Deeply Learned 2D Tool Pose Estimation for Robot-to-Camera Registration

K.K. Maninis[1], A. Chhatkuli[1], J. Pont-Tuset[1], S. Hecker[1], T. Probst[1], M. Ourak[2], E. Vander Poorten[2], L. Van Gool[1,2]

[1]*Computer Vision Lab, ETH Zürich* [2]*KU Leuven*

kmaninis@vision.ee.ethz.ch

INTRODUCTION

Robot-assisted eye surgery is the central topic of the EU funded project EurEyeCase. Major objectives of the project comprise the development of methodologies to perform two surgical procedures that cannot be easily carried out by human surgeons, namely retinal vein cannulation and retinal membrane peeling. In the proposed assistive system, visual guidance is provided from a camera mounted on the microscope. In order to guide the robot using visual cues, it is necessary to register the camera coordinates to the robot coordinates.

To this end, we propose a framework that estimates the position and the pose of the tool to register the two different coordinate systems. Using recent advances in convolutional neural networks (CNNs), we present a comparative study among different intuitive architectural designs, and suggest a methodology to register the coordinates by detecting pre-defined keypoints. Results suggest that tool pose estimation can be highly accurate, running in real-time on a GPU.

MOTIVATION

Visual guidance of a robot from visual cues of a camera requires accurate calibration between the camera and the robot. In the case of retinal surgery, results obtained by post-processing of the imaging system such as segmentation of the veins, and marking of no-go areas, may be used as a feedback to the robot, and thus to the surgeon [1]. To achieve registration of the 3D coordinates of the robot to the 2D coordinates of the camera, one needs to work on the projection equation of a pinhole camera:

$$p = KR^T(P - C) \quad (1)$$

which models the projection of a 3D point $P = \{X, Y, Z\}$ into a pixel $p = \{x, y\}$ of the image, through the camera. K is the intrinsic matrix of the camera. The unknowns of Eq. 1 are the 3D rotation matrix R (3 DoF - Euler angles), and the 3D translation matrix C (3 DoF). To solve this system of 6 unknowns in total, we need 3 pairs of $\{p_i, P_i\}$ correspondences, each of which contributes 2 linearly independent equations. This is in fact a well-studied problem, the solution of which can be obtained through the P3P algorithm [2], once the correspondences are established. This requires the detection of the predefined locations of the tool in each image and its 2D pose.

Recent advances in computer vision employ CNNs to detect keypoints in images for a variety of tasks, the most representative of which is human pose estimation [3, 4]. We adapt such deep learning techniques to establish the correspondences between the robot and the camera, by detecting pre-defined keypoints on the surgical instrument, which is attached to the robot. The advantages of using the surgical instrument is two-fold: (a) The 3D location of the keypoints can be retrieved by the kinematics of the robot (b) during normal use of the instrument, the corresponding 2D locations are always observable by the camera.

DATA GATHERING AND LABELING

To train supervised deep learning algorithms we need to manually annotate our dataset. We selected frames from 8 video sequences acquired by the microscope. The labeled database consists of 400 annotated frames, each annotated with 3 distinguishable keypoints of the tool. The instrument is a grasper designed for membrane peeling. The tooltips and the junction (grasper joints) are selected as the keypoints. We use 6 of the video sequences to train the CNN models, and the remaining 2 to validate the results. Extensive data augmentation was used to prevent overfitting. All architectures are trained from scratch, with randomly initialized weights.

ARCHITECTURAL DESIGNS

There have been numerous different CNN designs proposed in the literature for various tasks. We examine them separately to select the one most suited for the task. In Figure 1, a combination of such architectures is illustrated.

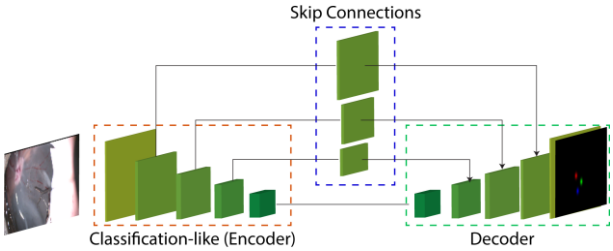

Figure 1: Different architectural designs for Convolutional Neural Networks.

Classification-like: The classification-like architecture was originally used for large scale image recognition, and consists of successive steps of convolutions, ReLU non-linearities, and max-pooling layers. The pooling layers are used to downsample, and produce invariant, semantically rich features in the deeper layers of the CNNs. The downside of such architectures is that they cannot be directly used for tasks that require spatially variant output, like pose estimation, or segmentation. The reason is that it is difficult to recover from the

heavy downsampling, which results in coarse, blobby results (as shown in Figure 2).

Encoder-Decoder: To overcome the effects of downsampling, the encoder-decoder network consists of layers that learn how to gradually upsample the result. In fact, the decoder is added to the head of a classification-like architecture, performing the opposite procedure. The encoder-decoder architecture is able to recover from the coarse scales of the deeper layers.

Skip-Connections: The basic features in the shallow layers of a CNN have proven to be very helpful for many tasks that need detailed output. Skip connections provide a combination of low-level and semantically rich features, usually by concatenation or summation.

Residual blocks: Changing the simple convolutional block to the residual block, proposed in [5], provides a considerable improvement to the results. The idea behind this architecture is that the output G of a layer can be modelled as the sum of its input and a residual, as $G(x) = x + res(x)$. This way, the network only needs to learn the residual, instead of the entire transformation. Indeed, the residual blocks can be trained faster than the simple convolutional ones.

Stacked models: The authors of [3] and [4] proposed an architecture which consists of repetitive downsampling and upsampling phases. This design, referred to as the Stacked Hourglass, takes advantage of successive coarse and fine features, and has been proven very successful in human pose estimation. We observe that these networks lead to the best results in our task.

All architectures were implemented using the PyTorch framework. Training each architecture takes roughly 1 hour, while the inference time is always faster than 30 Hz on a modern GPU.

RESULTS

We evaluate the performance on the validation videos of the EurEyeCase tool pose dataset. For the quantitative evaluation of our study, we use the PCK metric, which measures the accuracy of localization, for each keypoint. In PCK, a detection is correct if it falls in a neighbourhood around the label. The distances are normalized by the distance d of the tooltip to the joint, in each frame. We present results for PCK@0.1, meaning that results that are mislocalized by more than $0.1d$ are considered as false detections.

Results in Table 1 suggest that the classification-like architecture provides inaccurate results for the task, with large errors in pose estimation. The encoder-decoder architecture works a lot better, and provides more accurate results when enriched with skip connections and residual blocks. Finally, by using a stack of modules we observe a further boost in accuracy, reaching 97.2% for the task. This result proves that tool pose estimation can provide accurate correspondences for the desirable 3D-to-2D registration.

Table 1: Quantitative evaluation of keypoint localization, using the PCK@0.1 measure.

Architecture	Top	Middle	Bottom	Mean
Classification-like	08.8	08.8	03.8	07.1
Encoder-Decoder	57.5	72.5	83.8	71.2
+ Skip connections	86.1	86.1	84.7	85.6
+ Residual Blocks	90.0	90.0	87.5	89.2
+ Stacked Modules	**98.6**	**95.8**	**97.2**	**97.2**

Qualitative results are illustrated in Figure 2, for the baseline and the improved architecture. The classification-like architecture shown on the left, seems to suffer from downsampling, leading to highly uncertain results. The improved version, which uses all the aforementioned developments leads to much more certain, and accurate localization of the keypoints.

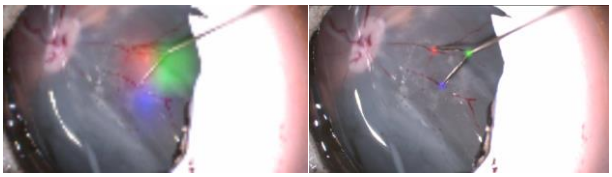

Figure 2: Pose estimation qualitative examples: Left: Classification-like Network, Right: Improved Architecture

CONCLUSIONS AND DISCUSSION

We have presented a study of different CNN architectures to tackle the task of 2D tool pose estimation and localization. Results on our manually annotated dataset show that by using suitable network designs, it is possible to reach almost perfect accuracy for the task. In the future, we would like to reach our end goal of robot-to-camera registration, by using the estimated tool pose.

ACKNOWLEDGEMENTS

This work is funded by the EU H2020 project EurEyeCase – Grant Agreement No 645331.

We thank Georgios Pavlakos and Michal Havlena for all fruitful discussions.

REFERENCES

[1] K.K Maninis, J. Pont-Tuset, P. Arbelaez, L. Van Gool. "Deep Retinal Image Understanding", in MICCAI, 2016

[2] X. Gao, X. Hou, J. Tang, H. Cheng. "Complete Solution Classification for the Perspective-Three-Point Problem". IEEE TPAMI, 2003

[3] A. Newell, K. Yang, J. Deng. "Stacked Hourglass Networks for Human Pose Estimation", in ECCV, 2016

[4] G. Pavlakos, X. Zhou, K. G. Derpanis, K. Daniilidis. "Coarse-to-Fine Volumetric Prediction for Single-Image 3D Human Pose" in CVPR, 2017.

[5] K. He, X. Zhang, S. Ren, J. Sun. "Deep Residual Learning for Image Recognition", in CVPR, 2016