Deep Multi-View Stereo for Dense 3D Reconstruction from Monocular Endoscopic Video

Gwangbin Bae[1], Ignas Budvytis[1], Chung-Kwong Yeung[2], and Roberto Cipolla[1]

[1] Department of Engineering, University of Cambridge, Cambridge, United Kingdom
{gb585, ib255, rc10001}@cam.ac.uk
[2] Bio-Medical Engineering (HK) Limited, Hong Kong
ck.yeung@nisi.hk

Abstract. 3D reconstruction from monocular endoscopic images is a challenging task. State-of-the-art multi-view stereo (MVS) algorithms based on image patch similarity often fail to obtain a dense reconstruction from weakly-textured endoscopic images. In this paper, we present a novel deep-learning-based MVS algorithm that can produce a dense and accurate 3D reconstruction from a monocular endoscopic image sequence. Our method consists of three key steps. Firstly, a number of depth candidates are sampled around the depth prediction made by a pre-trained CNN. Secondly, each candidate is projected to the other images in the sequence, and the matching score is measured using a patch embedding network that maps each image patch into a compact embedding. Finally, the candidate with the highest score is selected for each pixel. Experiments on colonoscopy videos demonstrate that our patch embedding network outperforms zero-normalized cross-correlation and a state-of-the-art stereo matching network in terms of matching accuracy and that our MVS algorithm produces several degrees of magnitude denser reconstruction than the competing methods when same accuracy filtering is applied.

Keywords: Multi-View Stereo · 3D Reconstruction · Endoscopy

1 Introduction

The capability of estimating depth can improve the quality and safety of the monocular endoscopy. The obtained depth information can be used to estimate the shape and size of the lesions, improving the accuracy of the visual biopsy, or to identify the safest navigation path. Numerous attempts have been made with such motivations. However, methods that require device modification (e.g. additional light source [13, 10], depth sensors [1], stereo setup [3]) could not be evaluated *in vivo* due to engineering and regulatory barriers.

A cheaper alternative is to perform 3D reconstruction directly from the images by using multi-view geometry. For example, structure-from-motion (SfM) [14] identifies the matches between multiple images of the scene (taken from

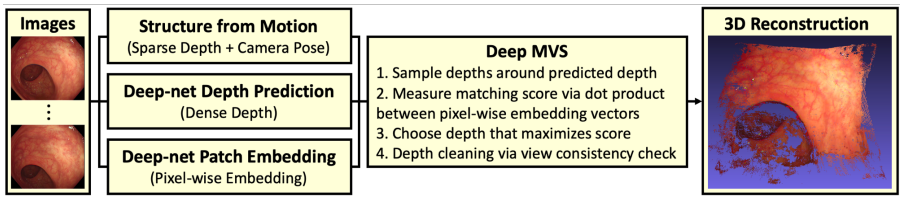

Fig. 1. Our deep multi-view stereo pipeline

different viewpoints) and jointly optimizes their 3D coordinates and the relative camera poses by minimizing the reprojection error. The reconstructed points can be trusted as they are geometrically verified. However, the resulting reconstruction is very sparse. For a sequence of 8 colonoscopy images (of resolution 624×540) SfM can only reconstruct about 200 points.

Recent works [5, 8] have shown that it is possible to train deep neural networks with sparse SfM reconstruction. After training, such a network can predict dense pixel-wise depth map for a given image. Nonetheless, the predicted depth map lacks accuracy as it is generated from a single image, and is not validated from other viewpoints. For example, single-view depth prediction can be affected by the presence of motion blur, light speckles, or fluids.

A possible solution is to use multi-view stereo (MVS) algorithms. Once the camera poses are estimated (e.g. via SfM), an MVS algorithm tries to find the optimal depth each pixel should have in order for it to be projected to the visually similar pixels in the other images. However, classical MVS algorithms suffer from two weaknesses - large search space and poor matching accuracy. For example, in PatchMatch stereo [2], the depth map is initialized randomly from a uniform distribution ranging between some pre-set upper ($d_{\max}$) and lower limit ($d_{\min}$). Then, the depths with high matching score are propagated to the neighboring pixels. In BruteForce stereo [9], selected number of depths ranging from $d_{\min}$ to $d_{\max}$ are tried for each pixel, and the one with the highest score is selected. In both scenarios, finding the correct depth becomes challenging if $(d_{\max} - d_{\min})$ is large. Another problem with the conventional MVS approaches is the inaccuracy of the patch-matching. A typical matching function is the zero-normalized cross-correlation (ZNCC). Since the computational cost increases quadratically with the patch size, small patch sizes (e.g. 7×7) are often preferred. However, small patch can lead to ambiguous matches especially in texture-less images.

In this paper, we present a deep-learning-based MVS pipeline (see Fig. 1) that can solve the aforementioned problems. The novelty of our approach is three-fold. Firstly, we use a monocular depth estimation network to constrain the search space for the depth candidate sampling. Secondly, we introduce a novel patch embedding network that significantly improves the accuracy and reduces the computation cost compared to the ZNCC and other stereo matching network. Lastly, we demonstrate that, after measuring the scores for the neighboring images in the sequence, selecting the minimum, as opposed to maximum, improves the quality of the reconstruction by enforcing the multi-view consis-

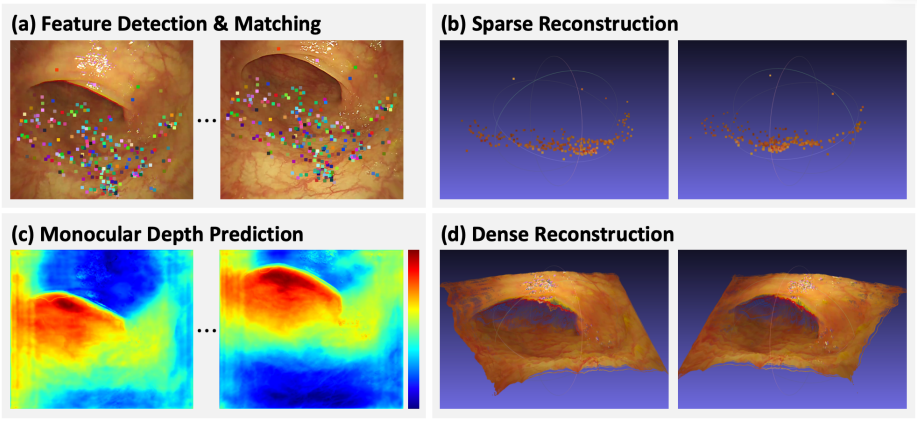

Fig. 2. (a-b) Sparse reconstruction obtained via SfM. (c-d) Dense depth prediction obtained via CNN trained on SfM reconstructions.

tency. Our method is evaluated on colonoscopy videos but can be extended to any monocular endoscopy.

2 Method

Our method consists of three pre-processing steps followed by a multi-view stereo reconstruction. The three pre-processing steps are (1) sparse reconstruction via SfM, (2) monocular depth estimation, and (3) embedding vector generation via patch embedding network. Then, the MVS pipeline generates a geometrically validated 3D reconstruction. The following sections provide details of each step.

2.1 Sparse Depth and Camera Pose Estimation via SfM

Firstly, the sparse reconstruction and relative camera poses are estimated via SfM. In order to minimize the effect of the non-rigid surface deformation, the endoscopy videos are split into short sequences, consisting of 8 consecutive frames separated by 0.08 seconds. For each sequence, the SfM reconstruction is obtained using OpenSfM [9]. When optimizing the 3D feature coordinates and the camera poses, only the features that appear in all 8 images are considered. This ensures that the camera poses are supported by all feature coordinates and are hence accurate. See Fig. 2 for an example of a resulting sparse reconstruction.

2.2 Dense Depth Prediction via CNN

The sparse reconstruction obtained via SfM is then used to train a monocular depth estimation network. Due to the inherent scale ambiguity of SfM, the training loss is computed after matching the scale of the prediction to that of the ground truth. More formally, the loss is defined as:

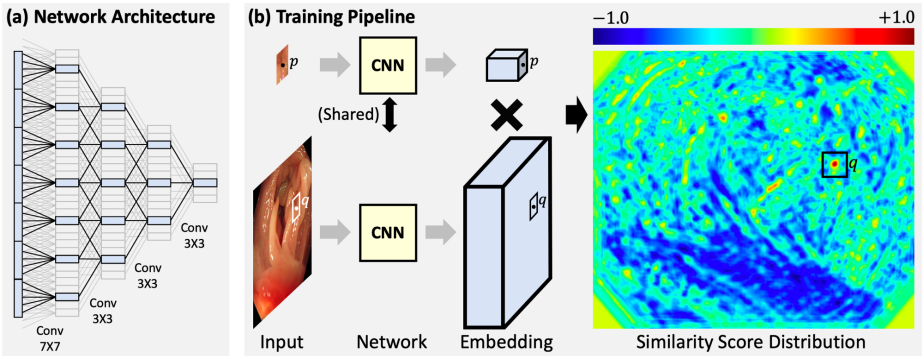

Fig. 3. (a) The architecture of our patch embedding network. (b) During training, a reference patch and the target image are passed through the network and the score distribution is optimized.

$$L = \min_{s \in \mathcal{S}} \sum_{\mathbf{p}} \mathbb{1}\left(d_{\mathbf{p}}^{\text{true}} > 0\right) \left(d_{\mathbf{p}}^{\text{true}} - s \times d_{\mathbf{p}}^{\text{pred}}\right)^2 \tag{1}$$

where $\mathcal{S}$ is a discrete set of scaling factors (e.g. ranging from 0.5 to 2.0) and $\mathbb{1}\left(d_{\mathbf{p}}^{\text{true}} > 0\right)$ is a binary variable which is equal to 1 if a true depth is available for the pixel $\mathbf{p} = (u, v)$ and is 0 otherwise. By applying such scale-invariant loss, the network is able to learn the relative depth (i.e. the ratio between pixel-wise depths). In this paper, we use the U-Net architecture [12] with a single output channel. See Fig. 2 for an example of a predicted depth map.

2.3 Pixel-wise Embedding via Patch Embedding Network

The goal of the patch embedding network is to map an image patch around each pixel into an embedding vector $\mathbf{f}$, so that the dot product between two vectors can be used as a measure of their patch similarity. Inspired by the SIFT feature descriptor [6], we divide the receptive field (of size 49×49 pixel) into 7×7 cells of the same size. Then, a 7×7 convolutional layer (with 64 output channels) is used to identify the low-level features in each cell. This results in a feature map of size $7 \times 7 \times 64$. Then, a set of three convolutional layers with 3×3 kernels maps the feature map into a 64-dimensional vector. The same network can be applied to a full image (suitably padded) using dilated convolutions (see Fig. 3). Each convolutional layer is followed by a batch normalization and a rectified linear unit (ReLU). Following Luo et. al. [7], we use linear activation in the last layer to preserve the information in the negative values. Compared to the conventional patch embedding networks which use repeated convolutional layers of small size (e.g. [15, 7, 16]), our architecture can incorporate larger visual context while having fewer parameters. Lastly, the output vector is normalized, so that the dot product of two vectors can range between -1 and 1.

Fig. 3 illustrates the training pipeline. Suppose that the pixel $\mathbf{p}$ in the reference image corresponds to the pixel $\mathbf{q}^{\text{true}}$ in the target image. The reference patch (centered at $\mathbf{p}$) and the target image pass through the embedding network (the two branches share the parameters). The network outputs a vector $\mathbf{f}_{\mathbf{p}}^{\text{ref}}$ and a vector map of size $H \times W \times 64$. $\mathbf{f}_{\mathbf{p}}^{\text{ref}}$ is then multiplied to the embedding vector at each pixel in the target image. This generates a pixel-wise score distribution.

Ideally, we want the score to be high only near the pixel $\mathbf{q}^{\text{true}}$. To achieve this objective, we introduce a novel soft contrastive loss, which is defined as:

$$\begin{aligned} L_{\mathbf{p},\mathbf{q}^{\text{true}}} = &\sum_{\mathbf{q}} \max(w_{\mathbf{q}}, 0) \left(1 - \mathbf{f}_{\mathbf{p}}^{\text{ref}} \cdot \mathbf{f}_{\mathbf{q}}^{\text{target}}\right) \\ &+ \sum_{\mathbf{q}} \max(-w_{\mathbf{q}}, 0) \max\left(\mathbf{f}_{\mathbf{p}}^{\text{ref}} \cdot \mathbf{f}_{\mathbf{q}}^{\text{target}} - \alpha, 0\right) \\ \text{where} \quad w_{\mathbf{q}} = &\cos\left(\frac{||\mathbf{q} - \mathbf{q}^{\text{true}}||\pi}{5}\right) \quad \text{if } ||\mathbf{q} - \mathbf{q}^{\text{true}}|| \leq 5 \text{ and } -1 \text{ otherwise.} \end{aligned} \tag{2}$$

As a result, the score is maximized if $\mathbf{q}$ is less than 2.5 pixels away from $\mathbf{q}^{\text{true}}$ and is minimized elsewhere if it is larger than the threshold, α. In this paper α is set to 0.7 empirically.

2.4 Deep Multi-View Stereo Reconstruction

Our MVS reconstruction pipeline consists of three key steps. Firstly, 50 depth candidates are sampled uniformly from $0.9 \times d_{\text{pred}}$ to $1.1 \times d_{\text{pred}}$, where d_{pred} is the predicted depth. Each candidate is then projected to all the other images in the sequence, and the score is measured for each of them. This gives $N_{\text{image}} - 1$ scores. From these, the minimum is selected and is assigned to the depth candidate. Once the score is assigned to every candidate, the one with the highest score is selected. More formally, the depth at pixel $\mathbf{p}$ in the i-th image is selected as:

$$\hat{d}_{\mathbf{p}}^{i} = \underset{d \in \mathcal{D}}{\text{argmax}} \left[\min_{j} \left(\mathbf{f}_{\mathbf{p}}^{i} \cdot \mathbf{f}_{P^j(d)}^{j}\right)\right] \tag{3}$$

where $\mathcal{D}$ and $P^j(d)$ represent the set of depth candidates and the projection of the depth candidate d on the j-th image.

The resulting depth-map is then filtered via view consistency check. In this step, $\hat{d}_{\mathbf{p}}^{i}$ is projected to a different image in the sequence and is compared to the estimated depth value at the projected pixel. If there is less than 1% difference between the two values, the two depths are considered "consistent". If this is satisfied for all 7 images in the sequence, $\hat{d}_{\mathbf{p}}^{i}$ is merged into the final reconstruction. The number of survived pixels is then used as a quantitative measure of the reconstruction accuracy.

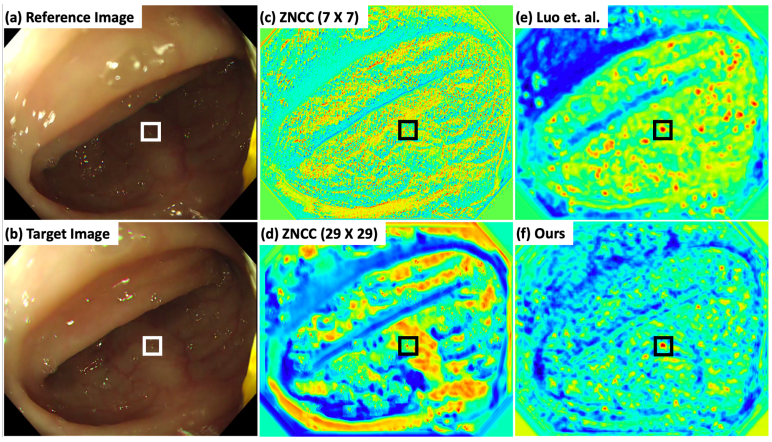

Fig. 4. Qualitative comparison between the score distribution generated by different methods shows that our method leads to significantly reduced matching ambiguity.

3 Experiments

3.1 Experimental Setup

Our dataset consists of 51 colonoscopy videos, each containing a full procedure ($\sim$ 20min) of a different patient. 201,814 image sequences are extracted, of which 34,868 are successfully reconstructed via SfM. The sequences from 40 videos are used to train, validate and test the depth estimation network and the patch embedding network. For the training and testing of the patch embedding network, the SIFT [6] features that survived the SfM reconstruction are used to establish the ground truth matching. Lastly, the performance of our MVS pipeline is tested on the sequences from the remaining 11 videos.

The depth estimation and patch embedding networks are implemented and trained using PyTorch [11] framework. Both networks are trained for 80 epochs with a batch size of 32 using Adam optimizer [4] ($\beta_1 = 0.9$, $\beta_2 = 0.999$). The learning rate is set to 0.001 and is reduced every 20 epochs, to 0.0007, 0.0003 and 0.0001, respectively.

3.2 Accuracy of the Patch Embedding Network

The aim of this experiment is to quantitatively evaluate the matching accuracy of our patch embedding network. The accuracy is measured as following: The embedding vector generated for the reference pixel is convolved with all the embeddings in the target image. The pixel in the target image that maximizes the score (i.e. dot product) is then selected as $\mathbf{q}^{\text{pred}}$. The error is defined as $||\mathbf{q}^{\text{pred}} - \mathbf{q}^{\text{true}}||$. We report the median error and the percentage of matches with error larger than 3, 5, and 10 pixels. We also measure the time it takes to run a stereo reconstruction with each scoring method. Table 1 shows the results.

Table 1. Accuracy of stereo matching networks and ZNCC-based matching

	Patch Size	# Params	Median Error	> 3px	> 5px	> 10px	Runtime(s)
ZNCC	7×7	N/A	125.419	0.675	0.671	0.664	64.9
ZNCC	29×29	N/A	1.0	0.174	0.127	0.109	777.3
ZNCC	49×49	N/A	1.414	0.209	0.135	0.097	2029.1
Luo et. al.[7]	37×37	695136	**1.000**	**0.064**	0.034	0.020	**28.0**
Ours	49×49	120768	**1.000**	0.077	**0.028**	**0.013**	**28.0**

Table 2. Number of pixels that survive view consistency check

Method	Search Space	Matching Function	Score Selection	Average	Median
BruteForce[9]	$(d_{\min}, d_{\max})$	ZNCC (7×7)	Max	31	6
PatchMatch[9]	$(d_{\min}, d_{\max})$	ZNCC (7×7)	Max	17	1
Ours	$(0.9, 1.1) \times d_{\text{pred}}$	ZNCC (7×7)	Max	157	29
		DeepNet ([7])	Max	164	39
		DeepNet (ours)	Max	479	96
		DeepNet ([7])	Min	5910	3622
		DeepNet (ours)	Min	**10454**	**6452**

While small patch does not contain sufficient information, large patch includes the surrounding pixels the appearance of which is highly view-dependent. This explains why the accuracy of the ZNCC matching peaks at an intermediate patch size (29×29). On the contrary, deep-learning-based approaches show high accuracy despite the large patch size. Compared to the state-of-the-art stereo matching network [7], our network contains fewer parameters, has larger receptive field, and achieves better accuracy except for the 3 pixel error rate. Since the patch embedding networks encode each patch into a concise representation, the matching requires less computation compared to the ZNCC, resulting in significantly reduced reconstruction runtime.

Fig. 4 shows the score distribution generated by each method. For ZNCC, small patch leads to ambiguous matches, while large patch results in over-smoothed score distribution. Our network, compared to Luo et. al. [7], shows high score only near the correct pixel. This is mainly due to the soft-contrastive loss (Eq. 2) which penalizes the large scores at incorrect pixels.

3.3 Evaluation of the MVS Reconstruction

This experiment aims to compare the accuracy of our MVS pipeline to that of the competing methods. Since ground truth dense reconstruction is not available, we use the number of pixels that survive the view consistency check (see Sect. 2.4) as a quantitative measure of accuracy. Table 2 shows the obtained results. For our method, we also show the contribution of each component. Compared to the BruteForce and PatchMatch reconstruciton (implemented in OpenSfM [9]), our method produces several orders of magnitude denser reconstruction. Fig. 5 provides qualitative comparison between the resulting 3D reconstructions.

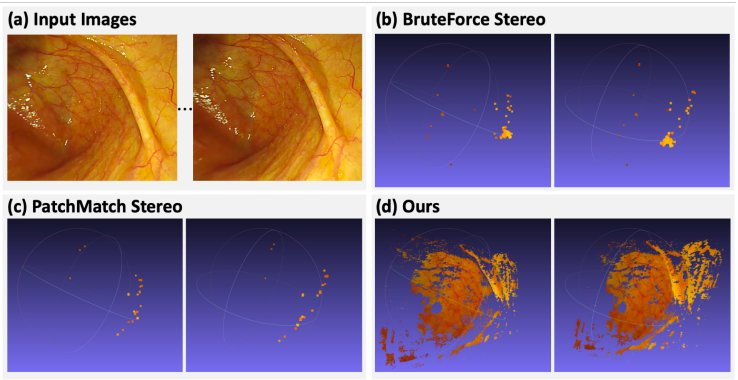

Fig. 5. 3D reconstructions obtained from different methods

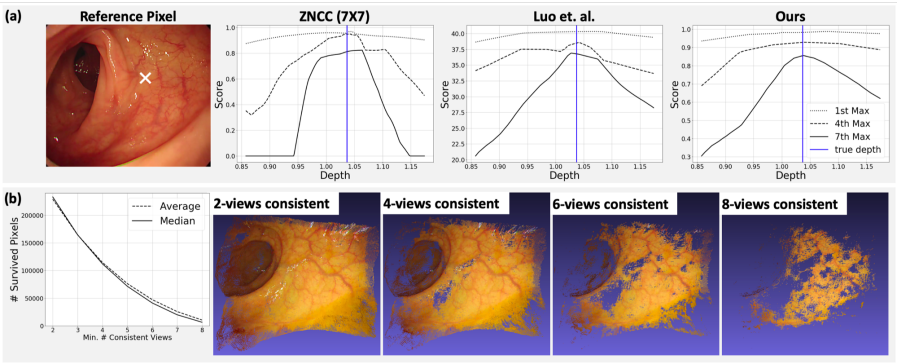

Fig. 6. (a) Visual justification for selecting the minimum score. (b) Trade-off between density and accuracy.

Fig. 6 shows the results of two additional experiments. Firstly, the score for each depth candidate is computed with different n-th best selection of the matching score. In BruteForce and PatchMatch, the maximum is selected, while we choose the minimum. The result shows that selecting the 7-th best (i.e. minimum) enforces the depth candidate to be supported by all available views, thereby suppressing the scores of the ambiguous matches. Such behavior is best observed when using our patch embedding network. Second experiment shows the relationship between the minimum number of consistent view (used in depth cleaning) and the number of survived pixels. By decreasing this parameter, it is possible to obtain denser reconstruction, while sacrificing the accuracy.

4 Conclusion

In this work, we proposed a deep-learning-based multi-view stereo reconstruction method that can produce dense and accurate 3D reconstruction from a sequence

of monocular endoscopic images. We demonstrated that a pre-trained depth estimation network can constrain the search space and improve the reconstruction accuracy. We also introduced a novel patch embedding network that outperforms ZNCC and the state-of-the-art stereo matching network. For a fixed constraint on view consistency, our method produces several degrees of magnitude denser reconstruction than the competing methods.

References

1. Sensor-based guidance control of a continuum robot for a semi-autonomous colonoscopy. Robotics and Autonomous Systems **57**(6), 712 – 722 (2009)
2. Bleyer, M., Rhemann, C., Rother, C.: Patchmatch stereo-stereo matching with slanted support windows. In: Bmvc. vol. 11, pp. 1–11 (2011)
3. Hou, Y., Dupont, E., Redarce, T., Lamarque, F.: A compact active stereovision system with dynamic reconfiguration for endoscopy or colonoscopy applications. In: Golland, P., Hata, N., Barillot, C., Hornegger, J., Howe, R. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014. pp. 448–455. Springer International Publishing, Cham (2014)
4. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
5. Liu, X., Sinha, A., Unberath, M., Ishii, M., Hager, G.D., Taylor, R.H., Reiter, A.: Self-supervised learning for dense depth estimation in monocular endoscopy. In: OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis, pp. 128–138. Springer (2018)
6. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004)
7. Luo, W., Schwing, A.G., Urtasun, R.: Efficient deep learning for stereo matching. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
8. Ma, R., Wang, R., Pizer, S., Rosenman, J., McGill, S.K., Frahm, J.M.: Real-time 3d reconstruction of colonoscopic surfaces for determining missing regions. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 573–582. Springer (2019)
9. Mapillary: Opensfm. https://github.com/mapillary/OpenSfM (2017)
10. Parot, V., Lim, D., González, G., Traverso, G., Nishioka, N.S., Vakoc, B.J., Durr, N.J.: Photometric stereo endoscopy. Journal of biomedical optics **18**(7), 076017 (2013)
11. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., dAlché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019)
12. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)

13. Schmalz, C., Forster, F., Schick, A., Angelopoulou, E.: An endoscopic 3d scanner based on structured light. Medical image analysis **16**(5), 1063–1072 (2012)
14. Ullman, S.: The interpretation of structure from motion. Proceedings of the Royal Society of London **203**(1153), 405–426 (1979)
15. Zagoruyko, S., Komodakis, N.: Learning to compare image patches via convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4353–4361 (2015)
16. Žbontar, J., LeCun, Y.: Stereo matching by training a convolutional neural network to compare image patches. The journal of machine learning research **17**(1), 2287–2318 (2016)