospitals, UK
[4] Katholieke Universiteit Leuven, Belgium
[5] Universitair Ziekenhuis Leuven, Belgium

Abstract. Real-time tool segmentation is an essential component in computer-assisted surgical systems. We propose a novel real-time automatic method based on Fully Convolutional Networks (FCN) and optical flow tracking. Our method exploits the ability of deep neural networks to produce accurate segmentations of highly deformable parts along with the high speed of optical flow. Furthermore, the pre-trained FCN can be fine-tuned on a small amount of medical images without the need to hand-craft features. We validated our method using existing and new benchmark datasets, covering both *ex vivo* and *in vivo* real clinical cases where different surgical instruments are employed. Two versions of the method are presented, non-real-time and real-time. The former, using only deep learning, achieves a balanced accuracy of 89.6% on a real clinical dataset, outperforming the (non-real-time) state of the art by 3.8 percentage points. The latter, a combination of deep learning with optical flow tracking, yields an average balanced accuracy of 78.2% across all the validated datasets.

1 Introduction

Tool detection, segmentation and tracking is a core technology that has many potential applications. It may for example be used to increase the context-awareness of surgeons in the operating room [1]. In the context of delicate surgical interventions, such as fetal [2] and ophthalmic surgery [3], providing the clinical operator with accurate real-time information about the surgical tools could be highly valuable and help to avoid human errors. Identifying tools is also part of other computational pipelines such as mosaicking, visual servoing and skills assessment. Image mosaicking can provide reconstructions larger than the image provided by the usual endoscopic view. The mosaic is normally generated by stitching endoscopic images as the endoscope moves across the operating site [4]. However, surgical tools present in the images occlude the surgical scene being reconstructed. Real-time instrument detection and tracking facilitates the localisation of the instruments and the further separation from the underlying tissue,

so that the final mosaic only contains patient's tissue. Another application of tool segmentation is visual servoing of articulated or flexible surgical robots. As the dexterity of the instruments rises [5], it becomes increasingly difficult for the surgeon to understand the shape of these instruments. With the miniaturisation of said instruments, the kinematics of these devices become less deterministic due to effects from friction, hysteresis and backlash alongside with increased instrument compliance and safety. Furthermore, it is challenging to embed position or shape sensing on them without increasing their size. A key advantage of visual tool tracking versus fiducial markers or auxiliary technologies is that there is no need to modify the current workflow or propose alternative exotic instruments. Previous work has addressed detection [6], localisation [7] and pose estimation of instruments [8] using different cues and classification strategies. For example, employing information about the geometry of the instruments [9], fiducial markers [10], 3D coordinates of the insertion point [11], fusing visual and kinematic information [12] and through multi-class pixel-wise classification of colour, texture and position features with different machine learning techniques such as Random Forests (RF) [13] and Boosted Decision Forests [1]. Recent advances in Region-based Convolutional Neural Networks (R-CNN) [14] and Region Proposal Networks (RPN) [15] have enabled the possibility of object detection (with a bounding box) near real-time (17fps for images on Pascal VOC 2007 [16]). EndoNet [17] has been recently proposed as a solution for phase recognition and tool presence detection on laparoscopic videos. However, there is still a need for an automatically initialised real-time (i.e. camera frame rate) segmentation algorithm for non-rigid tools with unknown geometry and kinematics.

There are a number of challenges that need to be addressed for real-time detection and tracking of surgical instruments. Endoscopic images typically present a vast amount of specular reflections (from both tissue and instruments), which is a source of confusion for segmentation algorithms as pixels that look the same belong to different objects (e.g. background and foreground). Changing lighting conditions, shadows and motion blur, combined with the complexity of the scene and the motion of organs in the background are also a challenge, as can be observed in fig. 1. As a result, anatomical structures and surgical instruments may look more similar than they actually are. Occlusions caused by body fluids and smoke also represent a major issue. Particularly for the case of fetal surgery, the turbidity of the amniotic fluid, makes the localisation of instruments really challenging, as can be observed in fig. 1. Fetal surgery also has the additional difficulty of relying on miniature endoscopes that contain several tens of thousands of fibres in an imaging guide. Transformed into pixels the number of fibres results in a very poor resolution (e.g. 30K in a Karl Storz GmbH 11508 AAK curved fetoscope [18]).

To the best of our knowledge, in this paper, we present the first real-time ($\approx$ 30 fps) surgical tool segmentation pipeline. Our pipeline takes monocular video as input and produces a foreground/background segmentation based on both deep learning semantic labelling and optical flow tracking. The method is instrument-agnostic and can be used to segment different types of rigid or

non-rigid instruments. We demonstrate that deep learning semantic labelling outperforms the state of the art on an open neurosurgical clinical dataset [1]. Our results also show competitive performance between real-time and non-real-time implementations of our method.

2 Methods

Convolutional-Neural-Network-based segmentation. There are several benefits of using a Convolutional Neural Network (CNN) compared to other state-of-the-art machine learning approaches [1]. First, there is no need for trial and error to hand-craft features, as features are automatically extracted during the network training phase. As demonstrated in [19], automatic feature selection does not negatively affect the segmentation quality. Furthermore, CNNs can be pre-trained on large general purpose datasets from the Computer Vision community and fine-tuned with a small amount of domain-specific images, as explained in [20]. This particular feature of CNNs allows us to overcome the scarcity of labelled images faced by the CAI community. Therefore, it conveys the possibility of having an instrument segmentation mechanism that is not tool dependent, as demonstrated by our results.

Fully Convolutional Networks (FCN) are a particular type of CNN recently proposed by Long et al. [20]. As opposed to previous CNNs such as AlexNet [21] or VGG16 [22], FCN are tailored to perform semantic labelling rather than classification. However, the two are closely related as FCN are built from adapting and fine-tuning pre-trained classification networks. In order to achieve this conversion from classification to segmentation two key steps are performed. First, the fully connected (FC) layers of the classification network are replaced with convolutions so that spatial information is preserved. Second, *upsampling filters*

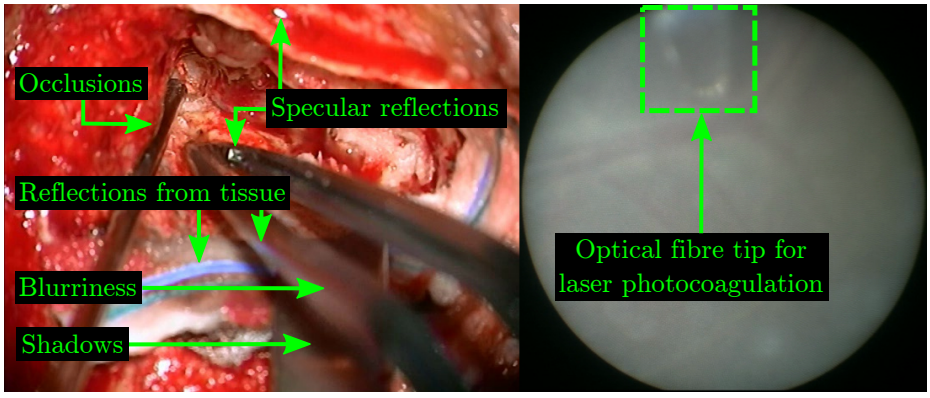

Fig. 1. Challenges encountered by tool detection and localisation algorithms in real interventions. *In vivo* neurosurgery [1] (left). Twin-to-twin transfusion syndrome laser photocoagulation (right).

(also called *deconvolution layers*) are employed to generate a multi-class pixel-level output segmentation that features the same size of the input image. An essential characteristic of the *upsampling filters* present in FCN is that their weights are not fixed, but initialised to perform bilinear interpolation and then learnt during the fine-tuning process. As a consequence, these networks are able to accept an arbitrary-sized input, produce a labelled output of equivalent dimensions and rely on end-to-end learning of labels and locations. That is, they behave as *deep non-linear filters* that perform semantic labelling. There are three versions of the FCN introduced by Long et al., FCN-8s (shown in fig. 2), FCN-16s and FCN-32s (available in the Caffe Model Zoo [23]). The difference between them being the use of intermediate outputs (such as the one coming from POOL_3 or POOL_4 in fig. 2) in order to achieve finer segmentations.

In this work, we have adapted and fine-tuned the FCN-8s [20] for instrument segmentation. Its state-of-the-art performance in multi-class segmentation of general purpose computer vision datasets makes it a sensible choice for the task. The FCN-8s we employed was pre-trained on the PASCAL-context 59-class (60 including background) [24] dataset. As we are concerned with the separation of non-rigid surgical instruments from background, the structure of the network was adapted to provide only two scores per pixel by changing the number of outputs to just *two* in the scoring and upsampling layers. This modification of parameters is highlighted within the dashed line in fig. 3. After this change, the network can be fine-tuned with a small amount of data belonging to a particular surgical domain. During inference, the final per-pixel scores provided by the FCN are normalised and calculated via *argmax* to obtain per-pixel labels.

We have also implemented an improved learning process for the FCN. The optimiser selected to update the weights was the standard *Stochastic Gradient*

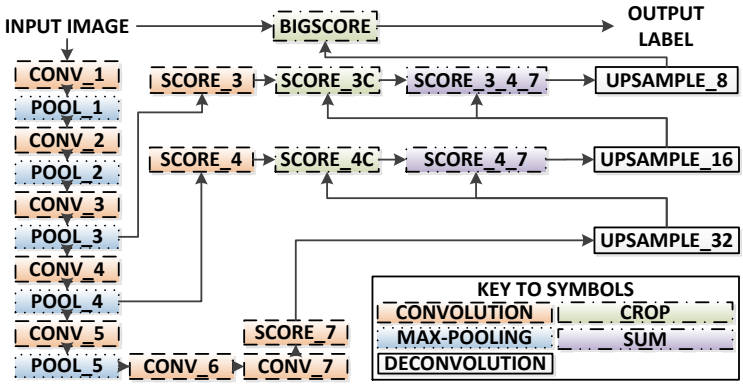

Fig. 2. Illustration of the FCN-8s network architecture, as proposed in [20]. In our method, the architecture of the network remains the same, but the number of outputs in SCORE_3, SCORE_4, SCORE_5, UPSAMPLE_8, UPSAMPLE_16 and UPSAMPLE_32 has been changed so that they produce only two scores per pixel, background and foreground.

Descent (SGD). A key hyper-parameter of the fine-tuning process is the *learning rate* (LR), which is the weight applied to the negative gradient used in the update rule of the optimisation. It has been recently shown in [25] that letting the learning rate fluctuate during the fine-tuning process achieves convergence to a higher accuracy in less number of iterations. This policy, introduced by Smith as Cyclical Learning Rate (CLR) [25], may be implemented with different shapes

Name	CONV_1_1	CONV_1_2	POOL_1	CONV_2_1	CONV_2_2	POOL_2
Type	Convolution	Convolution	Max-pooling	Convolution	Convolution	Max-pooling
Number of filters	64	64	N/A	128	128	N/A
Kernel size	3	3	2	3	3	2
Stride	1	1	2	1	1	2
Activation function	ReLU	ReLU	N/A	ReLU	ReLU	N/A

Name	CONV_3_1	CONV_3_2	CONV_3_3	POOL_3	CONV_4_1	CONV_4_2
Type	Convolution	Convolution	Convolution	Max-pooling	Convolution	Convolution
Number of filters	256	256	256	N/A	512	512
Kernel size	3	3	3	2	3	3
Stride	1	1	1	2	1	1
Activation function	ReLU	ReLU	ReLU	N/A	ReLU	ReLU

Name	CONV_4_3	POOL_4	CONV_5_1	CONV_5_2	CONV_5_3	POOL_5
Type	Convolution	Max-pooling	Convolution	Convolution	Convolution	Max-pooling
Number of filters	512	N/A	512	512	512	N/A
Kernel size	3	2	3	3	3	2
Stride	1	2	1	1	1	2
Activation function	ReLU	N/A	ReLU	ReLU	ReLU	N/A

Name	CONV_6	CONV_7	SCORE_3	SCORE_4	SCORE_7	UPSAMPLE_32
Type	Convolution	Convolution	Convolution	Convolution	Convolution	Deconvolution
Number of filters	4096	4096	**2**	**2**	**2**	**2**
Kernel size	7	1	1	1	1	4
Stride	1	1	1	1	1	2
Activation function	ReLU	ReLU	None	None	None	None

Name	UPSAMPLE_16	UPSAMPLE_8	SCORE_4_7	SCORE_3_4_7	SCORE_3C	SCORE_4C
Type	Deconvolution	Deconvolution	Sum	Sum	Crop	Crop
Number of filters	**2**	**2**	N/A	N/A	N/A	N/A
Kernel size	4	16	N/A	N/A	N/A	N/A
Stride	2	8	N/A	N/A	N/A	N/A
Activation function	None	None	None	None	None	None

Name	BIGSCORE
Type	Crop
Number of filters	N/A
Kernel size	N/A
Stride	N/A
Activation function	None

Fig. 3. Parameters of the adapted FCN. Changes with respect to the original FCN-8s [20] are shown surrounded by a dashed line.

(e.g. triangular, parabolic, sinusoidal). However, all of them produce similar results in [25]. We therefore choose the triangular window for the sake of simplicity. As we are only interested in fine-tuning the network, the LR was constrained to a small value to tailor the parameters to the surgical domain without altering the behaviour of the network. In our case, the LR boundaries, momentum and weight decay were set to [1e-13, 1e-10], 0.99 and 0.0005, respectively.

Real-time segmentation pipeline. The drawback of the FCN we used is that it cannot run in real-time. CAFFE performs forward evaluation in about 100ms for a 500×500 RGB image using an NVIDIA GeForce GTX TITAN X GPU, but this computational time is well below the frame-rate of the endoscopic video, which is generally 25, 30, or 60 fps.

The key insight that was employed here to overcome this problem is that in the short time slot between two FCN segmentations, the tool remains roughly rigid and its appearance changes can be captured sufficiently well by an affine transformation. This type of transformation provides a trade-off between representing small changes and being robust enough for fast fitting purposes. Based on this assumption, tracking is used to detect the small motion between the last FCN-segmented frame and the current one. By registering the last FCN-segmented frame (as opposed to the most recently segmented frame) with the current one, we avoid the time-consuming feature point extraction in every frame and potentially reduce the propagation of error across frames.

Our asynchronous pipeline is illustrated in fig. 4. The FCN segmenter runs asynchronously to the rest of the pipeline. That is, when a frame is read from the video feed, it is sent to the FCN segmenter only if the FCN is not currently busy processing a previous frame. When the FCN finishes a segmentation, it updates the *last* segmentation mask, which is stored in synchronised memory. Furthermore, the image just segmented is converted to grayscale (as matching feature points is faster than in colour images) and stored along with some (maximum 4000) foreground feature points for later use by the optical flow tracker. The feature points used are corners provided by the `GoodFeaturesToTrack` extractor (OPENCV implementation of the Shi-Tomasi corner detector [26]), which in combination with optical flow forms a widely successful tracking framework used for temporal constraints that satisfies our real-time requirement. All the output segmentations are computed according to the following process. First, pyramidal Lukas-Kanade [27] optical flow is employed to find the correspondence between the foreground points in the previous FCN-segmented frame and the current received frame. Then the affine transformation between the two sets of points is estimated by solving the linear least squares problem

$$\boldsymbol{A}^*,\, \mathbf{t}^* := \underset{\boldsymbol{A},\,\mathbf{t}}{\operatorname{argmin}} \left(\sum_{i \,\in\, \text{inliers}} \|\mathbf{n}[i] - \boldsymbol{A}\,\mathbf{p}[i] - \mathbf{t}\|^2 \right)$$

with a RANSAC approach (`estimateRigidTransform`, OPENCV implementation to compute an optimal affine transformation between two 2D point sets) where i is the iterator over the inlier feature-point matches, $\mathbf{p}$ is the set of points

in the last FCN-segmented frame, $\mathbf{n}$ is the set of points in the frame that we are currently trying to segment and $[\boldsymbol{A}|\mathbf{t}]$ is the affine transformation between the two sets of points that we are estimating.

Once the affine transformation is obtained, it is applied to the *last* segmentation mask produced by the FCN. This warped label is the final segmentation for the frame.

3 Experiments and Results

With the aim of demonstrating the flexibility of the presented methodology, three datasets have been used for validation. They contain training and test data for a wide variety of surgical settings, including *in vivo* abdominal and neurological surgery and different set-ups of *ex vivo* robotic surgery. Furthermore, they also contain different surgical instruments, i.e. rigid, articulated and flexible, respectively.

`EndoVisSub` [28]. MICCAI 2015 Endoscopic Vision Challenge - Instrument Segmentation and Tracking Sub-challenge. This dataset consists of two sub-datasets, *robotic* and *non-robotic*. The training data for the *robotic* sub-dataset is formed by four *ex vivo* 45-second videos and the test data is formed by four 15-second and two 60-second videos. All of them having a resolution of 720×576 and 25 fps. The training data for the *non-robotic* sub-dataset is formed by 160 *in vivo* abdominal images (coming from four different sequences) and the test

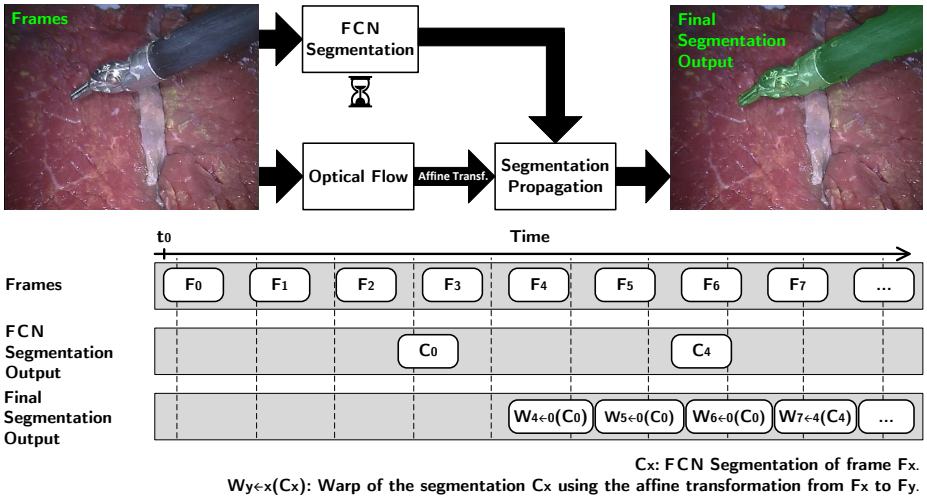

Fig. 4. Real-time segmentation diagram and timeline. For the first few frames no FCN-based segmentation is available, hence the system does not provide any output. As soon as the first FCN output is retrieved, the system provides a segmentation per video frame. All the segmentation outputs $\mathbf{W}$ were obtained based on the last FCN-based output $\mathbf{C}$.

data is formed by 4600 images (coming from nine different sequences). All of them having a resolution of 640×480. No quantitative results are reported for the non-robotic `EndoVisSub` sub-dataset as ground-truth was not available from the challenge website.

`NeuroSurgicalTools` [1]. This dataset consists of 2476 monocular images (1221 for training and 1255 for testing) coming from *in vivo* neurosurgeries. The resolution of the images varies from 612×460 to 1920×1080.

`FetalFlexTool`. *Ex vivo* fetal surgery dataset consisting of 21 images for training and a video sequence of 10 seconds for testing. In both the images and the video a non-rigid McKibben artificial muscle [5] is actuated close to the surface of a human placenta. In order to prove the generalisation capabilities of the method, the training images were captured in air and the video was recorded under water, to facilitate different backgrounds and lighting conditions. The ground truth of both the training images and the testing video was produced through manual segmentation. The *ex vivo* placenta used to generate this dataset was collected following a caesarean section delivery and after obtaining a written informed consent from the mother at University College London Hospitals (UCLH). The Joint UCL/UCLH Committees on Ethics of Human Research approved the study.

We implemented our method in C++, making use of the Caffe-future branch, acceleration from the NVIDIA CUDA Deep Neural Network library v4, using the Intel(R) Math Kernel Library as BLAS choice and the CUDA module of OpenCV 3.1. The results have been generated with an Intel(R) Xeon(R) (CPU) E5-1650 v3 @ 3.50GHz computer and a GeForce GTX TITAN X (GPU). All the results reported were obtained by fine-tuning the FCN for each dataset.

The first experiment carried out analysed the feasibility of FCN-based semantic labelling for instrument segmentation tasks without considerations for real-time requirements. The quantitative results can be seen in table 1 and some segmentation examples are shown in fig. 5 and the supplementary material. As can be seen in table 1, the balanced accuracy = (sensitivity + specificity) / 2 achieved for the *in vivo* `NeuroSurgicalTools` dataset is 89.6%, which is higher than the 85.8% reported by [1].

The real-time pipeline, including the mask propagation based on optical flow, was evaluated on `EndoVisSub (robotic)` and `FetalFlexTool` (no real-time results are reported for `NeuroSurgicalTools` due to lack of frame-by-frame video

Table 1. Non-real-time quantitative results of the FCN-based segmentations. The results have been calculated based on the semantic labelling obtained for the testing images of each dataset. Three different FCN (one per dataset) have been fine-tuned to obtain these results.

Dataset	Sensitivity	Specificity	Balanced Accuracy
`EndoVisSub (robotic)`	72.2%	95.2%	83.7%
`NeuroSurgicalTools`	82.0%	97.2%	89.6%
`FetalFlexTool`	84.6%	99.9%	92.3%

ground-truth). Quantitative results can be seen in table 2. The real-time pipeline captures the tool with a performance which is acceptable in comparison to the off-line counterpart, as illustrated in fig. 6 and the supplementary material. Our method was able to produce real-time ($\approx$ 30Hz) results for all the datasets.

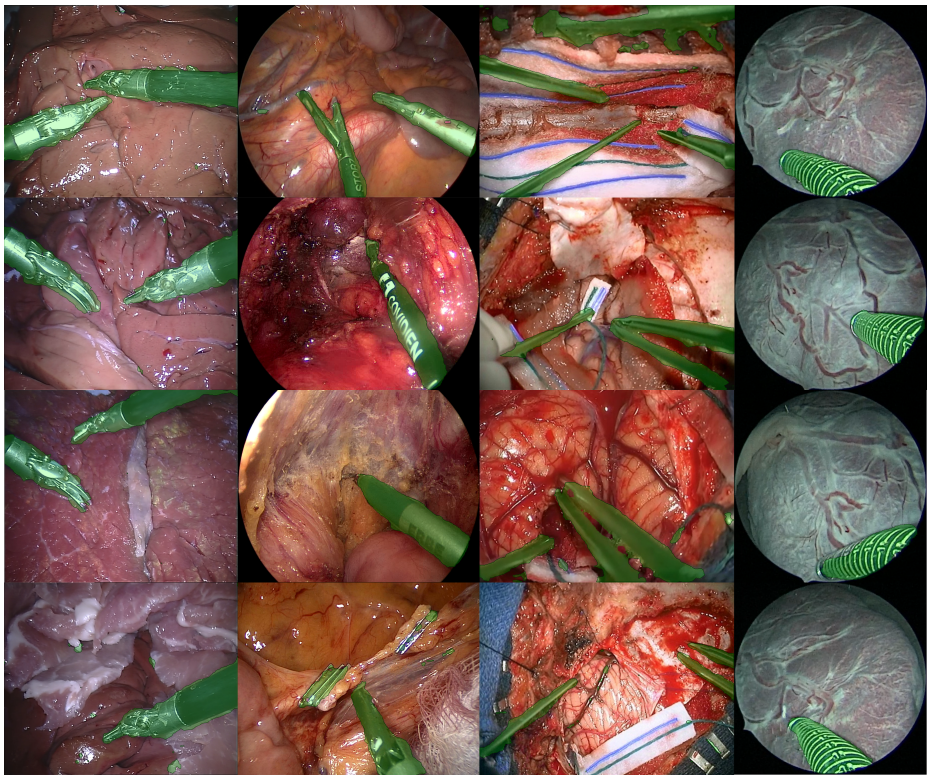

Fig. 5. FCN-based segmentation of four testing images, each one belonging to a different dataset. From left to right, `EndoVisSub (robotic)`, `EndoVisSub (non-robotic)`, `NeuroSurgicalTools` (see [1] Fig.5 for a qualitative comparison) and `FetalFlexTool`.

Fig. 6. Comparison between FCN-based segmentation and tracking-based propagation. From left to right, previous frame segmented with FCN (C_x), current frame segmented with FCN (C_y) and tracking-based propagation ($W_{y \leftarrow x}(C_x)$).

4 Discussion and Conclusion

FCN stand out as a very promising technology for labelling endoscopic images. They can be fine-tuned with a small amount of medical images and no discriminative features have to be hand-crafted. Furthermore, these advantages are not at the expense of lowering the segmentation performance.

To the best of our knowledge this paper presents the first real-time FCN-based surgical tool labelling framework. Optical flow tracking can be successfully employed to propagate FCN segmentations in real-time. However, the quality of the results depends on how deformable the instruments being segmented are and how fast they move, as can be observed in the different results reported in table 2. The balanced accuracy achieved by the FCN-based labelling of the `EndoVisSub (robotic)` dataset (83.7%) is lower than the one achieved by the real-time version (88.3%). The increase in balanced accuracy from the FCN-based segmentation to the real-time version for the `EndoVisSub` is at the expense of a reduction in specificity. This is due to an inflation of the warped segmentation and related to the fact that several tools are present in the foreground and move in different directions. This may benefit the accuracy score by increasing sensitivity, similar effects have been observed for anchor box trackers (votchallenge.net). For the `FetalFlexTool` dataset which consists of a flexible McKibben actuator the balanced accuracy was reduced from 92.3% to 68.1%.

According to the results reported for the different datasets, we can conclude that the presented methodology is flexible enough to easily adapt to different clinical scenarios. Furthermore, feasibility for real-time segmentation of different surgical instruments has been demonstrated. This including non-rigid tools, as it is the case in the `FetalFlexTool` dataset.

However, as it would be expected, non-rigid foreground movements (either caused by the presence of several instruments or due to genuine non-rigid tool movements) that are faster than the time elapsed between two FCN segmentations (typically 100ms) affect the segmentation quality and will not be captured as well. This could be further addressed by separating the feature points detected on the foreground in different groups and using a set of affine transformations rather than a single one for the whole foreground.

Future work includes the possibility of detecting multiple instruments and also the inclusion of a Tracking Learning Detection framework [29]. At this stage, temporal information of previous segmentations is not fed to the FCN but is only

Table 2. Quantitative results of the full real-time segmentation pipeline. The reported numbers are based on the frame-by-frame comparison of the binary labels provided by the presented real-time method and the ground truth video segmentations (for those datasets which have it).

Dataset	Sensitivity	Specificity	Balanced Accuracy
`EndoVisSub (robotic)`	87.8%	88.7%	88.3%
`FetalFlexTool`	36.3%	99.9%	68.1%

used by the tracking system. It would be interesting to use long-term tracking information to both speed-up and improve the segmentation results.

Acknowledgements. This work was supported by Wellcome Trust [WT101957], EPSRC (NS/A000027/1, EP/H046410/1, EP/J020990/1, EP/K005278), NIHR BRC UCLH/UCL High Impact Initiative and a UCL EPSRC CDT Scholarship Award (EP/L016478/1). The authors would like to thank NVIDIA for the donated GeForce GTX TITAN X GPU, their colleagues E. Maneas, S. Moriconi, F. Chadebecq, M. Ebner and S. Nousias for the ground truth of `FetalFlexTool` and E. Maneas for preparing setup with an *ex vivo* placenta.

References

1. Bouget, D., Benenson, R., Omran, M., Riffaud, L., Schiele, B., Jannin, P.: Detecting Surgical Tools by Modelling Local Appearance and Global Shape. IEEE Transactions on Medical Imaging **34**(12) (2015) 2603–2617
2. Daga, P., Chadebecq, F., Shakir, D., Garcia-Peraza Herrera, L.C., Tella, M., Dwyer, G., David, A.L., Deprest, J., Stoyanov, D., Vercauteren, T., Ourselin, S.: Real-time mosaicing of fetoscopic videos using SIFT. In: SPIE Medical Imaging. (2015)
3. Sznitman, R., Ali, K., Richa, R., Taylor, R.H., Hager, G.D., Fual, P.: Data-driven visual tracking in retinal microsurgery. MICCAI (2012) 568–75
4. Tella, M., Daga, P., Chadebecq, F., Thompson, S., Shakir, D., Dwyer, G., Wimalasundera, R., Deprest, J., Stoyanov, D., Vercauteren, T., Ourselin, S.: A combined EM and visual tracking probabilistic model for robust mosaicking of fetoscopic videos. In: IWBIR. (2016)
5. Devreker, A., Rosa, B., Desjardins, A., Alles, E., Garcia-Peraza, L., Maneas, E., Stoyanov, D., David, A., Vercauteren, T., Deprest, J., Ourselin, S., Reynaerts, D., Vander Poorten, E.: Fluidic actuation for intra-operative in situ imaging. In: IROS, IEEE (2015) 1415–1421
6. Reiter, A., Allen, P.K., Zhao, T.: Marker-less Articulated Surgical Tool Detection. CARS (2012)
7. Allan, M., Ourselin, S., Thompson, S., Hawkes, D.J., Kelly, J., Stoyanov, D.: Toward Detection and Localization of Instruments in Minimally Invasive Surgery. IEEE Transactions on Biomedical Engineering **60**(4) (apr 2013) 1050–1058
8. Allan, M., Thompson, S., Clarkson, M.J., Ourselin, S., Hawkes, D.J., Kelly, J., Stoyanov, D.: 2D-3D Pose Tracking of Rigid Instruments in Minimally Invasive Surgery. In: IPCAI. (2014)
9. Pezzementi, Z., Voros, S., Hager, G.D.: Articulated object tracking by rendering consistent appearance parts. In: ICRA, IEEE (2009) 3940–3947
10. Reiter, A., Goldman, R.E., Bajo, A., Iliopoulos, K., Simaan, N., Allen, P.K.: A learning algorithm for visual pose estimation of continuum robots. In: IROS, IEEE (sep 2011) 2390–2396
11. Voros, S., Orvain, E., Cinquin, P., Long, J.A.: 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. Volume 2006., IEEE (2006) 1107–1112

12. Reiter, a., Allen, P.K., Zhao, T.: Appearance learning for 3D tracking of robotic surgical tools. The International Journal of Robotics Research **33**(2) (feb 2014) 342–356
13. Allan, M., Ourselin, S., Thompson, S., Hawkes, D.J., Kelly, J., Stoyanov, D.: Toward detection and localization of instruments in minimally invasive surgery. IEEE Transactions on Biomedical Engineering **60**(4) (2013) 1050–1058
14. Girshick, R.: Fast R-CNN. ICCV (2015) 1440–1448
15. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. (2015) 1–9
16. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL VOC Challenge 2007 Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
17. Twinanda, A.P., Shehata, S., Mutter, D., Marescaux, J., de Mathelin, M., Padoy, N.: EndoNet: A Deep Architecture for Recognition Tasks on Laparoscopic Videos. In: CVPR. (2016) 1–10
18. Fetoscope: https://www.karlstorz.com/doc/interactivebrochure/3317862/html5
19. Noh, H., Hong, S., Han, B.: Learning Deconvolution Network for Semantic Segmentation. In: ICCV. (2015) 1520–1528
20. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR, IEEE (2015) 3431–3440
21. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. In: NIPS. (2012) 1097–1105
22. Guerra, E., de Lara, J., Malizia, A., Díaz, P.: Supporting user-oriented analysis for multi-view domain-specific visual languages. Information and Software Technology **51**(4) (2009) 769–784
23. Caffe Model Zoo: http://github.com/BVLC/caffe/wiki/Model-Zoo
24. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The Role of Context for Object Detection and Semantic Segmentation in the Wild. In: CVPR. (2014)
25. Smith, L.N.: No More Pesky Learning Rate Guessing Games. Arxiv (jun 2015)
26. Jianbo Shi, Tomasi: Good features to track. In: CVPR, IEEE Comput. Soc. Press (1994) 593–600
27. Bouguet, J.Y.: Pyramidal Implementation of the Lucas Kanade Feature Tracker: Description of the Algorithm. Technical report, Intel Corporation Microprocessor Research Labs (2000)
28. MICCAI 2015 Endoscopic Vision Challenge Instrument Segmentation and Tracking Sub-challenge: http://endovissub-instrument.grand-challenge.org
29. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-Learning-Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(7) (2012) 1409–1422