Learning Camera Pose from Optical Colonoscopy Frames Through Deep Convolutional Neural Network (CNN)

Mohammad Ali Armin[1,2(✉)], Nick Barnes[1,4], Jose Alvarez[1], Hongdong Li[4], Florian Grimpen[3], and Olivier Salvado[2]

[1] CSIRO (Data61), Canberra, Australia
m.a.armin@gmail.com, Nick.Barnes@data61.csiro.au, Olivier.Salvado@csiro.au
[2] Biomedical Informatics Group, Brisbane, Australia
[3] Department of Gastroenterology and Hepatology, Royal Brisbane and Women's Hospital, Brisbane, Australia
[4] College of Engineering and Computer Science (ANU), Canberra, Australia

Abstract. Optical colonoscopy is performed by insertion of a long flexible colonoscope into the colon. Estimating the position of the colonoscope tip with respect to the colon surface is important as it would help localization of cancerous polyps for subsequent surgery and facilitate navigation. Knowing camera pose is also essential for 3D automatic scene reconstruction, which could support clinicians inspecting the whole colon surface thereby reducing missed polyps. This paper presents a method to estimate the pose of the colonoscope camera with six degrees of freedom (DoF) using deep convolutional neural network (CNN). Because obtaining a ground truth to train the CNN for camera pose from actual colonoscopy videos is extremely challenging, we trained the CNN using realistic synthetic videos generated with a colonoscopy simulator, which could generate the exact camera pose parameters. We validated the trained CNN on unseen simulated video datasets and on actual colonoscopy videos from 10 patients. Our results showed that the colonoscopy camera pose could be estimated with higher accuracy and speed than feature based computer vision methods such as the classical structure from motion (SfM) pipeline. This paper demonstrates that transfer learning from surgical simulation to actual endoscopic based surgery is a possible approach for deep learning technologies.

Keywords: Optical colonoscopy · Convolutional neural network (CNN) · Camera pose

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-67543-5_5) contains supplementary material, which is available to authorized users.

M.J. Cardoso et al. (Eds.): CARE/CLIP 2017, LNCS 10550, pp. 50–59, 2017.
DOI: 10.1007/978-3-319-67543-5_5

1 Introduction

Colorectal cancer is ranked as the type of cancer that is third most likely to claim people's lives in Australia, and the fourth worldwide [1, 2]. Optical colonoscopy has been known as the gold standard method for detecting and removing colonic polyps, the precursor of bowel cancer [3].

Estimating the colonoscope position with high accuracy is important as it can determine the location of detected polyps, especially when there is a need for subsequent surgery for removing cancerous polyps [4]. Despite the work that has been done in estimating the colonoscope position (camera pose) from optical colonoscopy [5, 6], accurately localizing the position of colonoscope with respect to the colon's surface remains a critical issue.

Conventional methods to estimate camera motion from an endoscopy procedure such as optical flow [5–7] or hybrid methods [8, 9] are time consuming, sensitive to feature matching, require an offline camera or sensor calibration and resulted in a drift in camera pose estimation. Here, we develop a method based on deep convolutional neural network (CNN) to estimate relative camera pose between two consecutive frames, which is independent to traditional feature detection and tracking, and reduces the camera drift.

In recent years, convolutional neural networks have been widely used in various computer vision fields. Although they were initially designed for classification purposes [10], the recent CNNs with advanced architectures have shown significant results in problems including object recognition [11], optical flow estimation [12], and dense feature matching [13] by means of simulated or actual data. Using artificial neural networks (ANN), Bell et al. [14] estimated the camera pose of teleoperated flexible endoscopes by training ANNs with optical flow magnitude and angle when the endoscope was moved by a robotic hand inside a plastic colon phantom. Recently, Kendall et al. [15] regressed the camera pose from a single RGB image by training a CNN with camera pose which was estimated offline by a structure from motion algorithm as ground truth. The main challenge then was lack of ground truth for scenes which had not enough features to track to estimate ground truth through SfM. Since annotating real images is difficult and expensive, application of synthetic data has boosted its popularity as an alternative to train networks [16, 17].

In this paper, we aim to estimate camera pose from actual optical colonoscopy video frames. To achieve this, rather than using SfM [15] to generate a ground truth, we trained a CNN by simulated colonoscopy frames for which the camera poses were available from the simulator as ground truth. The camera pose for actual colonoscopy frames was then regressed when the actual colonoscopy frames were passed to the network. The results obtained from the CNN were compared to a feature based algorithm which is explained in [18]. In addition, the performance of different networks architecture and input data (optical flow) were investigated. A diagram of our method is demonstrated in Fig. 1 and described in the following sections.

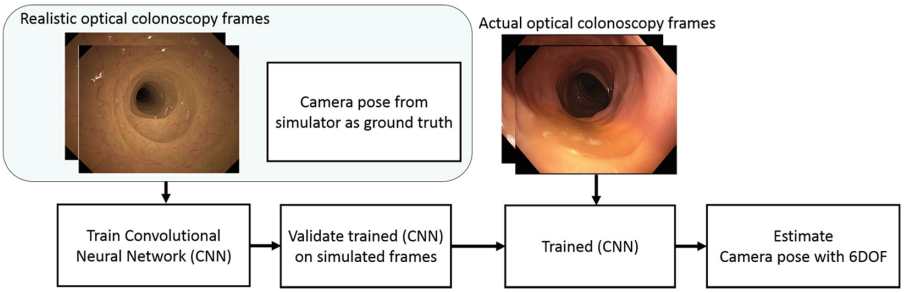

Fig. 1. Main processing steps of our proposed method

2 Method

We presented two approaches, we trained the CNN by optical flow patterns between consecutive frames, by or alternatively directly by consecutive frames. The camera pose parameters inferred from the CNN were compared to a structure from motion (SfM) algorithm [18]. First we briefly describe the preprocessing including frame preparation for training CNNs with frames, SIFT flow [19] estimation for training with motion field, and SfM as a camera pose estimation method, before explaining the proposed CNN's model architecture and training details.

2.1 Pre-processing and SfM

Frames were prepared through the following steps; (i) frames were converted to grayscale, (ii) the black corners were removed as they had no information (iii) frames were resized to train modified AlexNet and GoogLeNet. The size of input data (*height* $\times$ *width* $\times$ *number of frames*) for AlexNet and GoogLeNet were ($227 \times 227 \times 2$), ($224 \times 224 \times 2$) respectively. To estimate the optical flow pattern, the SIFT flow algorithm [19] was utilized to extract and match features between two consecutive grayscale frames. The final input data had the size of ($227 \times 227 \times 2$), and included motion field in u and v direction.

2.2 Model for Estimating Camera Pose

In this section, we describe the CNN models that estimate the camera pose parameters. The input to our models are either: the two consecutive grayscale frames; or optical flow pattern between consecutive frames. The outputs are relative camera translations and rotations with respect to the colon surface with six degrees of freedom (DoF).

Learning camera translation and rotation. Camera rotation and translation parameters were regressed by training the CNN to minimize the following objective function:

$$Loss = \beta \cdot \left\| R_{target} - R_{predicted} \right\|^2 + \left\| T_{target} - T_{predicted} \right\|^2$$

Where β is the weight factor for our dataset is one, (R_{target}, T_{target}) are camera rotation (degree) and translation (mm) which are available from the simulator as ground truth, and $(R_{predicetd}, T_{predicetd})$ are camera pose parameters predicted by the CNN. The camera translation is normalized to unit vector and is unit less. In our experiments, the camera rotation is represented in Eulerian angle (α,ψ,γ). Applying Euclidian distance on Euler angle may result in more than one set of values which can yield the same angle representation. According to [21], this can be prevented under the following conditions on the Euler angles: α, $\gamma \in [-\pi, \pi)$; $\psi \in [-\pi/2, \pi/2)$, and therefore L2 on rotation is a metric on SO(3) [21]. In our dataset the maximum of relative rotation is below the mentioned range in [21].

Network architecture. The base of the architecture of our CNN is the state-of-the-art GoogLeNet for the direct image pair. We also modified AlexNet for the optical flow approach to compare the results. These networks were originally designed for image classification. We applied the following changes on both GoogLeNet and AlexNet to regress the camera parameters; (i) considering the input data to the network, which are motion features (u,v) in two dimensions or two consecutive frames in grayscale, the first convolutional layer filter (filter size; input channel; number of filters) was modified to (11;2;96) for AlexNet and (7;2;4) for GoogelNet, allowing networks to operate in two dimensions; and (ii) the Softmax classifier was replaced by two fully connected

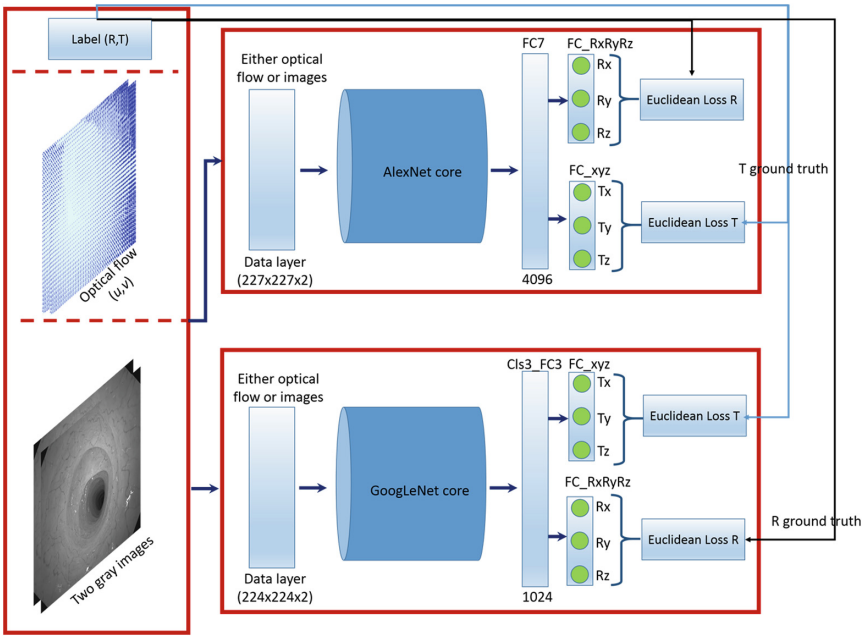

Fig. 2. The general architecture of our CNNs to predict camera pose parameters, the first layer is modified to accept two gray images or optical flow, and the classification layer was replaced by Euclidean loss to optimize predicted rotation and translation by network. Note that we trained AlexNet with both optical flow and image for comparison purposes.

layers, each with three outputs to estimate relative camera rotation and translation. The outputs of last layers then passed to a Euclidean loss function to regress the camera pose. The schematic of our networks are shown in Fig. 2.

Training details and transfer learning. The Matconvnet toolbox [22] was used for the implementation of our CNN model. We trained our network using stochastic gradient descent on our dataset which included 30,000 grayscale frames and their optical flow patterns. The batch size and iteration were 580 and 2500 respectively. To prevent any bias, input data were shuffled and randomly chosen for each batch. Since the pre-trained networks were used for our experiments, the learning rate was initialized to be 10^{-4}, and every 1000 epochs it was reduced by 0.1, and the momentum was set to 0.9. We used multi-GPU (Nvidia) for training to accelerate the training computational speed. The trained networks by simulated data then were used to predict camera pose from actual colonoscopy videos.

3 Dataset

3.1 Simulated Video

The simulated colonoscopy video frames were generated by the CSIRO colonoscopy simulator, which is explained in [23]. The simulator uses a 3D analytical model of the colon, with a haptic device allowing inspection of the simulated colon. The parametric mathematical model of the colon geometry embedded in the simulator allowed us to generate realistic human colonoscopy videos. The simulator utilized OpenGL to simulate realistic colonoscopy video based on the model and camera pose, which was used as ground truth in this paper. Appearance parameters such as illumination and specular reflection also modeled in simulator software to generate realistic colonoscopy frames.

We generated 30,000 frames from 15 different simulated colons with different structures, and a variety of possible camera motions. Each frame's size was 1352×1080 pixels, and the simulator recording rate was 30 fps.

3.2 Real Colonoscopy Video

We predicted the camera motion for actual colonoscopy video frames with our trained CNN on five segments from five different patients, each of which covered around 20 cm of colon. The videos were captured by a 190HD Olympus endoscope, with 50 fps (frame size was 1352×1080 pixels). In general 2500 vivo frames were used for validation.

4 Experiments and Results

4.1 Simulated Video

The networks were trained with 80% of data (chosen from different videos), which were shuffled to prevent bias in the training phase. The trained networks with optical

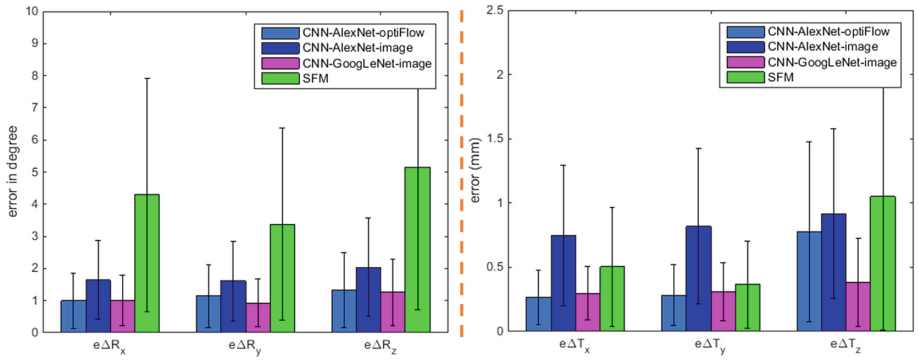

Fig. 3. The root mean square error (RMS) and standard deviation (STD) between ground truth and the camera rotations and translations estimated by SfM, modified AlexNet trained by optical flow and grayscale frames, and modified GoogLeNet trained by grayscale frames.

flow pattern and two consecutive grayscale frames were tested on the remaining data. In our study to demonstrate the performance of CNNs in comparison to a feature based algorithm; we estimated motion features between consecutive frames, removed uninformative frames [20] and computed camera translation and rotation with respect to the colon surface [18]. The results including the root mean square error (RMS) and standard deviation (STD) from the ground truth for both CNNs and SfM are shown in Fig. 3. The results indicate the higher performance of modified GoogLeNet trained by grayscale frames in comparison to other methods.

To investigate the ability of the trained networks in generalizing the results, the camera poses were computed using our trained networks from a simulated video consisting of 450 frames which were never observed by the networks during training or validation. The outcome for the distance traveled by the colonoscope camera along the Z direction is shown in Fig. 4, which demonstrates the high performance of the modified GoogLeNet trained by gray scale images.

4.2 Validation Using a Colonoscopy Phantom

Prior to validating our trained network on actual colonoscopy frames which were obtained from patients, we estimated the camera pose when colonoscope traveled back and forth in a straight phantom. It started from a start point, which represented as frame (s) in Fig. 5, and returned to the same place frame (e). Results for the distance traveled by camera in Z direction from different networks are shown in Fig. 5. The modified GoogLeNet which was trained by frames shows the lowest drift in comparison to SfM (D2) and the AlexNet when it was trained by optical flow (D1).

4.3 Application to Actual Colonoscopy Video

Actual colonoscopy videos from different parts of colons were chosen, specifically when the camera moved back and forth (a common practice during colonoscopy) to

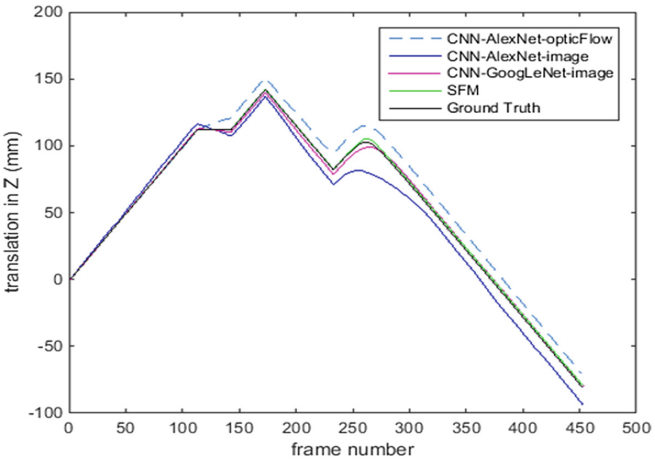

Fig. 4. The comparison for generalization of camera motion estimation by AlexNet when optical flow and frames were used for training and GoogLeNet when frames were used for training on a dataset that has not been seen before. Here, GoogLeNet shows better performance.

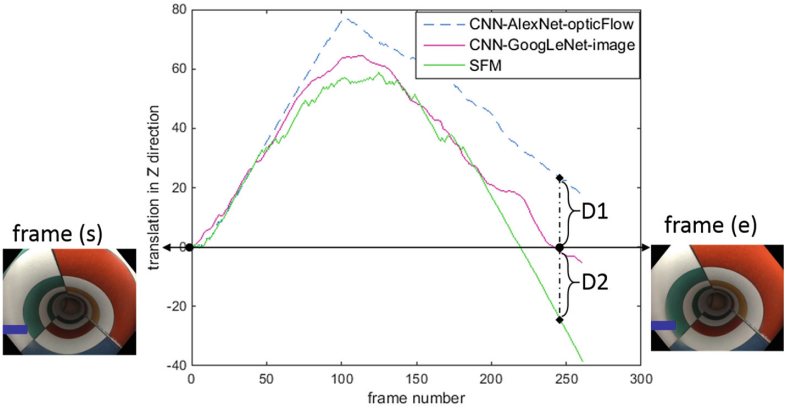

Fig. 5. Distance traveled by camera in the *Z* direction estimated by modified AlexNet, GoogLeNet and SfM (D1 and D2 represent the drift in camera motion estimation by modified AlexNet and SfM respectively), the GoogLeNet trained by frames shows very low drift.

allow validation. We estimated the distance camera traveled in the Z direction by SfM, and our CNN methods. Figure 6 represents a qualitative evaluation of the methods in the Z coordinate. During one typical examination, the colonoscope was moved back and forth during withdrawal (video in supplementary materials). The graph shows the estimation of Z coordinate along the center line for three different methods (see legend). The image inserts compares different frames that are estimated to be from the same Z location. The orange frame 54 (left insert) was visually closer to frame 166 (top magenta on right inserts) than to frame 141 (green image on right insert), suggesting

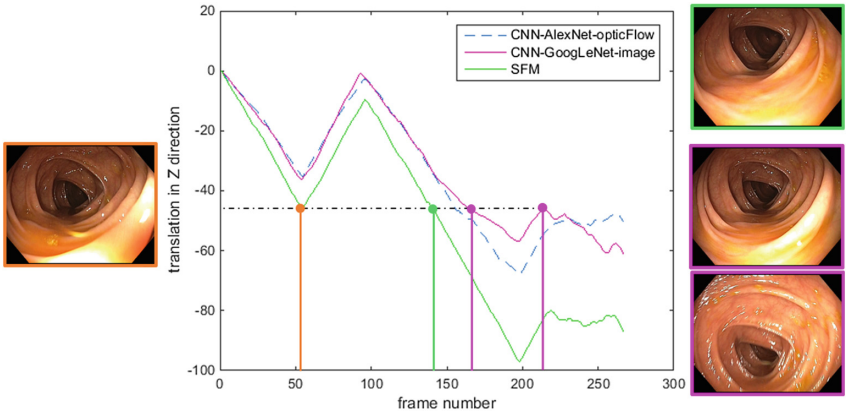

Fig. 6. The camera translation in the Z direction estimated by modified AlexNet, GoogLeNet and SfM on an actual colonoscopy video frames. Frames in orange, green and magenta are chosen to be from same Z height, but we visually understand that the closest frame to orange is the first magenta on the right inset. The modified GoogLeNet trained by grayscale frames and modified AlexNet trained by optical flow are showing better results in comparison to SfM. (color figure online)

that the CNN-based method is more accurate than the previous SfM approach. In addition, the CNN method estimated that frame 215 (magenta, bottom right insert) was also at the same location as frame 54 (orange left insert), which visually matches, whereas a drift was observed for the SfM method.

5 Discussion and Conclusion

In this paper, we presented a method to estimate the relative camera pose with six DoF from actual colonoscopy video frames. We used two separate new approaches: one modified and trained GoogLeNet using two consecutive grayscale frames as input; the other modified and trained AlexNet and used the optical flow pattern (SIFT flow) and consecutive frames (for the sake of comparison) as input. The networks, trained by simulated data were validated on simulated and actual colonoscopy frames. Our results, which are presented in Fig. 3 showed that the network which was trained with two consecutive frames could outperform the one which was trained by optical flow. In addition, modified GoogLeNet which was trained by frames had better performance in generalizing results for frames that had not been observed in training or validation stage in comparison to the one which was trained by optical flow, as it shown in Fig. 4. Some colonoscopy frames are feature-poor, thus it is hard to find accurate matches between frames, and that rendering the conventional SfM approach is inaccurate. In contrast, CNN-based approach is more robust to these issues, and resulting in higher accuracy Fig. 3.

The computational time for estimating relative camera pose from a trained network was 0.1 s on average, whereas SfM with a bundle adjustment as optimizer took three seconds when Matlab scripts were used.

In this study, we had the ground truth for camera pose from the simulator software, which we used to generate thousands of frames with a variety of camera motions incorporating translation and rotation for training and validation. Previously, others used a robot hand, magnetic sensor [14] or SfM algorithm to estimate camera pose as ground truth to train a network [15]. Any error in calibrating the sensor or estimating camera pose by SfM as ground truth could result in false data for training.

Our results on actual colonoscopy which presented in Fig. 6 indicate the high performance of CNNs in comparison to SfM for estimating the distance in the Z direction that a colonoscope camera traveled in returning to a previously seen location.

One of the main challenges in our work was transfer learning from simulated to actual frames domain. Although we could obtain remarkable results using simulated data and pre-trained networks, as a part of our future work we aim at implementing domain transfer method to improve our current results. We also investigate the performance of other networks such as visual geometry group (VGG) and will propose our network to estimate colonoscope pose.

References

1. Australian Institute of Health and Welfare. http://www.aihw.gov.au/
2. World Health Organization (WHO). Fact sheet # 297: Cancer. http://www.who.int/mediacentre/factsheets/fs297/en/
3. Hewett, D.G., Kahi, C.J., Rex, D.K.: Does colonoscopy work? J. Natl. Compr. Cancer Netw. JNCCN **8**, 67–76 (2010). quiz 77
4. Cotton, P.B., Williams, C.B.: Practical Gastrointestinal Endoscopy. Wiley-Blackwell, Oxford (2008)
5. Puerto-Souza, G.A., Staranowicz, A.N., Bell, C.S., Valdastri, P., Mariottini, G.-L.: A comparative study of ego-motion estimation algorithms for teleoperated robotic endoscopes. In: Luo, X., Reichl, T., Mirota, D., Soper, T. (eds.) CARE 2014. LNCS, vol. 8899, pp. 64–76. Springer, Cham (2014). doi:10.1007/978-3-319-13410-9_7
6. Liu, J., Subramanian, K.R., Yoo, T.S.: A robust method to track colonoscopy videos with non-informative images. Int. J. Comput. Assist. Radiol. Surg. **8**, 575–592 (2013)
7. Armin, M.A., Chetty, G., De Visser, H., Dumas, C., Grimpen, F., Salvado, O.: Automated visibility map of the internal colon surface from colonoscopy video. Int. J. Comput. Assist. Radiol. Surg. **11**, 1599–1610 (2016)
8. Rai, L., Helferty, J.P., Higgins, W.E.: Combined video tracking and image-video registration for continuous bronchoscopic guidance. Int. J. Comput. Assist. Radiol. Surg. **3**, 315–329 (2008)
9. Bao, G., Pahlavan, K., Mi, L.: Hybrid localization of microrobotic endoscopic capsule inside small intestine by data fusion of vision and RF sensors. IEEE Sens. J. **15**, 2669–2678 (2015)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
11. Aubry, M., Maturana, D., Efros, A.A., Russell, B.C., Sivic, J.: Seeing 3D Chairs: Exemplar Part-Based 2D-3D Alignment Using a Large Dataset of CAD Models, June 2014

12. Dosovitskiy, A., Fischery, P., Ilg, E., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., Brox, T.: Flownet: learning optical flow with convolutional networks. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 2758–2766. IEEE (2015)
13. Zhou, T., Krähenbühl, P., Aubry, M., Huang, Q., Efros, A.A.: Learning Dense Correspondence via 3D-guided Cycle Consistency. ArXiv Prepr. arXiv:1604.05383 (2016)
14. Bell, C.S., Obstein, K.L., Valdastri, P.: Image partitioning and illumination in image-based pose detection for teleoperated flexible endoscopes. Artif. Intell. Med. **59**, 185–196 (2013)
15. Kendall, A., Grimes, M., Cipolla, R.: Convolutional networks for real-time 6-DOF camera relocalization. Proceedings of the International Conference on Computer Vision (ICCV) (2015)
16. Su, H., Qi, C.R., Li, Y., Guibas, L.J.: Render for CNN: viewpoint estimation in images using CNNs trained with rendered 3D model views. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2686–2694 (2015)
17. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: Presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
18. Armin, M.A., De Visser, H., Chetty, G., Dumas, C., Conlan, D., Grimpen, F., Salvado, O.: Visibility map: a new method in evaluation quality of optical colonoscopy. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9349, pp. 396–404. Springer, Cham (2015). doi:10.1007/978-3-319-24553-9_49
19. Liu, C., Yuen, J., Torralba, A., Sivic, J., Freeman, W.T.: SIFT flow: dense correspondence across different scenes. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5304, pp. 28–42. Springer, Heidelberg (2008). doi:10.1007/978-3-540-88690-7_3
20. Armin, M.A., Chetty, G., Jurgen, F., De Visser, H., Dumas, C., Fazlollahi, A., Grimpen, F., Salvado, O.: Uninformative frame detection in colonoscopy through motion, edge and color features. In: Luo, X., Reichl, T., Reiter, A., Mariottini, G.-L. (eds.) CARE 2015. LNCS, vol. 9515, pp. 153–162. Springer, Cham (2016). doi:10.1007/978-3-319-29965-5_15
21. Huynh, D.Q.: Metrics for 3D rotations: comparison and analysis. J. Math. Imaging Vis. **35**, 155–164 (2009)
22. Vedaldi, A., Lenc, K.: MatConvNet: Convolutional Neural Networks for MATLAB (2015)
23. De Visser, H., Passenger, J., Conlan, D., Russ, C., Hellier, D., Cheng, M., Acosta, O., Ourselin, S., Salvado, O.: Developing a next generation colonoscopy simulator. Int. J. Image Graph. **10**, 203–217 (2010)