Simultaneous Recognition and Pose Estimation of Instruments in Minimally Invasive Surgery

Thomas Kurmann[1](✉), Pablo Marquez Neila[2], Xiaofei Du[3], Pascal Fua[2], Danail Stoyanov[3], Sebastian Wolf[4], and Raphael Sznitman[1]

[1] University of Bern, Bern, Switzerland
thomas.kurmann@artorg.unibe.ch
[2] École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
[3] University College London, London, UK
[4] University Hospital of Bern, Bern, Switzerland

Abstract. Detection of surgical instruments plays a key role in ensuring patient safety in minimally invasive surgery. In this paper, we present a novel method for 2D vision-based recognition and pose estimation of surgical instruments that generalizes to different surgical applications. At its core, we propose a novel scene model in order to simultaneously recognize multiple instruments as well as their parts. We use a Convolutional Neural Network architecture to embody our model and show that the cross-entropy loss is well suited to optimize its parameters which can be trained in an end-to-end fashion. An additional advantage of our approach is that instrument detection at test time is achieved while avoiding the need for scale-dependent sliding window evaluation. This allows our approach to be relatively parameter free at test time and shows good performance for both instrument detection and tracking. We show that our approach surpasses state-of-the-art results on *in-vivo* retinal microsurgery image data, as well as *ex-vivo* laparoscopic sequences.

1 Introduction

Vision-based detection of surgical instruments in both minimally invasive surgery and microsurgery has gained increasing popularity in the last decade. This is largely due to the potential it holds for more accurate guidance of surgical robots such as the da Vinci®(Intuitive Surgical, USA) and Preceyes (Netherlands), as well as for directing imaging technology such as endoscopes [1] or OCT imaging [2] at manipulated regions of the workspace.

In recent years, a large number of methods have been proposed to either track instruments over time or detect them without any prior temporal information, in both 2D and 3D. In this work, we focus on 2D detection of surgical instruments as it is often required for tracking in both 2D [3] and 3D [4]. In this context, [5,6] proposed to build ensemble-based classifiers using hand-crafted features to detect instruments parts (*e.g.* shaft, tips or center). Similarly, [7] detected multiple instruments in neurosurgery by repeatedly evaluating a boosted classifier based on semantic segmentation.

© Springer International Publishing AG 2017
M. Descoteaux et al. (Eds.): MICCAI 2017, Part II, LNCS 10434, pp. 505–513, 2017.
DOI: 10.1007/978-3-319-66185-8_57

Yet for most methods described above two important limitations arise. The first is that instrument detection and pose estimation (*i.e.* instrument position, orientation and location of parts) have been tackled in two phases, leading to complicated pipelines that are sensitive to parameter tuning. The second is that at evaluation time, detection of instruments has been achieved by repeated window sliding at limited scales which is both inefficient and error prone (*e.g.* small or very large instruments are missed). Both points heavily reduce the usability of proposed methods.

In order to overcome these limitations, we propose a novel framework that avoids these and can be applied to a variety of surgical settings. Assuming a known maximum number of instruments and parts that could appear in the field of view, our approach, which relies on recent deep learning strategies [8], avoids the need for window sliding at test time and estimates multiple instruments and their pose simultaneously. This is achieved by designing a novel Convolutional Neural Network (CNN) architecture that explicitly models object parts and the different objects that may be present. We show that when combined with a cross-entropy loss function, our model can be trained in an end-to-end fashion, thus bypassing the need for traditional two-stage detection and pose estimation. We validate our approach on both *ex-vivo* laparoscopy images and on *in-vivo* retinal microsurgery, where we show improved results over existing detection and tracking methods.

2 Multi-instrument Detector

In order to detect multiple instruments and their parts in a coherent and simple manner, we propose a scene model which assumes that we know what would be the maximum number of instruments in the field of view. We use a CNN to embody this model and use the cross-entropy to learn effective parameters using a training set. Our CNN architecture takes as input an image and provides binary outputs as to whether or not a given instrument is present as well as 2D location estimates for its parts. A visualization of our proposed detection framework can be seen in Fig. 1. Conveniently then, detecting instruments and estimating the joint positions on a test frame is simply achieved by a feed forward pass of the network. We now describe our scene model and our CNN in more detail.

2.1 Scene Model

Let $I \in \mathbb{R}^{w \times h}$ be an image that may contain up to M instruments. In particular, we denote $T = \{T_1, \ldots, T_M\}, T_m \in \{0, 1\}$ to be the set of instruments that could appear in the field of view such that $T_m = 0$ if the tool is not present and $T_m = 1$ if it is. In addition, each instrument present in the image is defined as a set of N parts, or *joints*, $\{J_m^n \in \mathbb{R}^2\}_{n=0}^N$ consisting of 2D image locations. Furthermore, let $GT_m^n \in \mathbb{R}^2$ be the ground truth 2D position for joint n of instrument T_m and $t_m \in \{0, 1\}$ be the ground truth variable indicating if the mth instrument is visible in the image. Assuming that the instrument presence is unknown and is

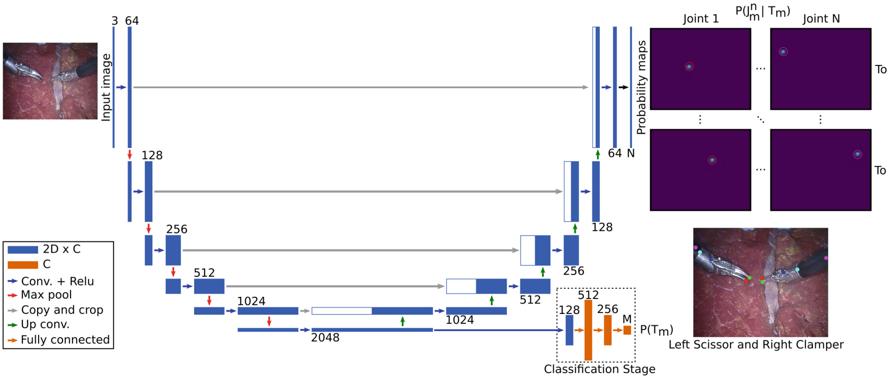

Fig. 1. Proposed multi-instrument detector network architecture. The network produces probabilistic outputs for both the presence of different instruments and position of their joints. The number of channels C is denoted on top of the box.

probabilistic in nature, our goal is to train a network to estimate the following scene model

$$P(T_1,\ldots,T_M,J_1^1,\ldots,J_1^N,\ldots,J_M^1,\ldots,J_M^N)=\prod^{m}P(T_m)\prod^{m}\prod^{n}P(J_m^n|T_m) \quad (1)$$

where $P(T_m)$ are Bernoulli random variables and the likelihood models $P(J_m^n|T_m)$ are parametric probability distributions. Note that Eq. (1) assumes independence between the different instruments as well as a conditional independence between the various joints for a given instrument. Even though both assumptions are quite strong, they provide a convenient decomposition and a model simplification of what would otherwise be a complicated distribution. Letting P be the predicted distribution by our CNN and $\hat{P}$ be a probabilistic interpretation of the ground truth, then the cross-entropy loss function can be defined as

$$H(\hat{P},P)=-\sum_{s\in\mathcal{S}}\hat{P}(s)\log P(s) \quad (2)$$

where $\mathcal{S}$ is the probability space over all random variables $(T_1,\ldots,T_M,\ J_1^1,\ \ldots,\ J_1^N,\ldots,J_M^1,\ldots,J_M^N)$. Replacing P and $\hat{P}$ in Eq. (2) with the model Eq. (1) and simplifying the term gives rise to

$$\begin{aligned}H(\hat{P},P)=\sum_m H\left(\hat{P}(T_m),P(T_m)\right)+\\ \sum_m\sum_n H\left(\hat{P}(J_m^n|T_m=t_m),P(J_m^n|T_m=t_m)\right)\end{aligned} \quad (3)$$

To model the ground truth distribution $\hat{P}$, we let $\hat{P}(T_m)=0$ if $t_m=0$ and $\hat{P}(T_m)=1$ if $t_m=1$, and specify the following likelihood models from the ground truth annotations,

$$\forall_n \forall_m,\ \hat{P}(J_m^n = j | T_m = t_m) = \begin{cases} \mathcal{U}(j; 0, wh), & \text{if } t_m = 0 \\ \mathcal{G}(j; GT_m^n, \sigma^2 \mathbb{I}), & \text{if } t_m = 1 \end{cases}$$

where $\mathcal{U}$ is a Uniform distribution in the interval 0 to wh and $\mathcal{G}$ denotes a Gaussian distribution with mean GT_m^n and covariance $\sigma^2 \mathbb{I}$ (*i.e.* assuming a symmetric and diagonal covariance matrix). We use this Gaussian distribution to account for the inaccuracies in the ground truth annotations such that $\hat{P}(J_m^n | T_m = t_m)$ is a 2D probability map generated from the ground truth and which the network will try to estimate by optimizing Eq. (3). In this work, we fix $\sigma^2 = 10$ for all experiments. That is, our network will optimize both the binary cross-entropy loss of each of the instruments as well as the sum of the pixel-wise probability map cross-entropy losses.

2.2 Multi-instrument Detector Network

In order to provide a suitable network with the loss function of Eq. (3), we modify and extend the U-Net [8] architecture originally used for semantic segmentation. Illustrated in Fig. 1, the architecture uses down and up sampling stages, where each stage has a convolutional, a ReLU activation and a sampling layer. Here we use a total of 5 down and 5 up sampling stages and a single convolutional layer is used per stage to reduce the computational requirements. The number of features is doubled (down) or halved (up) per stage, starting with 64 features in the first convolutional layer. All convolutional kernels have a size of 3×3, except for the last layer where a 1×1 kernel is used. Batch normalization [9] is applied before every activation layer. In order to provide output estimates $\forall(m, n), P(T_m), P(J_m^n | T_m)$, we extend this architecture to do two things:

1. We create classification layers stemming from the lowest layer of the network by expanding it with a fully connected classification stage. The expansion is connected to the lowest layer in the network such that this layer learns to spatially encode the instruments. In particular this layer has one output per instrument which is activated with a sigmoid activation function to force a probabilistic output range. By doing so, we are effectively making the network provide estimates $P(T_m)$.
2. Our network produces $M \times N$ maps of size $w \times h$ which correspond to each of the $P(J_m^n | T_m = 1)$ likelihood distributions. Note that explicitly outputting $P(J_m^n | T_m = 0)$ is unnecessary. Each output probability map $P(J_m^n | T_m = 1)$ is normalized using a softmax function such that the joint position estimate of GT_m^n is equal to the arg $\max_z P(J_m^n = z | T_m = 1)$.

When combined with the loss function Eq. (3), this network will train to both detect multiple instruments as well as estimate their joint parts. We implemented this network using the open source TensorFlow library [10] in Python[1].

[1] Code and models available at: https://github.com/otl-artorg/instrument-pose.

3 Experiments

Retinal Microsurgery. We first evaluate our approach on the publicly available *in-vivo* retinal microsurgery instrument dataset [11]. The set contains 3 video sequences with 1171 images, each with a resolution of 640 × 480 pixels. Each image contains a single instrument with 4 annotated joints (start shaft, end shaft, left tip and right tip). As in [11], we trained our network on the first 50% of all three sequences and evaluated the rest. Optimization of the network was performed with the Adam optimizer [12] using a batch size of 2 and an initial learning rate of 10^{-4}. The network was trained for 10 epochs. Training and testing was performed on a Nvidia GTX 1080 GPU running at an inference rate of approximately 9 FPS.

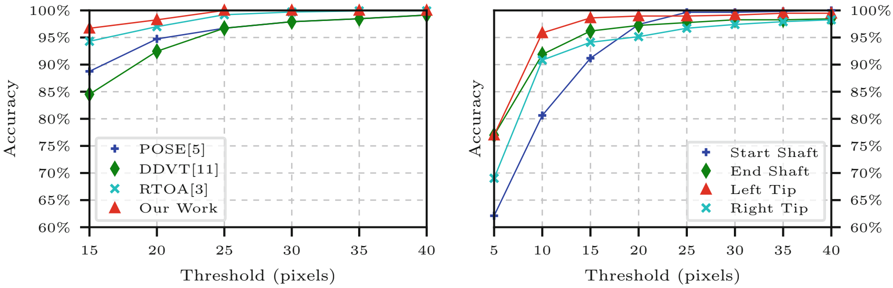

Fig. 2. Detection accuracy. (left) percentage of correctly detected end of shaft joints as a function of the accuracy threshold. (right) percentage of correctly detected joints.

The network was trained on three joints (left tip, right tip and end shaft) while only the end shaft joint was evaluated. Similar to [3,11,13], we show the proportion of frames where the end shaft is correctly identified as a function of detection sensitivity. We show the performance of our approach as well as state-of-the-art detection and tracking methods in Fig. 2. Our method achieves an accuracy of 96.7% at a threshold radius of 15 pixels which outperforms the state-of-the-art of 94.3%. The other two joints (left tip, right tip) achieve an accuracy of 98.3% and 95.3%, showing that the method is capable of learning all joint positions together with a high accuracy. The mean joint position errors are 5.1, 4.6 and 5.5 pixels. As the dataset includes 4 annotated joints, we propose to also evaluate the performance for all joints and report in Fig. 2 (right) the accuracy of the joints after the network was trained with all joints using the same train-test data split. Overall, the performance is slightly lower than when training and evaluating with 3 joints because the 4th joint is the most difficult to detect due to blur and image noise. Figure 3 depicts qualitative results of our approach and a video of all results can be found at https://www.youtube.com/watch?v=ZigYQbGHQus.

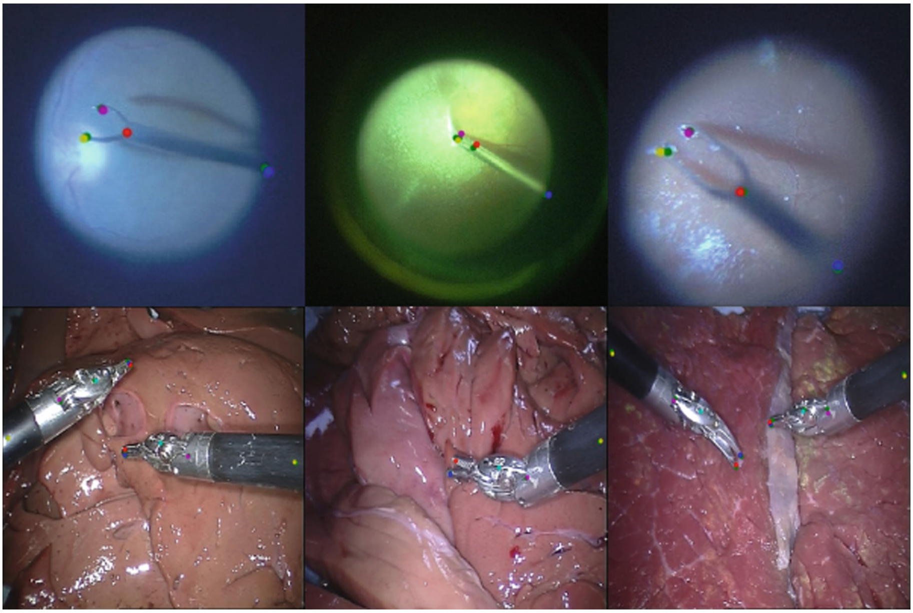

Fig. 3. Visual results on retinal microsurgery image sequences 1–3 (*top*) and laparoscopy sequences (*bottom*). The first two laparoscopy sequences contain claspers, whereas the right most contains a scissor and a clasper. The ground truths are denoted with green points.

Robotic Laparoscopy. We also evaluated our approach on the MICCAI 2015 endoscopic vision challenge for laparoscopy instrument dataset tracking[2]. The dataset includes 4 training and 6 testing video sequences. In total 3 different tools are visible in the sequences: left clasper, right clasper and left scissor which is only visible in the test set. The challenge data only includes a single annotated joint (extracted from the operating da Vinci®robot) which is inaccurate in a large number of cases. For this reason, 5 joints (left tip, right tip, shaft point, end point, head point) per instrument in each image were manually labeled and then used instead[3]. Images were resized to 640 × 512 pixels due to memory constraints when training the network. The training set consists of 940 images and the test set of 910 images. Presence of tools T_m is given if a single joint is annotated. We define the instruments $T_{1...4}$ as left clasper, right clasper, left scissor and right scissor. To evaluate our approach, we propose two experiments: (1) Uses the same training and test data as in the original challenge, with an unknown tool in the test set. (2) We modified the training and test sets, such that the left scissor is also available during training by moving sequence 6 of the test set to the training set. By flipping the images in this sequence left-to-right, we

[2] https://endovissub-instrument.grand-challenge.org/.

[3] https://github.com/surgical-vision/EndoVisPoseAnnotation.

augment our training data so to have the right scissor as well. Not only does this increase the complexity of the detection problem, but it also allows flipping data augmentation to be used.

Experiment 1. Using the original dataset, we first verified that the network can detect specific tools. As the left scissor has not been trained on, we expect this tool to be missed. The training set was augmented using left-right and up-down flips. On the test set, only two images were wrongly classified, with an average detection rate of 99.9% (right clasper 100%, left clasper 99.89%). Evaluation of the joint prediction accuracy was performed as with the microsurgery dataset and the results are illustrated in Fig. 4 (left). The accuracy is over 90% at 15 pixels sensitivity on all joints except for the two tips on the left clasper. The lower performance is explained by the left clasper only being visible in 40 frames, and to that the method fails on 7 images where the tool tips of both the left and right clasper are in the vicinity of each other or overlapping.

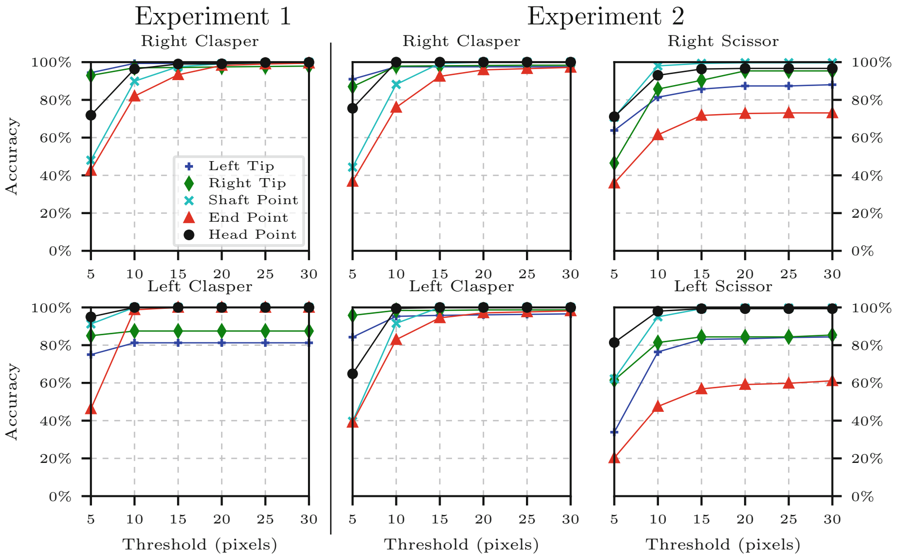

Fig. 4. Accuracy threshold curves: left *Experiment 1* and right *Experiment 2*

Experiment 2. Here the dataset was modified so that the right scissor is also visible in the training set by placing sequence 6 from the test set into the training set. The classification results of the instruments are: right clasper 100%, left clasper 100%, right scissor 99.78% and left scissor 99.67%. Figure 4 (right) shows the results of all joint accuracies for this experiment. The accuracy of the left clasper tool is slightly improved compared to the previous experiment due to the increased augmented training size. However, the method still fails on the same images as in *Experiment 1*. The scissors show similar results for both left and right, which is to be expected due to them being from the same flipped images. Further, for the scissor results it is visible that one joint performs poorer than the rest. Upon visual inspection, we associate this performance drop to the inconsistency in our annotations and the joint not being visible in certain images. Given that our method assumes all joints are visible if a tool is present, detection failures occur when joints are occluded. Due to the increased input image size compared to the retinal microsurgery experiments, the inference rate is lower at around 6 FPS using the same hardware.

4 Conclusion

We presented a deep learning based surgical instrument detector. The network collectively estimates joint positions and instrument presence using a combined loss function. Furthermore, the network obtains all predictions using a single feed-forward pass. We validated the method on two datasets, an *in-vivo* retinal microsurgery dataset and an *ex-vivo* laparoscopy set. Evaluations on the retinal microsurgery dataset showed state-of-the-art performance, outperforming even the current tracking methods. Our detector method is uninfluenced by previous estimations which is a key advantage over tracking solutions. The laparoscopy dataset showed that the method is capable of classifying instrument presence with a very high accuracy while jointly estimating the position of 20 joints. This points to our method being able to simultaneously count, estimate joint locations and classify whether instruments are visible in a single feed-forward pass.

References

1. Wolf, R., Duchateau, J., Cinquin, P., Voros, S.: 3D tracking of laparoscopic instruments using statistical and geometric modeling. In: Fichtinger, G., Martel, A., Peters, T. (eds.) MICCAI 2011. LNCS, vol. 6891, pp. 203–210. Springer, Heidelberg (2011). doi:10.1007/978-3-642-23623-5_26
2. Alsheakhali, M., Eslami, A., Roodaki, H., Navab, N.: CRF-based model for instrument detection and pose estimation in retinal microsurgery. Comput. Math. Methods Med. **2016**, 10 p. (2016). Article ID 1067509. doi:10.1155/2016/1067509. https://www.hindawi.com/journals/cmmm/2016/1067509/cta/
3. Rieke, N., Tan, D.J., Tombari, F., Vizcaíno, J.P., di San Filippo, C.A., Eslami, A., Navab, N.: Real-time online adaption for robust instrument tracking and pose estimation. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9900, pp. 422–430. Springer, Cham (2016). doi:10.1007/978-3-319-46720-7_49
4. Du, X., Allan, M., Dore, A., Ourselin, S., Hawkes, D., Kelly, J.D., Stoyanov, D.: Combined 2d and 3d tracking of surgical instruments for minimally invasive and robotic-assisted surgery. IJCARS **6**, 1109–1119 (2016)
5. Reiter, A., Allen, P.K., Zhao, T.: Feature classification for tracking articulated surgical tools. In: Ayache, N., Delingette, H., Golland, P., Mori, K. (eds.) MICCAI 2012. LNCS, vol. 7511, pp. 592–600. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33418-4_73

6. Sznitman, R., Becker, C., Fua, P.: Fast part-based classification for instrument detection in minimally invasive surgery. In: Golland, P., Hata, N., Barillot, C., Hornegger, J., Howe, R. (eds.) MICCAI 2014. LNCS, vol. 8674, pp. 692–699. Springer, Cham (2014). doi:10.1007/978-3-319-10470-6_86
7. Bouget, D., Benenson, R., Omran, M., Riffaud, L., Schiele, B., Jannin, P.: Detecting surgical tools by modelling local appearance and global shape. IEEE Trans. Med. Imaging **34**(12), 2603–2617 (2015)
8. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). doi:10.1007/978-3-319-24574-4_28
9. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint (2015). arXiv:1502.03167
10. Abadi, M., et al.: TensorFlow: large-scale machine learning on heterogeneous distributed systems (2015)
11. Sznitman, R., Ali, K., Richa, R., Taylor, R.H., Hager, G.D., Fua, P.: Data-driven visual tracking in retinal microsurgery. In: Ayache, N., Delingette, H., Golland, P., Mori, K. (eds.) MICCAI 2012. LNCS, vol. 7511, pp. 568–575. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33418-4_70
12. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR) (2014)
13. Rieke, N., Tan, D.J., Alsheakhali, M., Tombari, F., di San Filippo, C.A., Belagiannis, V., Eslami, A., Navab, N.: Surgical tool tracking and pose estimation in retinal microsurgery. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9349, pp. 266–273. Springer, Cham (2015). doi:10.1007/978-3-319-24553-9_33