Chapter 5

Real-time Tracking and Reconstruction with Deep Representation

In the previous chapters, we have introduced a complete pipeline for surface reconstruction from a monocular endoscopic video with automatic video-CT registration. The pipeline can provide a high-quality surface model of the anatomy and camera trajectory. However, for some applications (*e.g.*, surgical navigation) in endoscopy, a real-time solution is needed so that the estimate of surface geometry and camera trajectory can provide instant feedback (*e.g.*, regions not yet inspected) to assist the surgeon during the endoscopy procedure. This could also potentially enable au-

tomatic applications such as intelligent endoscope holder and automatic endoscopy inspection.

Simultaneous Localization and Mapping (SLAM) is a type of algorithm that can estimate geometry and trajectory estimates in real-time. Many monocular visual SLAM methods have been developed for general scenes [53, 54, 56, 222–231] and clinical applications such as endoscopy [49, 110, 232–234]. Though such systems have been studied and developed for decades, many practical and theoretical challenges remain. Specifically for endoscopy, scarce texture, illumination variation, tissue deformation, and surgical manipulation are several typical challenges. These challenges either result in low robustness and accuracy of the system running or break certain assumptions of the existing SLAM systems.

In this chapter, we exploit deep learning-based representations to handle the scarce texture and illumination variation, to improve the robustness and accuracy of the system. The deep representation also enables the system to produce surface geometry of the anatomy. Based on our evaluation, the proposed SLAM system generalizes well to unseen endoscopes and subjects and has superior performance compared with a state-of-the-art feature-based SLAM system [54].

5.1 Related Work

5.1.1 Representation Learning for Visual Tracking and Mapping

In recent years, researchers have worked on exploiting prior information learned from previous data to improve the performance of SLAM and Visual Odometry (VO). Different forms of deep depth prior have been used, such as single depth estimate [110, 224, 234], self-improving depth estimate [229], depth estimate with uncertainty [235], and depth estimate with optimizable code [227, 231, 236].

Deep appearance representations have been studied to replace the role of RGB image, which improves convergence basin and enables scenarios with no photometric constancy. BA-Net [236] proposed representation learning with differentiable BA-related loss. DeepSFM [237] extracted implicit feature representation and built cost volume to jointly optimize depth map and relative pose. In this work, we use learning-based viewpoint- and illumination-robust appearance representation and optimizable depth to effectively integrate priors into the SLAM system.

There are also works exploiting other forms of priors for the VO and SLAM systems. For example, Yang *et al*. [235] exploit a pose prior to enable better convergence and mitigate the scale-drift issue; Zhan *et al*. [238] estimate dense optical flow to gain

more robustness towards camera tracking.

5.1.2 Simultaneous Localization and Mapping in Endoscopy

Many SLAM systems have been studied and proposed for the general scene [53, 54, 56, 222–231]. In endoscopy, additional challenges exist compared with other SLAM scenarios such as driving scenes, which are illumination changes, scarce textures, deformation, *etc*.

Feature-based SLAM [49, 109, 239, 240] has been developed for its robustness to illumination changes. However, these systems are not robust to scarce and repetitive textures and thus not suitable for our application. To deal with the scarce texture that causes inaccuracy in terms of trajectory and reconstruction, works have been proposed using either hardware [51] or algorithmic [15, 234, 241] solution. However, the estimated geometry from these systems is not dense and thus cannot allow for target applications of this work that require such information. Deformation happens in endoscopy, especially in certain cases such as laparoscopy and when surgical operations are applied. Works have been developed to confront this challenge [56, 57, 242, 243]. In this work, we exploit learning-based priors and dense geometry to improve the robustness of the system to illumination changes and scarce texture.

5.2 Contributions

In this work, we made the following contributions:

- An effective training scheme to jointly learn optimizable depth and illumination-robust representations with differentiable non-linear optimization.
- A full-feature learning-based dense SLAM system is developed for endoscopy with decent generalizability.
- We demonstrate the effectiveness of the proposed method on *in vivo* and *ex vivo* nasal endoscopic videos, by comparing the performance with state-of-the-art feature-based system ORB-SLAM v3 [54].

From the system point of view, this work presents a real-time option to estimate camera trajectory and dense geometry with a slight sacrifice of accuracy. Many endoscopic applications require such a real-time solution. For example, it enables the surgeon to know which regions have not been observed from the endoscope yet, which increases the chance to find pathology such as polyps, cancer, *etc*. It improves the awareness of surgeons on the critical structures underlying the surface if pre-operative CT imaging is available, which is enabled by the video-CT registration with the dense geometry estimate from the SLAM.

5.3 Representation Learning

In this section, we introduce the learning scheme of various representations used in the proposed SLAM system.

5.3.1 Network Architecture

Two separate networks are used to learn geometry and appearance representations, respectively. In terms of geometry, a depth network is responsible for producing an average depth estimate, which is correct up to a global scale, and a collection of depth bases. The average depth estimate captures the expectation of the depth estimate based on the input color image. However, the task of depth estimation from a single image is ill-posed and therefore errors are expected. The depth bases consist of a set of depth variations that could be used to explain the variation of geometry given the appearance of the input. Therefore, such bases provide a way to further refine the depth estimate, using information from other frames (*e.g.*, geometric consistency), during the optimization process in a SLAM system run.

As shown in Fig. 5.1, the depth network is close to UNet [244] with partial convolution [245]. The endoscope image mask is used in the partial convolutions so that input regions outside the mask do not contribute to the final output. There are two output branches, where one, with absolute function as output activation, predicts the

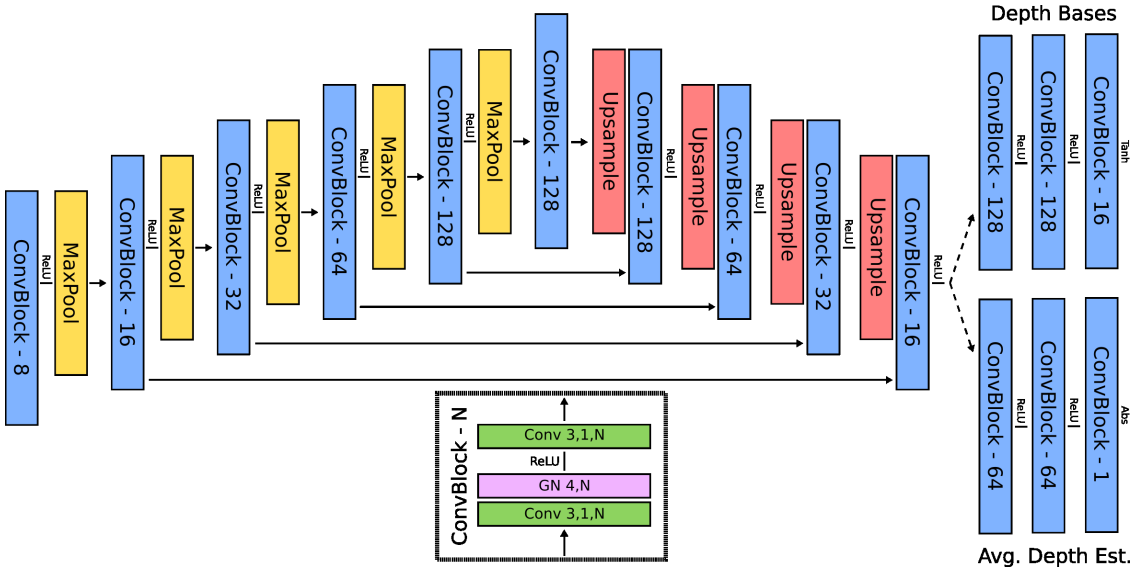

Figure 5.1: **Network architecture for optimizable depth estimation.** Each ConvBlock consists of two partial convolution layers with kernel size as 3 and stride as 1, one group normalization layer with a group size of 4, and one ReLU activation, which are arranged in the way as the figure above. The number after the ConvBlock means the size of the output channel dimension. Two output branches exist in the network for the average depth estimate and the depth bases, described in Sec. 5.3.1. Hyperbolic tangent and absolute functions are used as output activation in these branches.

average depth estimate, and the other produces depth bases with hyperbolic tangent as output activation. The architecture of the discriminator used for depth training is shown in Fig. 5.2.

In terms of appearance, a feature network produces two types of representations. One set of representations, named descriptor map, is used as image descriptors in pair-wise feature matching that are involved in the Reprojection Factor and Sparse Matched Geometry Factor, described in Sec. 5.4.2. A similar training approach as Chapter 2 is used, except that we use point correspondences computed from the sur-

face reconstruction and trajectory instead of the correspondences from SfM. The other set, named feature map, is used for the computation of the Feature-metric Factor as a drop-in replacement of the original video frame. In the image, the illumination of the same location of the scene changes as the viewpoint varies because the lighting source moves with the camera. On the other hand, feature maps can be robust to illumination and viewpoint changes, if the feature network is trained correspondingly.

In this work, we use the task of pair-wise image alignment with differentiable non-linear optimization to train both the appearance and geometry representations, with more details in Sec. 5.3.4. The network architecture for the feature network is the same as the depth network, except for the two output branches. The sizes of channel dimension for the three layers in both the descriptor map and feature map output branches (from hidden to output) are 64, 64, and 16; the output activation functions are both hyperbolic tangent.

5.3.2 Differentiable Optimization

To make the networks learn to master the task of pair-wise image alignment, a differentiable non-linear optimization method is required. In this work, we use Levenberg-Marquardt (LM) algorithm [75] as the optimization solver. LM is a trust-region algorithm to find a minimum of a function over a space of parameters. It is also known as a damped least-squares method because a damping factor is involved in the

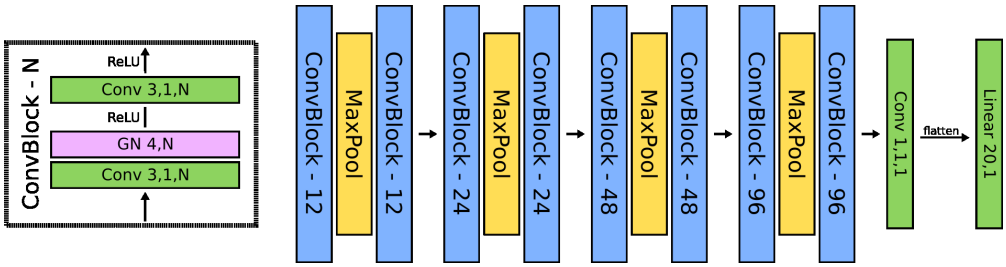

Figure 5.2: **Network architecture of discriminator for depth estimation learning.** The input is the RGB image and the normalized depth map, concatenated along the channel dimension, with a resolution of 64×80. Each ConvBlock consists of two normal convolution layers with kernel size as 3 and stride as 1, one group normalization layer with a group size of 4, and two ReLU activation layers, which are arranged in the way as the figure above. The number after the ConvBlock means the size of the output channel dimension. The final convolution layer, with kernel size as 1, stride as 1, and output channel size as 1, and linear layer, with input channel size as 20 and output channel size as 1, converts the feature map to a scalar value used to indicate the predicted validity of the input sample. Note that before being fed to the final linear layer, the output map from the final convolution layer is first flattened along the sample-wise dimensions.

method that explicitly controls how large the trust region is. The damping factor will decrease or increase based on whether the proposed parameter updates in a single step results in a lower error or not. The larger the damping factor is, the closer LM will be to the gradient descent method. On the other hand, the smaller the damp is, the closer LM will be to the Gauss-Newton method [246], which is a quadratic optimization method.

In the computation graph, all accepted steps are connected, while the accept determination stage and rejected steps are not involved. This removes the need to have an additional network, which is used in BA-Net [236], to predict the damping factor of LM optimization for each iteration, and reduces the complexity of the computa-

tion graph by removing those unnecessary steps. A differentiable solve method for the linear system [82] is used to solve the variable update of a single iteration. With gradient checkpoint technique [82], the number of accepted steps is almost not limited by the memory storage, because each added one will only require a negligible amount of memory space. Therefore, in each iteration of the network training, a long optimization chain can be used.

5.3.3 Loss Design

For each iteration, when the LM optimization converges, several outputs before, during, and after the optimization process will be involved in the loss computation for the network training. Both the average and the optimized depth estimate should agree with the groundtruth depth map up to a global scale. We do not let the depth network try to predict the correct scale and instead leave it to the optimization during SLAM running because predicting a correct depth scale from a monocular endoscopic image is nearly impossible. Therefore, a scale-invariant loss is used for this objective. With a predicted depth map $\boldsymbol{D} \in \mathbb{R}^{1\times H\times W}$, the corresponding groundtruth depth map $\tilde{\boldsymbol{D}} \in \mathbb{R}^{1\times H\times W}$, and the binary video mask $\boldsymbol{V} \in \mathbb{R}^{1\times H\times W}$, the loss is defined as

$$\mathcal{L}_{\text{si}} = \frac{\sum \boldsymbol{D}_{\text{ratio}}^2}{\sum \boldsymbol{V}} + \frac{\left(\sum \boldsymbol{D}_{\text{ratio}}\right)^2}{\left(\sum \boldsymbol{V}\right)^2} \quad , \tag{5.1}$$

where $\boldsymbol{D}_{\text{ratio}} = \log\left(\boldsymbol{V}\boldsymbol{D} + \epsilon\right) - \log\left(\boldsymbol{V}\tilde{\boldsymbol{D}} + \epsilon\right)$. Note all operations, except $\sum$, are element-wise ones; $\sum$ operation sums all elements along the sample-wise dimensions; $\epsilon \in \mathbb{R}$ is a small number to prevent logarithm over zero. Note that all groundtruth data used in this work can be obtained from the surface reconstruction and camera trajectory produced in Chapter 3.

To guide the intermediate depth maps during optimization, we additionally use an adversarial loss. Intuitively, this loss functions as a regularizer and helps make intermediate depth maps more physically feasible given the visual cues (*e.g.*, illumination distribution) in the input color image. This should thus encourage the network to produce a better set of depth bases to produce such depth estimates. The real sample for the GAN will be a color image and the corresponding normalized groundtruth depth map; the fake sample will be the color image and the corresponding normalized depth estimate. For normalization, these depth maps are divided by their maximum value so that the discriminator judges the fidelity of the sample pair based only on the relative geometry and not on the depth scale. The loss form in LS-GAN [247] is used in this work.

For the descriptor map, the RR loss defined in Chapter 2 is used. Because a descriptor map is also used for loop closure detection, besides producing good feature matches on images with large scene overlap, having dissimilar descriptions for images with small or no scene overlap is also desired. A histogram loss is used to make

sure the similarity between histograms of descriptor maps for the source and target images is higher than that for the source and far images. The definitions of these three images are in Sec. 5.3.4. The histogram loss is defined as

$$\mathcal{L}_{\text{hist}} = \frac{1}{C} \sum_{i \in \{1,\dots,C\}} \min \left(\frac{1}{K} d_{\text{EMD}} \left(\boldsymbol{h}_i^{\text{src}}, \boldsymbol{h}_i^{\text{tgt}} \right) - \frac{1}{K} d_{\text{EMD}} \left(\boldsymbol{h}_i^{\text{src}}, \boldsymbol{h}_i^{\text{far}} \right) + \eta_{\text{hist}}, 0 \right) \quad , \tag{5.2}$$

where $d_{\text{EMD}} \left(\boldsymbol{h}_1, \boldsymbol{h}_2 \right) = \left\| \text{CDF} \left(\boldsymbol{h}_1 \right) - \text{CDF} \left(\boldsymbol{h}_2 \right) \right\|_2^2$ measures the earth mover's distance between two histograms. CDF is the operation to produce cumulative density function (CDF) from a histogram. $\boldsymbol{h}_i^{\text{src}} \in \mathbb{R}^K$ is the soft histogram of elements within the valid region of source descriptor map $\boldsymbol{I}^{\text{src}} \in \mathbb{R}^{C \times H \times W}$ along the i^{th} channel, which is $\boldsymbol{I}_i^{\text{src}} \in \mathbb{R}^{1 \times H \times W}$; K is the number of bins in each cumulative density function (CDF) and C is the channel size of the descriptor map; $\eta_{\text{hist}} \in \mathbb{R}$ is a constant margin.

To compute soft CDF differentiably, we refer to the method in [248]. The value of k^{th} bin in the histogram $\boldsymbol{h}_i^{\text{src}}$ can be written as follows

$$\boldsymbol{h}_i^{\text{src}} \left(k \right) = \frac{1}{\left| \Omega^{\text{src}} \right|} \sum_{\boldsymbol{x} \in \Omega^{\text{src}}} \left(\sigma \left(\frac{\boldsymbol{I}_i^{\text{src}} \left(\boldsymbol{x} \right) - \mu_k + 1/K}{\beta} \right) - \sigma \left(\frac{\boldsymbol{I}_i^{\text{src}} \left(\boldsymbol{x} \right) - \mu_k - 1/K}{\beta} \right) \right) \quad , \tag{5.3}$$

where the center value of k^{th} bin is $\mu_k = -1 + \left(2k + 1 \right) / K \in \mathbb{R}$; the kernel function is $\sigma \left(a \right) = 1 / \left(1 + \mathrm{e}^{-a} \right)$. The values used in μ_k are related to that the descriptor map has a value range of $(-1, 1)$ because of the architectural design of the feature network. The output activation function is hyperbolic tangent for the descriptor map. Ω^{src} is a

set consisting of all 2D locations within the source video mask; $\beta \in \mathbb{R}$ is a bandwidth parameter. The histograms for target and source images are the same as above except the corresponding descriptor maps are used for calculation instead of the source one.

Intuitively, after the optimization process in Sec. 5.3.2, the source image should be warped to the target frame with good alignment, using the estimate of status. Such a warping process can be described with a 2D scene flow. Therefore, to guide the learning process to produce better image alignment, another loss is to encourage the similarity between the groundtruth 2D scene flow, and the one estimated after the optimization process. This objective will provide signals to both the feature map branch of the feature network and the depth network. This is because a reasonable 2D scene flow can only be achieved if the feature maps are expressive and the depth estimates are accurate, especially when this loss is combined with the depth-related losses above. The flow loss is defines as

$$\mathcal{L}_{\text{flow}} = \frac{1}{\omega^{\text{s}\to\text{t}} \sum \boldsymbol{V}} \sum \boldsymbol{V} \left(\tilde{\boldsymbol{W}}^{\text{s}\to\text{t}} - \boldsymbol{W}^{\text{s}\to\text{t}} \right)^2 \quad , \tag{5.4}$$

where $\tilde{\boldsymbol{W}}^{\text{s}\to\text{t}} \in \mathbb{R}^{2\times H\times W}$ and $\boldsymbol{W}^{\text{s}\to\text{t}} \in \mathbb{R}^{2\times H\times W}$ are the groundtruth and estimated 2D scene flows from source to target frame, respectively. $\omega^{\text{s}\to\text{t}} \in \mathbb{R}$ is a normalization factor, defined as $\omega^{\text{s}\to\text{t}} = \frac{1}{2} \sum \boldsymbol{V} \left((\tilde{\boldsymbol{W}}^{\text{s}\to\text{t}})^2 + (\boldsymbol{W}^{\text{s}\to\text{t}})^2 \right)$. The estimated flow $\boldsymbol{W}^{\text{s}\to\text{t}}$ at

2D location $\boldsymbol{x}^{\mathrm{src}}$ is defined as

$$\boldsymbol{W}^{\mathrm{s}\to\mathrm{t}}\left(\boldsymbol{x}^{\mathrm{src}}\right)=\pi\left(\boldsymbol{p}^{\mathrm{s}\to\mathrm{t}}\right)-\boldsymbol{x}^{\mathrm{src}}\quad\text{, where} \tag{5.5}$$

$$\boldsymbol{p}^{\mathrm{s}\to\mathrm{t}}=\boldsymbol{T}_{\mathrm{src}}^{\mathrm{tgt}}\pi^{-1}\left(\boldsymbol{x}^{\mathrm{src}},\boldsymbol{D}^{\mathrm{src}}\left(\boldsymbol{x}^{\mathrm{src}}\right)\right)\quad. \tag{5.6}$$

$\boldsymbol{p}^{\mathrm{s}\to\mathrm{t}}\in\mathbb{R}^3$ is the 3D location of the lifted source 2D location $\boldsymbol{x}^{\mathrm{src}}\in\mathbb{R}^2$ in the target coordinate system, based on the current estimate of status. π and π^{-1} are the project and unproject operation of the camera geometry. These two operations are the same for all keyframes because camera intrinsics are assumed to be fixed throughout the video. $\boldsymbol{T}_{\mathrm{src}}^{\mathrm{tgt}}=\left(\boldsymbol{T}_{\mathrm{tgt}}^{\mathrm{wld}}\right)^{-1}\boldsymbol{T}_{\mathrm{src}}^{\mathrm{wld}}$ is the relative pose between target and source. $\boldsymbol{D}^{\mathrm{src}}\left(\boldsymbol{x}^{\mathrm{src}}\right)\in\mathbb{R}$ is the depth estimate at 2D location $\boldsymbol{x}^{\mathrm{src}}$ based on the current estimate of depth scale and depth code. It is defined as $\boldsymbol{D}^{\mathrm{src}}\left(\boldsymbol{x}^{\mathrm{src}}\right)=s^{\mathrm{src}}\left(\bar{\boldsymbol{D}}^{\mathrm{src}}\left(\boldsymbol{x}^{\mathrm{src}}\right)+\left(\boldsymbol{c}^{\mathrm{src}}\right)^{\intercal}\hat{\boldsymbol{D}}^{\mathrm{src}}\left(\boldsymbol{x}^{\mathrm{src}}\right)\right)$. The source average depth estimate and depth bases are $\bar{\boldsymbol{D}}^{\mathrm{src}}\in\mathbb{R}^{1\times H\times W}$ and $\hat{\boldsymbol{D}}^{\mathrm{src}}\in\mathbb{R}^{B\times H\times W}$. the source depth scale, depth code, and camera pose matrix are $s^{\mathrm{src}}\in\mathbb{R}$, $\boldsymbol{c}^{\mathrm{src}}\in\mathbb{R}^B$, and $\boldsymbol{T}_{\mathrm{src}}^{\mathrm{wld}}\in\mathrm{SE}\left(3\right)$, respectively. Note that the forms of all definitions related to the other images are the same as the source, except that the superscript symbol should be changed correspondingly.

5.3.4 Training Procedure

In each iteration, three images are used for training, which are the source, target, and far images. Source and target are two images with a large scene overlap, while the far image has a small or no scene overlap with the source. For the source and target images, the depth network produces the average depth estimate and depth bases, and the feature network produces a feature map and descriptor map. The far image is only used in the second-stage training described later and is only involved in the loss calculation for the descriptor map from the feature network.

The network training consists of two stages. At the first stage, the depth estimates (excluding the depth bases) and the descriptor maps (excluding the feature map) are trained separately with the scale-invariant loss and RR loss, respectively. After both networks are trained to a reasonable state, the training moves to the second stage, where two networks are jointly trained with the scheme below. The objective then becomes that, with good geometry and appearance representations produced from these two networks, a source image should be well aligned to a target image with a non-linear optimization. The variables that are optimized over are relative camera pose, depth scale, and depth code associated with the source image. The factors involved are pair-wise factors, FM, SMG, and GC, and prior factors, SC and CD. A random relative camera pose and all-zero depth code are initialized. The initial source depth scale is computed so that the mean values of target and source depth maps are equal. After

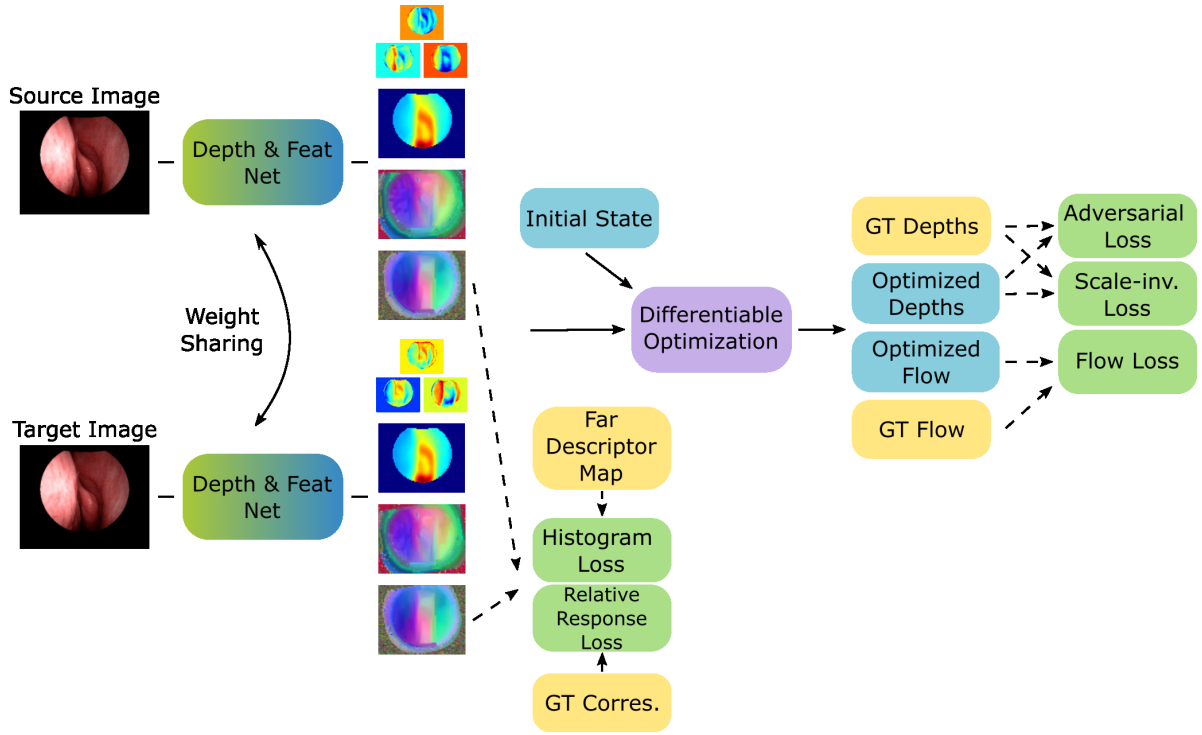

Figure 5.3: **Diagram of representation learning.** The network outputs for each image (from top to bottom) are the depth bases, average depth estimate, feature map, and descriptor map. Network outputs and the initial state of variables (relative pose, depth scales, depth codes) are input to the differentiable optimization pipeline to obtain optimized depth estimates and 2D flow map for loss computation. Descriptor map for the far image is used in the histogram loss. More details are described in Sec. 5.3.

these pre-processing, the optimization described in Sec. 5.3.2 is applied to minimize the objective described by the factors.

With the optimization finished, the loss functions described in Sec. 5.3.3 are calculated and the networks then get updated. Note that there is also a typical GAN-related training cycle [247] involved because we use the adversarial loss for depth training. The training diagram for the second stage is shown in Fig. 5.3.

5.4 Simultaneous Localization and Mapping

5.4.1 Overview

The SLAM system modules are organized into frontend and backend threads. Frontend consists of *Camera Tracking* and *Keyframe Creation* modules. The *Camera Tracking* module is used to track the new video frame against the reference keyframe, where the depth scale of the new frame and relative pose will be optimized over. The *Keyframe Creation* module is used to determine if a new keyframe is needed. If so, a new keyframe will be created and the connections to temporally close keyframes will be built. For each keyframe, a bag-of-words description will be created for efficient global loop detection later in the *Loop Closure* module.

Backend threads run *Loop Closure* and *Mapping* modules. The *Loop Closure* module constantly detects both local and global connections between all keyframe pairs. Whenever a global connection is detected, a lightweight pose-scale graph optimization will be applied to close the loop by adjusting depth scales and camera poses. The *Mapping* module runs full factor graph optimization constantly, where all depth codes, depth scales, and camera poses are jointly optimized with all factors that are described in Sec. 5.4.2. The overall diagram of the SLAM system is shown in Fig. 5.4.

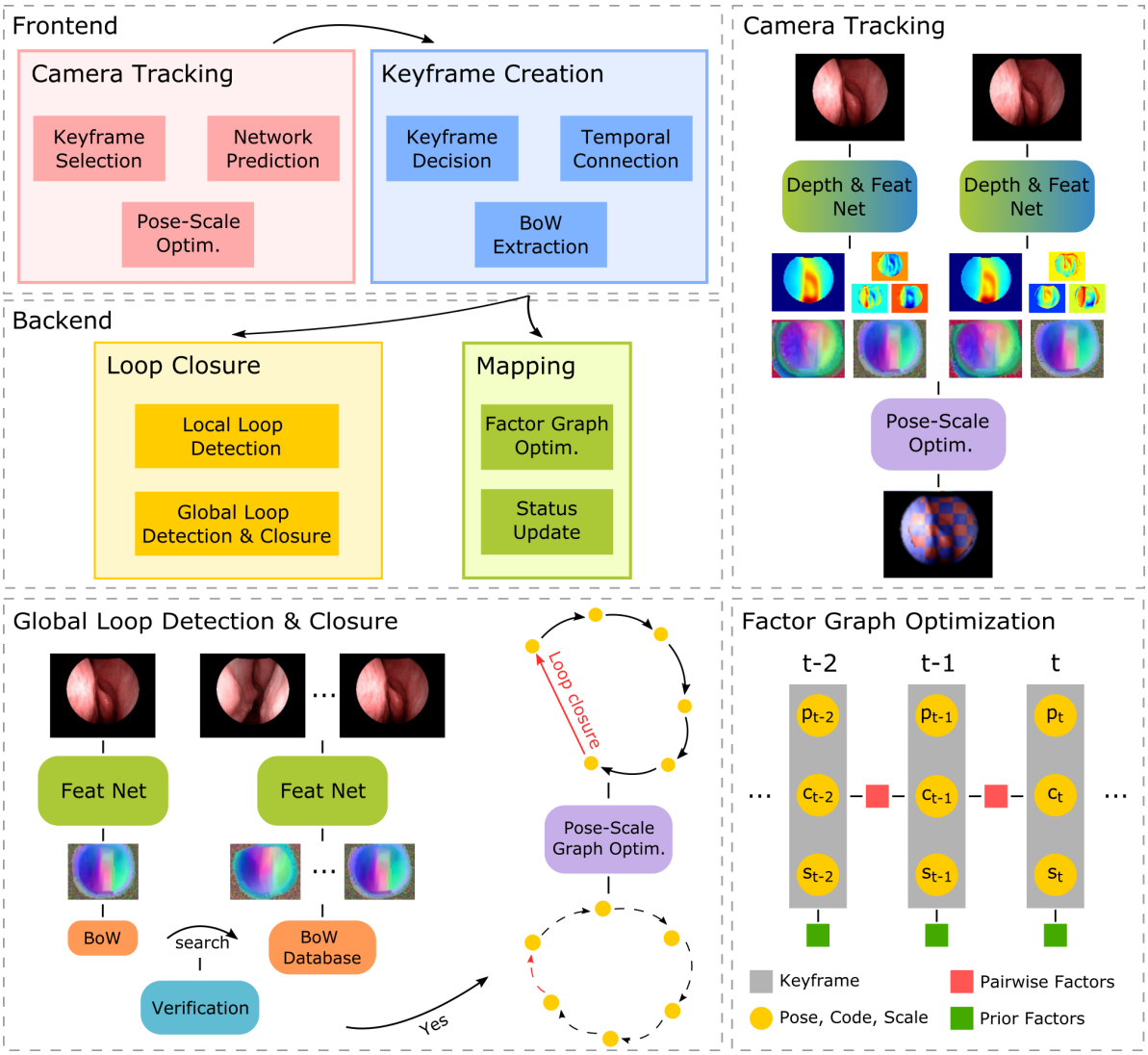

Figure 5.4: **Overall diagram of SLAM system.** The top left shows the module relationship in the proposed SLAM system. The top right demonstrates the network prediction and pose-scale optimization within the *Camera Tracking* module. Note that only a subset of depth bases is displayed. The bottom left shows the process of global loop detection and closure of the *Loop Closure* module. The bottom right demonstrates factor graph optimization in the *Mapping* module. In the pose-scale graph optimization of the global loop closure, only poses and depth scales are optimized. Note that pair-wise factors only between adjacent keyframes are displayed for simplicity.

5.4.2 Factor Design

Feature-metric Factor. Intuitively, if all relevant variables are accurate, the source image warped to the target image plane should align well with the target image in terms of appearance. In this factor, the feature map from the feature network is used as the appearance representation of a frame for reasons described in Sec. 5.3.1. The feature map is pre-processed to form a Gaussian pyramid with a specified number of levels to increase the convergence basin. To build a certain level of the Gaussian pyramid, the Gaussian smoothing operation with a specified size and sigma, and 2-time downsampling will be applied sequentially to the map in the previous level. Note that the binary endoscope mask is also used in generating the Gaussian pyramid so that invalid regions do not contribute to the Gaussian smoothing.

The source feature map pyramid is defined as $\mathcal{F}^{\text{src}} = \{\boldsymbol{F}_i^{\text{src}} | i = 1, \dots, L\}$, where L is the number of levels and $\boldsymbol{F}_i^{\text{src}} \in \mathbb{R}^{C \times H/2^{i-1} \times W/2^{i-1}}$ is the feature map at pyramid level i; The objective of this factor is defined below.

$$\mathcal{L}_{\text{fm}} = \frac{1}{L} \sum_{i=1}^{L} \frac{1}{|\Omega_{\text{src,tgt}}|} \sum_{\boldsymbol{x}^{\text{src}} \in \Omega_{\text{src,tgt}}} \left\| \boldsymbol{F}_i^{\text{tgt}} \left(\pi \left(\boldsymbol{p}^{\text{s} \to \text{t}} \right) \right) - \boldsymbol{F}_i^{\text{src}} \left(\boldsymbol{x}^{\text{src}} \right) \right\|_2^2 \quad , \tag{5.7}$$

where $\Omega_{\text{src,tgt}}$ is the set of source 2D locations that can be projected onto the target mask region given the estimate of the status.

Sparse Matched Geometry Factor. In cases where variables of two frames

are far from being accurate, it is difficult to rely only on the Feature-metric Factor to converge to the correct optimization minima. This is because, even though the feature network is trained to produce feature maps with a better convergence ability, it still has the issue of a relatively small convergence basin, which is common for the appearance-warping based objectives [101].

The descriptor map from the feature network can estimate 2D point correspondences between images through pair-wise feature matching described in Sec. 2.4.3. This enables the objective to have global convergence characteristics. Because in this work, each keyframe has a depth estimate, we can extend the 2D correspondences to 3D ones. Compared with 2D ones, this results in fewer outliers in the correspondences after the geometric outlier removal, which follows the feature matching process. It is because we have depth information available and the outlier removal based on point cloud alignment has less ambiguity than the 2D filtering method based on epipolar geometry. Intuitively, if the depth estimates of two keyframes are correct up to a global scale, a similarity transform estimated from the inlier matches should align two point clouds well. After alignment, outlier matches are those whose spatial distances are larger than the corresponding noise bounds.

The point cloud registration used in this work is Teaser++ [196], which is shown to be robust to a large outlier rate. Teaser++ originally allows single noise bound and we extend its implementation so that a point-wise noise bound can be used. For

each feature match, we set the noise bound to be the depth value of the matched point in the target image multiplying a specified constant factor. Geometrically, this corresponds to how many pixels are allowed in the location difference between the matched and projected 2D location. The definition of this factor is:

$$\mathcal{L}_{\text{smg}} = \frac{1}{|\mathcal{M}|} \sum_{(\boldsymbol{x}^{\text{src}}, \boldsymbol{x}^{\text{tgt}}) \in \mathcal{M}} \rho_{\text{fair}} \left(\left\| \boldsymbol{p}^{\text{s}\to\text{t}} - \pi^{-1} \left(\boldsymbol{x}^{\text{tgt}}, \boldsymbol{D}^{\text{tgt}} \left(\boldsymbol{x}^{\text{tgt}} \right) \right) \right\|_2^2 ; \delta_{\text{smg}}^{\text{src}} \right) \quad , \tag{5.8}$$

where $\mathcal{M}$ is a set of feature matches consisting of pairs of 2D locations $(\boldsymbol{x}^{\text{src}}, \boldsymbol{x}^{\text{tgt}}) \in \mathbb{R}^2 \times \mathbb{R}^2$, and $\delta_{\text{smg}}^{\text{src}} = \frac{\sigma_{\text{smg}}}{|\Omega^{\text{src}}|} \sum_{\boldsymbol{x} \in \Omega^{\text{src}}} \bar{\boldsymbol{D}}^{\text{src}} (\boldsymbol{x})$, which is the mean value of the source average depth estimate multiplying a constant factor $\sigma_{\text{smg}} \in \mathbb{R}$. The outlier-robust "Fair" loss [246] is used, which is defined as $\rho_{\text{fair}} (a; b) = 2(\sqrt{a/b} - \ln (1 + \sqrt{a/b}))$.

Reprojection Factor. This factor behaves similarly to the Sparse Matched Geometry Factor except that the objective is changed from minimizing the average distance of 3D point sets to minimizing the average distance of projected source-to-target 2D locations and target 2D locations. The factor is defined as:

$$\mathcal{L}_{\text{rp}} = \frac{1}{|\mathcal{M}|} \sum_{(\boldsymbol{x}^{\text{src}}, \boldsymbol{x}^{\text{tgt}}) \in \mathcal{M}} \rho_{\text{fair}} \left(\left\| \pi \left(\boldsymbol{T}_{\text{src}}^{\text{tgt}} \pi^{-1} \left(\boldsymbol{x}^{\text{src}}, \boldsymbol{D}^{\text{src}} \left(\boldsymbol{x}^{\text{src}} \right) \right) \right) - \boldsymbol{x}^{\text{tgt}} \right\|_2^2 ; \sigma_{\text{rp}} W^2 \right) \quad , \tag{5.9}$$

where $\sigma_{\text{rp}} \in \mathbb{R}$ is a multiplying factor and W is the width of the involved depth map. In this work, we assume all keyframes have the same resolution.

Geometric Consistency Factor. In all factors above, only one depth estimate

of the image pair is used, except the Sparse Matched Geometry Factor that is only used in the geometric verification described in Sec. 5.4.6. Therefore, the geometric consistency between two depth estimates is not enforced yet. On the other hand, this factor ensures such consistency by encouraging the source depth estimate transformed to the target coordinate to have consistent values as the target depth estimate. The factor is defined as:

$$\mathcal{L}_{\mathrm{gc}} = \frac{1}{|\mathcal{M}|} \sum_{(\boldsymbol{x}^{\mathrm{src}}, \boldsymbol{x}^{\mathrm{tgt}}) \in \mathcal{M}} \rho_{\mathrm{cauchy}} \left(\left\| z^{\mathrm{s}\to\mathrm{t}} - \boldsymbol{D}^{\mathrm{tgt}} \left(\pi \left(\boldsymbol{p}^{\mathrm{s}\to\mathrm{t}} \right) \right) \right\|_2^2 ; \delta_{\mathrm{gc}}^{\mathrm{src}} \right) \quad , \tag{5.10}$$

where $z^{\mathrm{s}\to\mathrm{t}}$ is the z-axis component of $\boldsymbol{p}^{\mathrm{s}\to\mathrm{t}}$; $\delta_{\mathrm{gc}}^{\mathrm{src}}$ is the same as $\delta_{\mathrm{smg}}^{\mathrm{src}}$, except that σ_{gc} is used instead of σ_{smg}. Cauchy loss [246] is used to increase the robustness of this factor, which is defined as $\rho_{\mathrm{cauchy}}\left(a; b\right) = \ln\left(1 + a/b\right)$.

Relative Pose Scale Factor. This factor is used only in the pose-scale graph optimization over depth scales and camera poses for the global loop closure described in Sec. 5.4.6. The intuition of this factor is the scale ambiguity of pair-wise factors for a keyframe pair. The error value for the pair-wise factors above will not change if depth scales and the translation component of the relative camera pose are scaled jointly. In the stage of global loop closure, all frame pairs except the newly detected global loop should have reasonably variable estimates. Therefore, the functionality of this factor is to keep variable estimates in the previous links as close to the original

estimates as possible up to a global scale and make the new global loop pair reach the goal. Specifically, the overall objective is to make the ratio of depth scales, rotation of the relative pose, and translation of the relative pose up to a scale reach the target values. The factor is defined as follows:

$$\begin{aligned}\mathcal{L}_{\text{rps}} = &\left\| \frac{\boldsymbol{t}_{\text{src}}^{\text{tgt}}}{s^{\text{src}}} - \frac{\tilde{\boldsymbol{t}}_{\text{src}}^{\text{tgt}}}{\tilde{s}^{\text{src}}} \right\|_2^2 + \omega_{\text{rot}} \left\| \log\left(\boldsymbol{R}_{\text{src}}^{\text{tgt}}\right) - \log\left(\tilde{\boldsymbol{R}}_{\text{src}}^{\text{tgt}}\right) \right\|_2^2 + \\ &\omega_{\text{scl}} \left(\log\left(\frac{s^{\text{tgt}}}{s^{\text{src}}}\right) - \log\left(\frac{\tilde{s}^{\text{tgt}}}{\tilde{s}^{\text{src}}}\right) \right)^2 \quad ,\end{aligned} \tag{5.11}$$

where $\boldsymbol{t}_{\text{src}}^{\text{tgt}} \in \mathbb{R}^3$ and $\boldsymbol{R}_{\text{src}}^{\text{tgt}} \in \text{SO}(3)$ are the translation and rotation components of the relative pose $\boldsymbol{T}_{\text{src}}^{\text{tgt}}$ described above, respectively. Note that the logarithm operation on the rotation components is the matrix logarithm of $\text{SO}(3)$ [249]. $\omega_{\text{rot}} \in \mathbb{R}$ and $\omega_{\text{scl}} \in \mathbb{R}$ are the weights for the rotation and scale components of this factor, respectively. In this equation and the ones below, every symbol with $\tilde{}$ on top represents the target counterpart of the one without it.

Code Factor. This factor is used to keep the depth code of a keyframe within a reasonable range. Each keyframe has this factor included in the full factor graph. Note that this factor and the following Scale Factor and Pose Factor only involve one keyframe per factor. It is defined as

$$\mathcal{L}_{\text{code}} = \frac{1}{B} \left\| \boldsymbol{c}^{\text{src}} - \tilde{\boldsymbol{c}}^{\text{src}} \right\|_2^2 \quad . \tag{5.12}$$

Scale Factor. This factor is to make the depth scale of a keyframe as close to the target scale as possible. It is used for the first keyframe in the full factor graph and the new global loop pair in the pose-scale graph of the process of global loop closure. It is defined as

$$\mathcal{L}_{\text{scale}} = \left(\log\left(s^{\text{src}}\right) - \log\left(\tilde{s}^{\,\text{src}}\right)\right)^2 \quad . \tag{5.13}$$

Pose Factor. This factor is used for the first keyframe to anchor the pose trajectory of the entire graph. It is defined as

$$\mathcal{L}_{\text{pose}} = \left\|\boldsymbol{p}_{\text{src}}^{\text{wld}} - \tilde{\boldsymbol{p}}_{\,\text{src}}^{\,\text{wld}}\right\|_2^2 + \omega_{\text{r}} \left\|\log\left(\boldsymbol{R}_{\text{src}}^{\text{wld}}\right) - \log\left(\tilde{\boldsymbol{R}}_{\text{src}}^{\text{wld}}\right)\right\|_2^2 \quad , \tag{5.14}$$

where $\omega_{\text{r}} \in \mathbb{R}$ is the weight of the rotation component of this factor.

5.4.3 Camera Tracking

This module is used to continuously track new video frames to provide a good initialization point for the other modules, shown in Fig. 5.4. When a new frame comes in, it will be tracked against the reference keyframe. The spatially closest keyframe against the latest tracked frame is used as the reference, where the distance is based on the current estimates of camera poses. In some cases, the selection could be wrong because of drifting errors, especially when it is temporally far from the latest keyframe. To verify the selection, the feature matching inlier ratio between

the new frame and the selected keyframe is computed as in the Reprojection Factor. The same metric is also computed between the new frame and the latest keyframe. If the former is smaller than the latter multiplying a factor, the latest keyframe will be used instead as the reference.

Camera tracking is solved with LM optimization over the relative camera pose between the new frame and reference and the depth scale of the new frame. The factors involved are the Feature-metric Factor and Reprojection Factor. The termination of optimization is based on several criteria, which are the maximum number of iterations, parameter update ratio threshold, and gradient threshold. In this module, only the relative pose $\boldsymbol{T}_{\text{src}}^{\text{tgt}}$ is optimized over. Once the optimization finishes, the pose of the new frame, labeled as source, can be calculated as $\boldsymbol{T}_{\text{src}}^{\text{wld}} = \boldsymbol{T}_{\text{tgt}}^{\text{wld}} \boldsymbol{T}_{\text{src}}^{\text{tgt}}$, where $\boldsymbol{T}_{\text{tgt}}^{\text{wld}}$ is the camera pose of the reference keyframe.

5.4.4 Keyframe Creation

This module is for handling keyframe creation and pre-processing. When the first keyframe is created, the depth scale is initialized so that the median value of the average depth estimate is set to one. In this way, the global scale of the camera trajectory is relatively stable. And thus the values of different components in the Relative Pose Scale Factor, used in the *Loop Closure* module, are stable across different videos and different trained models with a fixed parameter setting. This, in turn, results in more

stable global loop closure performance. In terms of the prior factors, for the first keyframe, all three prior factors, *i.e.*, Code Factor, Scale Factor, and Pose Factor, are integrated into the factor graph, while, for the other keyframes, only Code Factor will be constructed.

For every tracked new frame, this module determines if a new keyframe is needed. Because the global scale of the entire graph is ambiguous, no absolute distance threshold can be relied on. Instead, we use a set of more intuitive criteria that directly relate to the information gain of a new frame, which are scene overlap, feature match inlier ratio, and the average magnitude of 2D scene flow. Scene overlap measures the overlap between two frames and reflects how much new region is observed from a new frame. Feature match inlier ratio is the ratio of inlier matches over all the feature match candidates. This reflects how dissimilar the two frames are in terms of appearance, which may be due to a small region overlap, a dramatic texture change, *etc*. As for the texture change, it could be caused by auto exposure adjustment, tissue bleeding, and so on. The average magnitude of 2D scene flow measures how much movement the content of a frame has. It measures one additional movement that is the in-plane camera rotation. This is to track the camera movement of keyframes more continuously and to produce more consistent descriptors and feature maps between keyframes.

For each keyframe, a bag-of-words description is computed from the descriptor

map and added to the database for global loop indexing, described in Sec. 5.4.6. Temporal connections will be added to the new keyframe. These only consist of temporally close keyframes. The number of temporally connected keyframes depends on the feature match inlier ratio. At least one keyframe will be connected to the new one. Additional keyframes, up to a specified maximum number, will be connected only if the ratio between the additional keyframe and the new keyframe is larger than a specified threshold. The factors involved in the pair-wise keyframe connections are the Feature-metric Factor and Geometric Consistency Factor.

5.4.5 Mapping

Mapping is constantly running at the backend. The framework for factor graph optimization is ISAM2 [250]. The entire factor graph consisting of pair-wise and prior factors from all keyframes is optimized in this module, where Fig. 5.4 shows an example of the factor graph. The variables jointly optimized are camera poses, depth scales, and depth codes of all keyframes. Whenever a global loop closure in the *Loop Closure* module finishes, all involved variables in the full factor graph will be reinitialized with the new values.

5.4.6 Loop Closure

As another backend module, the *Loop Closure* module constantly tries to find potential keyframe pairs that can be local or global loop connections and handles the closure correspondingly. For local loop detection, the keyframes, which are visited before the query one within a specified temporal range, are searched. Because the temporal window is set to be small, the trajectory drifting error will not be large, the camera pose of each keyframe can still be roughly relied on for filtering candidates. For this reason, the spatial distance between keyframe pairs is first used.

For the following verification steps, the query keyframe and the closest one within its temporal connections are used as the reference pair. If the spatial distance between the candidate pair is smaller than the spatial distance between the reference pair multiplying a constant factor, the pair will be kept. For pairs being kept after distance filtering, the appearance verification will be run, where the feature match inlier ratio is computed. The candidate pair will be kept if the inlier ratio is larger than that of the reference pair multiplying a constant factor and a specified constant inlier ratio. Lastly, a geometric verification is applied, where a pair-wise optimization similar to the one in the *Camera Tracking* module is run. The difference in terms of factors is that the Sparse Match Geometry Factor is used in place of the Reprojection Factor. It is because the Sparse Match Geometry Factor optimizes 3D distances instead of 2D ones and therefore has higher robustness on variable initialization.

The local connection will only be accepted if the overlap ratio and flow magnitude, computed in the geometric verification, are larger and smaller than those of the reference pair multiplying a constant factor, respectively. After verification, only the best candidate left, in terms of overlap ratio and flow magnitude, will be used to build the local connection. The selected keyframe pairs are linked with pair-wise factors same as the temporal connections.

Another part of this module is global loop connection and closure, as shown in Fig. 5.4. Global loop detection searches for keyframe pairs whose interval is beyond a specified temporal range. Unlike the local loop detection where camera poses can still be relied on to choose candidates, global loop detection uses the appearance of keyframes for the initial candidate selection. The descriptor map estimated by the feature network per keyframe describes the point-wise appearance distinctively and is suitable to be used as the representation to build a bag-of-words place recognition model [251].

A hierarchical bag-of-words method [251] is used in this work, where the model is built from the estimated descriptor maps of a training dataset. Whenever a keyframe is created, the bag-of-words descriptor will be added to a database. When a global loop connection is searched for a query keyframe, the database will be searched through with the extracted bag-of-words descriptor. A specified number of keyframes that are similar to the query keyframe in terms of bag-of-words descriptor will be selected

as candidates. The candidates are then filtered so that the description similarities between the query keyframe and candidates are larger than the similarity between the reference pair multiplying a specified constant factor. One additional requirement is that candidates should not be temporally close to the query keyframe, opposite to the local loop connection. After that, the same appearance and geometric verification as the local loop detection are used to verify the global loop candidates. The verified candidates are ranked based on feature match inlier ratio and, from high to low, each candidate that is temporally far enough from the selected candidates is added to avoid connection redundancy.

Unlike the local loop connection, for the global one, the drifting error between the global keyframe pair is often large. Therefore, it is slow to rely on the full graph optimization in the *Mapping* module to close the gap. To this end, we design a lightweight pose-scale graph optimization for the global loop closure, where the camera poses and depth scales of all keyframes are optimized jointly. In this graph, a set of lightweight factors are used. For the new global loop pair, the Scale Factor and Relative Pose Scale Factor are used, where the target depth scales come from the geometric verification above; For all other keyframe connections, the Relative Pose Scale Factor is used, where the current values are used as the target scales and poses in the factors. The graph optimization terminates if one of two conditions is met: 1) the number of iterations reaches a specified number and 2) the number of consecutive iterations

with no relinearization reaches a specified number. After the optimization finishes, depth scales and camera poses of all keyframes, in the full factor graph of the *Mapping* module, are reinitialized correspondingly with the estimates from the pose-scale graph.

5.5 Experiments

5.5.1 Experiment Setup

The endoscopic videos used in the experiments were acquired from seven consenting patients and four cadavers under an Institutional Review Boards (IRB)-approved protocol. The anatomy captured in the videos is the nasal cavity. The total time duration of videos is around 40 minutes. The input images to both networks are 8-time spatially downsampled, resulting in a resolution of 128×160; the output maps of both networks have a resolution of 64×80. Note that the binary masks with the same resolution are also fed, together with images, into the networks to exclude contributions of invalid pixels. SGD optimizer with cyclic learning rate scheduler [84] is used for network training, where the learning rate range is $[1.0\mathrm{e}^{-4}, 5.0\mathrm{e}^{-4}]$. The weights for scale-invariant loss, RR loss, flow loss, histogram loss, generator adversarial loss, and discriminator adversarial loss are 20.0, 4.0, 10.0, 4.0, 1.0, and 1.0. In terms of the

hyperparameters related to loss design, ϵ is 1.0e^{-4}; η_{hist} is 0.3; β is $\frac{4}{5K}$; K is 100; C is 16; H is 64; W is 80; B is 16;

Full-range rotation augmentation is used for input images to the networks during training. The first stage of training lasts for 40 epochs and the second stage lasts until the loss curves plateau, where each epoch consists of 300 iterations with the batch size of 1. Image pairs are selected so that the groundtruth ratio of scene overlap is larger than 0.6; the initialized relative pose is randomized so that the initial ratio of scene overlap is larger than 0.4.

In terms of the hyperparameters of the differentiable LM optimization, damp value range is $[1.0\text{e}^{-6}, 1.0\text{e}^{-2}]$, with 1.0e^{-4} as the initial value. The increasing and decreasing multiplier of the damp value is 11.0 and 9.0, respectively. LM optimization terminates when one of the three below is met: 1) number of iterations reaching 40, 2) maximum gradient smaller than 1.0e^{-4}, 3) maximum parameter increment ratio smaller than 1.0e^{-2}. Factors involved have the same parameter setting as the SLAM system, which will be described below.

Below are the hyperparameters of the SLAM system. For the *Camera Tracking* module, the multiplying factor used for the reference keyframe selection is 0.6; the maximum number of iterations in the optimization is 40; the damp value range is $[1.0\text{e}^{-6}, 1.0\text{e}^{-2}]$, with 1.0e^{-4} as the initial value; the increasing and decreasing multiplier is 100.0 and 10.0, respectively; the jacobian matrix recompute condition is when

the error update between steps is larger than $1.0\mathrm{e}^{-2}$ of the current error. As for factors in the *Camera Tracking* module, settings are as follows. In the Feature-metric Factor, all samples within the video mask are used for computation; the weights for all 4 pyramid levels (from high resolution to low one) are 10.0, 9.0, 8.0, and 7.0. In the Reprojection Factor, the factor weight and σ_{rp} are 0.1 and 0.03, respectively. In the Sparse Matched Geometry Factor, the factor weight and σ_{smg} are 0.1 and 0.1, respectively; the number of feature match candidates before filtering is 256; in terms of the Teaser++ filtering, the maximum clique time limit, rotation maximum iterations, rotation graph, inlier selection mode, and noise bound multiplier are $50\mathrm{ms}$, 20, chain mode, no inlier selection, and 2.0, respectively; Other parameters of Teaser++ are set to the default ones.

For the *Keyframe Creation* module, settings are as below. The maximum ratios of scene overlap in terms of the area and the number of point inliers within the video mask for a new keyframe are 0.8 and 0.9, respectively; the maximum feature match inlier ratio is 0.4; the minimum average magnitude of 2D flow is 0.08 of the image width. For the temporal connection building in the *Keyframe Creation* module, the maximum number of temporal connections per keyframe is 3; the minimum feature match inlier ratio to connect a previous keyframe is 0.7.

For the *Loop Closure* module, settings are shown as follows. For the local loop detection, the temporal window for searching is 9; the rotation and translation weights

to compute pose distance for candidate filtering are both set to 1.0; the spatial distance multiplier for candidate filtering is 5.0; the metric multiplier for verification is 0.7; the minimum constant inlier ratio for verification is 0.2, which is the same in global loop detection; the minimum ratios of scene overlap for verification in terms of the area and the number of point inliers within the video mask are 0.5 and 0.5, respectively.

Regarding the global loop detection, only keyframes that are at least 10 keyframes away are considered; the multiplier of description similarity for verification is 0.7; the metric multiplier for verification is 0.7; a global loop candidate will be selected if it is at least 10 keyframes away from the ones already selected in a single global loop closure process. In the pose-scale graph optimization for loop closure, the weights of the Relative Pose Scale Factor for non-global and global connections are 1.0 and 5.0, respectively; within this factor, the weights of rotation and scale component, which are ω_{rot} and ω_{scl}, are 5.0 and 0.5, respectively; the weight of the Scale Factor within the loop closure optimization is 10.0; the number of maximum iterations of such optimization is 200; the number of maximum iterations with no relinearization is 5; the relinearization thresholds for pose and scale are 3.0e^{-3} and 1.0e^{-2}.

For the *Mapping* module, settings are as follows. In terms of hyperparameters of factors used in the full factor graph, the weights for the Pose Factor and Scale Factor of the first keyframe are 1.0e^{4}, which are used to anchor the graph in terms of camera pose and depth scale; The Feature-metric Factor and Geometric Consistency Factor

use all samples within the video mask for computation; the Feature-metric Factor has the same weight as the one in camera tracking; the Geometric Consistency Factor has the factor weight of 0.1 and σ_{gc} as 0.03; the weight of the Code Factor is 1.0e^{-4}. In terms of the hyperparameters in factor graph optimization algorithm ISAM2 [250], the relinearization thresholds for camera poses, depth scales, and depth codes are 1.0e^{-3}, 1.0e^{-3}, and 1.0e^{-2}, respectively; partial relinearization check and relinearization skipping are not used; Other parameters in ISAM2 are set to the default ones.

In cases where post-operative processing in a SLAM system is allowed, the *Mapping* and *Loop Closure* modules can be run for an additional amount of time after all frames have been tracked. The *Mapping* module will continue refining the full factor graph. The maximum number of iterations and consecutive no-relinearization iterations are 20 and 5, respectively. In the meantime, the *Loop Closure* module will search for loop pairs for the query keyframes that have not been processed before. When the *Mapping* module finishes, the entire system run will end.

5.5.2 Evaluation Metrics

The metrics used for camera trajectory evaluation are absolute trajectory error (ATE) and relative pose error (RPE) [252]. Note that only the frames that are treated as keyframes by the SLAM system will be evaluated in terms of both trajectory error and depth error. Therefore, synchronization needs to be done to associate the trajec-

tory estimate with the groundtruth one. The trajectory estimate will also be spatially aligned with the pseudo groundtruth trajectory from SfM results in Chapter 2, before computing metrics. The transformation model used for spatial alignment is the similarity transform, and all poses are used to estimate such a transform with the method described in [252].

ATE is used to quantify the whole trajectory and here the form of Root Mean Square Error is used. The rotation and translation components of this metric are defined as

$$\begin{aligned} \mathrm{ATE}_{\mathrm{rot}} &= \left(\frac{1}{N} \sum_{i=0}^{N-1} \left\| \log\left(\boldsymbol{R}_i^{\mathrm{ATE}} \right) \right\|_2^2 \right)^{\frac{1}{2}} \quad \text{and} \\ \mathrm{ATE}_{\mathrm{trans}} &= \left(\frac{1}{N} \sum_{i=0}^{N-1} \left\| \boldsymbol{t}_i^{\mathrm{ATE}} \right\|_2^2 \right)^{\frac{1}{2}} \quad , \end{aligned} \tag{5.15}$$

where $\boldsymbol{R}_i^{\mathrm{ATE}} = \tilde{\boldsymbol{R}}_i^{\mathrm{wld}} \left(\boldsymbol{R}_i^{\mathrm{wld}} \right)^{\intercal}$ and $\boldsymbol{t}_i^{\mathrm{ATE}} = \tilde{\boldsymbol{t}}_i^{\mathrm{wld}} - \boldsymbol{R}_i \boldsymbol{t}_i^{\mathrm{wld}}$. $\tilde{\boldsymbol{R}}_i^{\mathrm{wld}} \in \mathrm{SO}(3)$ and $\tilde{\boldsymbol{t}}_i^{\mathrm{wld}} \in \mathbb{R}^3$ are the groundtruth rotation and translation components of the i^{th} pose in the trajectory, respectively, while $\boldsymbol{R}_i^{\mathrm{wld}} \in \mathrm{SO}(3)$ and $\boldsymbol{t}_i^{\mathrm{wld}} \in \mathbb{R}^3$ are the estimated ones. $N \in \mathbb{R}$ is the number of poses in the synchronized and aligned trajectory estimate.

RPE measures the local accuracy of the trajectory over a fixed frame interval $\Delta \in \mathbb{R}$. This measures the local drift of the trajectory, which is less affected by the loop closure and emphasizes more on the other components of the system. The rotation

and translation components of this metric are defined as

$$\begin{aligned} \mathrm{RPE}_{\mathrm{rot}} &= \left(\frac{1}{N-\Delta} \sum_{i=0}^{N-\Delta-1} \left\| \log\left(\boldsymbol{R}_i^{\mathrm{RPE}} \right) \right\|_2^2 \right)^{\frac{1}{2}} \quad \text{and} \\ \mathrm{RPE}_{\mathrm{trans}} &= \left(\frac{1}{N-\Delta} \sum_{i=0}^{N-\Delta-1} \left\| \boldsymbol{t}_i^{\mathrm{RPE}} \right\|_2^2 \right)^{\frac{1}{2}} \quad , \end{aligned} \tag{5.16}$$

$\boldsymbol{R}_i^{\mathrm{RPE}} \in \mathrm{SO}\left(3\right)$ and $\boldsymbol{t}_i^{\mathrm{RPE}} \in \mathbb{R}^3$ are the rotation and translation components of $\boldsymbol{T}_i^{\mathrm{RPE}} \in \mathrm{SE}\left(3\right)$, respectively; $\boldsymbol{T}_i^{\mathrm{RPE}}$ is the i^{th} RPE matrix, which is defined as

$$\boldsymbol{T}_i^{\mathrm{RPE}} = \left((\tilde{\boldsymbol{T}}_i^{\mathrm{wld}})^{-1} \tilde{\boldsymbol{T}}_{i+\Delta}^{\mathrm{wld}} \right)^{-1} \left((\boldsymbol{T}_i^{\mathrm{wld}})^{-1} \boldsymbol{T}_{i+\Delta}^{\mathrm{wld}} \right) \quad . \tag{5.17}$$

To evaluate depth estimates, Absolute Relative Difference and Threshold [101] are used. Before computing metrics, different pre-processing is applied for two sets of metrics, which are $\mathrm{ARD}_{\mathrm{traj}}$ and $\mathrm{Threshold}_{\mathrm{traj}}$, and $\mathrm{ARD}_{\mathrm{frame}}$ and $\mathrm{Threshold}_{\mathrm{frame}}$. For the former, the estimated depth per keyframe is re-scaled with the scale component in the similarity transform obtained from the trajectory alignment above. For the latter, the depth estimates are re-scaled so that each estimate has the same scale as the groundtruth one, where the same scaling method in Sec. 3.4.2 is used. In terms

of the definitions of these metrics, ARD is

$$\text{ARD} = \frac{1}{N}\sum_{i=0}^{N-1}\frac{1}{|\Omega_i|}\sum_{\boldsymbol{x}\in\Omega}\frac{|\boldsymbol{D}_i(\boldsymbol{x}) - \tilde{\boldsymbol{D}}_i(\boldsymbol{x})|}{\tilde{\boldsymbol{D}}_i(\boldsymbol{x})} \quad ; \tag{5.18}$$

Threshold is

$$\text{Threshold} = \frac{1}{N}\sum_{i=0}^{N-1}\frac{1}{|\Omega_i|}\sum_{\boldsymbol{x}\in\Omega}\mathbb{1}\left[\max\left(\frac{\boldsymbol{D}_i(\boldsymbol{x})}{\tilde{\boldsymbol{D}}_i(\boldsymbol{x})}, \frac{\tilde{\boldsymbol{D}}_i(\boldsymbol{x})}{\boldsymbol{D}_i(\boldsymbol{x})}\right) < \theta\right] \quad . \tag{5.19}$$

Note that Ω_i here is the region where both scaled depth estimate $\boldsymbol{D}_i \in \mathbb{R}^{1\times H\times W}$ and groundtruth depth $\tilde{\boldsymbol{D}}_i \in \mathbb{R}^{1\times H\times W}$, for the i^{th} synchronized keyframe, have valid depths; $\theta \in \mathbb{R}$ is the threshold used to determine if the depth ratio between the estimate and groundtruth is small enough.

5.5.3 Cross-Subject Evaluation

To evaluate the performance of the SLAM system on endoscopic videos from unseen subjects, we run a cross-validation study. Four models are trained with different train/test splits on the 11 subjects in total. With subjects named as consecutive numbers, the test splits for 4 models are $\{1, 2, 3\}$, $\{4, 5, 6\}$, $\{7, 8, 11\}$, and $\{8, 9, 10\}$, and the train splits for each model are the subjects left. For each subject, several video sequences are available. For evaluation, the proposed SLAM is run on each

testing video and generates estimates of camera poses and dense depth maps for all keyframes. Besides, we also compare against a state-of-the-art feature-based SLAM system, ORB-SLAM3 [54], which we evaluate on all videos at once and use the same set of metrics for evaluation. We adjust the parameters of ORB-SLAM3 so that more keypoint candidates are detected per frame. The evaluation metrics, in Table 5.1, are averaged over all the sequences within the corresponding test split for each of our trained model. Table 5.2 shows the results by averaging each metric over all the sequences for evaluation, where we conduct the paired t-test analysis between the proposed system and ORB-SLAM v3. The results with ***, **, and * stand for p-value smaller than 0.001, 0.01, and 0.05, respectively.

Note that, to make the metrics physically meaningful in terms of the values, we roughly scaled all SfM results before evaluation based on the average size of an adult's nasal cavity. The metric values between methods are not strictly comparable. This is because different sets of images within a sequence are used as keyframes by different methods. However, considering the large number of point samples that are used for computation, the values should approximately indicate the performance difference. Δ in Eq. 5.16 is set to 7 for our results; for ORB-SLAM v3, Δ is set so that the number of original video frames between $\boldsymbol{T}_i^{\mathrm{wld}}$ and $\boldsymbol{T}_{i+\Delta}^{\mathrm{wld}}$ is roughly the same.

Subjects	$\{1,2,3\}$		$\{4,5,6\}$		$\{7,8,11\}$		$\{8,9,10\}$	
Methods / Metrics	Ours	ORB-SLAM v3 [54]	Ours	$\sim$	Ours	$\sim$	Ours	$\sim$
ATE_{trans}(mm)	$\mathbf{1.4 \pm 1.0}$	3.8 ± 2.7	$\mathbf{1.3 \pm 1.7}$	3.8 ± 4.6	$\mathbf{2.2 \pm 1.2}$	6.3 ± 4.8	$\mathbf{1.6 \pm 1.0}$	5.5 ± 3.0
ATE_{rot}(°)	$\mathbf{19.7 \pm 7.8}$	66.2 ± 59.5	$\mathbf{22.8 \pm 17.2}$	61.1 ± 68.1	$\mathbf{25.3 \pm 18.4}$	66.9 ± 48.9	$\mathbf{19.4 \pm 9.5}$	55.8 ± 22.4
RPE_{trans}(mm)	$\mathbf{1.3 \pm 0.4}$	2.5 ± 1.4	$\mathbf{1.4 \pm 0.7}$	2.7 ± 2.1	$\mathbf{1.9 \pm 0.6}$	4.8 ± 3.5	$\mathbf{1.2 \pm 0.5}$	3.6 ± 1.6
RPE_{rot}(°)	$\mathbf{5.9 \pm 1.7}$	6.4 ± 3.5	4.3 ± 2.0	$\mathbf{3.8 \pm 2.6}$	$\mathbf{7.4 \pm 2.6}$	7.7 ± 3.9	$\mathbf{4.5 \pm 1.1}$	8.5 ± 2.9
ARD_{traj}	$\mathbf{0.39 \pm 0.17}$	1.73 ± 1.02	$\mathbf{0.34 \pm 0.10}$	2.00 ± 1.82	$\mathbf{0.38 \pm 0.14}$	1.58 ± 1.42	$\mathbf{0.29 \pm 0.09}$	1.56 ± 1.20
ARD_{frame}	$\mathbf{0.17 \pm 0.04}$	1.73 ± 1.02	$\mathbf{0.17 \pm 0.04}$	2.00 ± 1.82	$\mathbf{0.18 \pm 0.03}$	1.58 ± 1.42	$\mathbf{0.15 \pm 0.02}$	1.56 ± 1.20
$Threshold_{traj}$ ($\theta = 1.25$)	$\mathbf{0.39 \pm 0.19}$	0.15 ± 0.13	$\mathbf{0.46 \pm 0.14}$	0.24 ± 0.21	$\mathbf{0.38 \pm 0.15}$	0.14 ± 0.14	$\mathbf{0.49 \pm 0.13}$	0.14 ± 0.15
$Threshold_{frame}$ ($\theta = 1.25$)	$\mathbf{0.39 \pm 0.19}$	0.15 ± 0.13	$\mathbf{0.46 \pm 0.14}$	0.24 ± 0.21	$\mathbf{0.38 \pm 0.15}$	0.14 ± 0.14	$\mathbf{0.49 \pm 0.13}$	0.14 ± 0.15
$Threshold_{traj}$ ($\theta = 1.25^2$)	$\mathbf{0.70 \pm 0.22}$	0.28 ± 0.22	$\mathbf{0.81 \pm 0.13}$	0.38 ± 0.29	$\mathbf{0.66 \pm 0.16}$	0.27 ± 0.23	$\mathbf{0.84 \pm 0.10}$	0.27 ± 0.22
$Threshold_{frame}$ ($\theta = 1.25^2$)	$\mathbf{0.70 \pm 0.22}$	0.28 ± 0.22	$\mathbf{0.81 \pm 0.13}$	0.38 ± 0.29	$\mathbf{0.66 \pm 0.16}$	0.27 ± 0.23	$\mathbf{0.84 \pm 0.10}$	0.27 ± 0.22

Table 5.1: **Cross-subject evaluation on SLAM systems per test split.** Note that $\sim$ is used as the name abbreviation of the comparison method.

Metrics / Methods	ATE_{trans} (mm)	ATE_{rot} (°)	RPE_{trans} (mm)	RPE_{rot} (°)	ARD_{traj}	ARD_{frame}	$Threshold_{traj}$ ($\theta = 1.25$)	$Threshold_{frame}$ ($\theta = 1.25$)	$Threshold_{traj}$ ($\theta = 1.25^2$)	$Threshold_{frame}$ ($\theta = 1.25^2$)
Ours	$\mathbf{1.6 \pm 1.4}$	$\mathbf{22.2 \pm 15.1}$	$\mathbf{1.5 \pm 0.6}$	$\mathbf{5.5 \pm 2.4}$	$\mathbf{0.36 \pm 0.16}$	$\mathbf{0.17 \pm 0.03}$	$\mathbf{0.42 \pm 0.17}$	$\mathbf{0.73 \pm 0.08}$	$\mathbf{0.74 \pm 0.21}$	$\mathbf{0.95 \pm 0.04}$
ORB-SLAM v3 [54]	$4.7 \pm 4.2^{***}$	$62.5 \pm 55.5^{***}$	$3.5 \pm 2.5^{***}$	6.3 ± 3.6	$1.76 \pm 1.49^{***}$	$24.27 \pm 42.07^{**}$	$0.17 \pm 0.18^{***}$	$0.37 \pm 0.13^{***}$	$0.31 \pm 0.25^{***}$	$0.56 \pm 0.15^{***}$

Table 5.2: **Cross-subject evaluation on SLAM systems.**

FT	FM	RT	Local	Global	$\mathrm{ATE}_{\mathrm{trans}}(\mathrm{mm})$	$\mathrm{ATE}_{\mathrm{rot}}(°)$	$\mathrm{RPE}_{\mathrm{trans}}(\mathrm{mm})$	$\mathrm{RPE}_{\mathrm{rot}}(°)$
✓	✓	✓	✓	✓	1.6 ± 1.4	22.2 ± 15.1	1.5 ± 0.6	5.5 ± 2.4
	✓	✓	✓	✓	$3.4 \pm 2.7^{***}$	$43.3 \pm 27.9^{***}$	$2.6 \pm 1.4^{***}$	$7.3 \pm 3.0^{***}$
		✓	✓	✓	$3.3 \pm 2.8^{***}$	$40.2 \pm 23.6^{***}$	$2.6 \pm 1.2^{***}$	$7.0 \pm 2.6^{***}$
✓	✓		✓	✓	2.7 ± 5.5	23.8 ± 14.5	2.1 ± 3.2	5.3 ± 2.1
✓	✓	✓	✓		$2.0 \pm 1.9^{*}$	$26.8 \pm 21.2^{*}$	1.5 ± 0.7	5.5 ± 2.4
✓	✓	✓			$2.0 \pm 1.9^{*}$	$25.5 \pm 18.5^{*}$	1.5 ± 0.7	5.4 ± 2.4

Table 5.3: **Ablation study for the SLAM system on trajectory-related metrics.** FT, FM, RT, Local, Global stand for the Feature-metric Factor in the *Camera Tracking* module, the Feature-metric Factor in the *Mapping* module, the Reprojection Factor in the *Camera Tracking* module, local loop detection in the *Loop Closure* module, and global loop detection and closure in the *Loop Closure* module, respectively. We conduct the paired t-test analysis for results of all the sequences between an ablation run and the standard run shown in the first row of this table. The results with ***, **, and * stand for p-value smaller than 0.001, 0.01, and 0.05, respectively. As can be seen, the Feature-metric Factor has a large impact on both trajectory and trajectory-scaled depth metrics; the Reprojection Factor mainly affects trajectory metrics; the *Loop Closure* module mainly affects the trajectory metrics $\mathrm{ATE}_{\mathrm{trans}}$ and $\mathrm{ATE}_{\mathrm{rot}}$.

5.5.4 Ablation Study

We evaluate the contributions of several components in the SLAM system by disabling some components in different runs. The components for ablation are the Feature-metric Factor in the *Camera Tracking* and *Mapping* modules, Reprojection Factor in the *Camera Tracking* module, local loop detection in the *Loop Closure* module, and global loop detection and closure in the *Loop Closure* module. All metrics described in Sec. 5.5.3 are evaluated in this ablation study. The results are shown in Table 5.3 and 5.4. Note that the value of each metric is averaged over all the sequences from all subjects, where each subset of the sequences is evaluated with the corresponding trained model so that all the sequences are unseen during training.

FT	FM	RT	Local	Global	ARD_{traj}	$\text{ARD}_{\text{frame}}$	$\text{Threshold}_{\text{traj}}$ ($\theta = 1.25$)	$\text{Threshold}_{\text{frame}}$ ($\theta = 1.25$)	$\text{Threshold}_{\text{traj}}$ ($\theta = 1.25^2$)	$\text{Threshold}_{\text{frame}}$ ($\theta = 1.25^2$)
✓	✓	✓	✓	✓	0.36 ± 0.16	0.17 ± 0.03	0.42 ± 0.17	0.73 ± 0.08	0.74 ± 0.21	0.95 ± 0.04
	✓	✓	✓	✓	$0.49 \pm 0.19^{***}$	0.17 ± 0.03	$0.29 \pm 0.16^{***}$	0.73 ± 0.08	$0.59 \pm 0.23^{***}$	0.95 ± 0.04
		✓	✓	✓	$0.50 \pm 0.25^{**}$	0.17 ± 0.03	$0.32 \pm 0.17^{**}$	0.74 ± 0.08	$0.61 \pm 0.24^{**}$	0.95 ± 0.04
✓	✓		✓	✓	0.35 ± 0.15	0.17 ± 0.03	0.43 ± 0.17	0.73 ± 0.08	0.76 ± 0.18	0.95 ± 0.04
✓	✓	✓	✓		0.36 ± 0.16	0.17 ± 0.03	0.42 ± 0.17	0.73 ± 0.08	0.74 ± 0.21	0.95 ± 0.04
✓	✓	✓			$0.35 \pm 0.16^{*}$	0.17 ± 0.03	0.42 ± 0.18	0.73 ± 0.08	0.74 ± 0.22	0.95 ± 0.04

Table 5.4: **Ablation study for the SLAM system on depth-related metrics.** The settings and notations are the same as Table 5.3.

5.5.5 Evaluation with CT

This study uses the residual error metric described in Sec. 3.6.2. Before computing the residual error, several pre-processing steps are required. First, the method described in Sec. 3.5.2 and 3.5.3 is applied to obtain a surface reconstruction from the depth maps and camera poses estimated by the proposed SLAM system. The slope in the truncated signed distance function is constant instead of the depth uncertainty which is used in Sec. 3.5.2. Then a point cloud registration algorithm based on [120] is applied between the surface reconstruction and the CT surface model, where a similarity transform is estimated. Note that before the registration, a manual initial alignment between these two models is applied. After the registration finishes, the residual error is computed between the registered surface reconstruction and the CT surface model.

In this study, we evaluate the accuracy of surface reconstructions from the videos of the four cadavers, where for each subject, the metrics of all the sequences are

averaged over to report here. The average residual errors for subject 7, 9, 10, and 11 are 0.83, 0.88, 0.78, and 0.86 mm, respectively.

5.6 Discussion

The accuracy of trajectory estimation depends on how consistent and distinctive the feature and descriptor maps are, as well as the accuracy of the depth estimates. Though the depth is optimizable during the SLAM running, the depth basis maps estimated from the input image still bound the variation mode of the final depth estimate. Therefore, if the depth network is not familiar with the scene, it is probable that a depth estimate close to the truth will not be obtained. Therefore, a representative collection of training data is crucial for the generalizability of such a learning-based SLAM system.

For the current system that is trained on sinus endoscopy dataset, we would expect the system generalizes decently to endoscopy on tubular structures, such as bronchoscopy. It could be less generalizable to domain such as laparoscopy because the overall geometry of the anatomy is unseen for the depth network. However, we would still expect the appearance representation to generalize well to these more distant cases because it only models texture instead of geometry. It is also expected the system can generalize even better if the capacity of the network is configured to be larger

with a larger dataset for training. The networks do not produce uncertainty estimates for now, and those could potentially further improve the generalizability and benefit the factor graph optimization if being accurate.

Currently, the system cannot recover from a spurious global loop connection and therefore the global loop detection criteria need to be strict to keep the false positive rate to zero. Such an error could potentially be detected by monitoring the overall objective of the full factor graph after each global loop closure [223]. For now, camera relocalization after tracking failure is not implemented and is likely required in cases where images with bad conditions (*e.g.*, image blurring) happen often, such as laparoscopy; a method similar to the searching in the global loop detection could be used. A keyframe culling method can be implemented to reduce the number of keyframes to reduce the memory requirement and accelerate the computation. A method similar to the reference keyframe selection in the Camera Tracking module may be used to find redundant keyframes.

The proposed SLAM system is currently designed for static scenes. Nevertheless, having additional optimization variables per keyframe to model geometry deformation could potentially make the system suitable for a deformable environment. As for the robustness to a dynamic environment with changing textures (*e.g.*, changing illumination and bleeding), it depends on whether there are similar conditions in the training dataset and how large the affected regions are within images. For example,

if the bleeding region is small relative to the entire image, even if the blood moves during the video capturing, the impact should be minimal because factors, such as Feature-metric Factor, will focus on the large unaffected regions during optimization.

5.7 Conclusion

In this chapter, we propose a SLAM system that is robust to texture-scarce scenarios with learning-based appearance and geometric representations. An effective training scheme is developed to learn such representations that are suitable for the SLAM system run. In the experiments, we show that the proposed system performs favorably compared with a state-of-the-art feature-based SLAM system in terms of the accuracy of both camera trajectories and geometry estimates.

The proposed SLAM system currently only works in the static environment. However, it is feasible to add another type of optimization variable to factor graph optimization to take care of the tissue deformation, such as the deformation-spline used in [253], and therefore worthy of working on as a future direction. Similar to the directions in Chapter 2, it is also worth exploring how to make such a SLAM system work in scenarios where the topology of anatomy is changed due to surgical operations. An additional map for each keyframe to notate which part of the region is unaffected could be one way to achieve this. Currently, the global loop closure needs

to have a zero false-positive rate to have reasonable performance, it is desired to have a failure-aware and recovery mechanism to relax such a constraint. Based on our observation, having enough global loop closures is critical for accurate estimation of the camera trajectory, a study on what scoping path is the best for each type of endoscopy could be worth exploring. Currently, for the robustness of the system, the local connection only considers the spatially closest pairs because a keyframe connection with a small scene overlap could be erroneous. Having accurate mid-range connections could further improve the performance and reduce the drifting errors even when no global loops are available, which was observed in [54].

For the task of surface reconstruction and endoscope tracking from a video with pre-operative model alignment, with the works developed in this thesis, there are in general one retrospective pipeline and one online one. The retrospective pipeline can already be fully built with the works described in this thesis, as described in Sec. 1.2.3. For the online pipeline described in this chapter, however, if an automatic alignment between the pre-operative and intra-operative surface model with iterative refinement is needed, some additional works are required. The registration method developed in Chapter 4 considers one-time model alignment and did not exploit the fact that a real-time SLAM system updates the map whenever a new scene is observed. Integrating such surface update into the optimization step of a registration method could potentially further improve the registration performance in terms of ac-

curacy and processing speed. Besides, to align the pre-operative and intra-operative models during a SLAM system run, a single surface model for the entire observed environment needs to be built and updated. In this work, the dense depth estimates are not fused during the system run and a real-time depth fusion and surface extraction method needs to be developed to obtain such a surface model.

Bibliography

[1] P. C. De Groen, "History of the endoscope [scanning our past]," *Proceedings of the IEEE*, vol. 105, no. 10, pp. 1987–1995, 2017.

[2] C. Nezhat, "History of endoscopy," http://sls.org/nezhats-history-of-endoscopy/, 2005.

[3] K. Ball, *Endoscopic Surgery*, ser. Mosby's perioperative nursing series. Mosby, 1997.

[4] C. J. Powers, "A brief history of endoscopy," *Seminars in Perioperative Nursing*, vol. 2, no. 3, pp. 129–132, Jul. 1993.

[5] B. I. Hirschowitz, "A personal history of the fiberscope," *Gastroenterology*, vol. 76, no. 4, pp. 864–869, Apr. 1979.

[6] M. Finocchiaro, P. Cortegoso Valdivia, A. Hernansanz, N. Marino, D. Amram, A. Casals, A. Menciassi, W. Marlicz, G. Ciuti, and A. Koulaouzidis, "Training

simulators for gastrointestinal endoscopy: Current and future perspectives," *Cancers*, vol. 13, no. 6, Mar. 2021.

[7] L. X. Harrington, J. W. Wei, A. A. Suriawinata, T. A. Mackenzie, and S. Hassanpour, "Predicting colorectal polyp recurrence using time-to-event analysis of medical records," *AMIA Jt Summits Transl Sci Proc*, vol. 2020, pp. 211–220, May 2020.

[8] W. Hong, J. Wang, F. Qiu, A. Kaufman, and J. Anderson, "Colonoscopy simulation," in *Medical Imaging 2007: Physiology, Function, and Structure from Medical Images*, vol. 6511. International Society for Optics and Photonics, 2007, p. 65110R.

[9] P. Fockens, "Endoscopic management of perforations in the gastrointestinal tract," *Gastroenterol. Hepatol.*, vol. 12, no. 10, pp. 641–643, Oct. 2016.

[10] R. Eliashar, J.-Y. Sichel, M. Gross, E. Hocwald, I. Dano, A. Biron, A. Ben-Yaacov, A. Goldfarb, and J. Elidan, "Image guided navigation system-a new technology for complex endoscopic endonasal surgery," *Postgrad. Med. J.*, vol. 79, no. 938, pp. 686–690, Dec. 2003.

[11] M. Hytönen, K. Blomgren, M. Lilja, and A. Mäkitie, "How we do it: septoplasties under local anaesthetic are suitable for short stay surgery; the clinical outcomes." *Clinical otolaryngology: official journal of ENT-UK; official journal*

of Netherlands Society for Oto-Rhino-Laryngology & Cervico-Facial Surgery, vol. 31, no. 1, pp. 64–68, 2006.

[12] T. F. P. Bezerra, M. G. Stewart, M. A. Fornazieri, R. R. de Mendonca Pilan, F. de Rezende Pinna, F. G. de Melo Padua, and R. L. Voegels, "Quality of life assessment septoplasty in patients with nasal obstruction," *Brazilian journal of otorhinolaryngology*, vol. 78, no. 3, pp. 57–62, 2012.

[13] A. L. Feng, C. R. Razavi, P. Lakshminarayanan, Z. Ashai, K. Olds, M. Balicki, Z. Gooi, A. T. Day, R. H. Taylor, and J. D. Richmon, "The robotic ent microsurgery system: a novel robotic platform for microvascular surgery," *The Laryngoscope*, vol. 127, no. 11, pp. 2495–2500, 2017.

[14] J. J. McGoran, M. E. McAlindon, P. G. Iyer, E. J. Seibel, R. Haidry, L. B. Lovat, and S. S. Sami, "Miniature gastrointestinal endoscopy: Now and the future," *World J. Gastroenterol.*, vol. 25, no. 30, pp. 4051–4060, Aug. 2019.

[15] X. Liu, Y. Zheng, B. Killeen, M. Ishii, G. D. Hager, R. H. Taylor, and M. Unberath, "Extremely dense point correspondences using a learned feature descriptor," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4847–4856.

[16] X. Liu, A. Sinha, M. Ishii, G. D. Hager, A. Reiter, R. H. Taylor, and M. Unberath,

"Dense depth estimation in monocular endoscopy with self-supervised learning methods," *IEEE transactions on medical imaging*, 2019.

[17] X. Liu, M. Stiber, J. Huang, M. Ishii, G. D. Hager, R. H. Taylor, and M. Unberath, "Reconstructing sinus anatomy from endoscopic video – towards a Radiation-Free approach for quantitative longitudinal assessment," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*. Springer International Publishing, 2020, pp. 3–13.

[18] X. Liu, B. D. Killeen, A. Sinha, M. Ishii, G. D. Hager, R. H. Taylor, and M. Unberath, "Neighborhood normalization for robust geometric feature learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 13 049–13 058.

[19] J. L. Schönberger and J. Frahm, "Structure-from-motion revisited," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 4104–4113.

[20] R. Roberts, S. N. Sinha, R. Szeliski, and D. Steedly, "Structure from motion for scenes with large duplicate structures," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3137–3144, 2011.

[21] N. Jiang, P. Tan, and L. F. Cheong, "Seeing double without confusion: Structure-

from-motion in highly ambiguous scenes," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1458–1465, 2012.

[22] C. Sweeney, T. Sattler, T. Hollerer, M. Turk, and M. Pollefeys, "Optimizing the viewing graph for structure-from-motion," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, pp. 801–809, 2015.

[23] C. Kong and S. Lucey, "Deep non-rigid structure from motion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1558–1567.

[24] C. Wang, C.-H. Lin, and S. Lucey, "Deep NRSfM++: Towards unsupervised 2D-3D lifting in the wild," in *2020 International Conference on 3D Vision (3DV)*, Nov. 2020, pp. 12–22.

[25] K. L. Lurie, R. Angst, D. V. Zlatev, J. C. Liao, and A. K. Ellerbee Bowden, "3D reconstruction of cystoscopy videos for comprehensive bladder records," *Biomed. Opt. Express*, vol. 8, no. 4, pp. 2106–2123, Apr. 2017.

[26] Q. Péntek, S. Hein, A. Miernik, and A. Reiterer, "Image-based 3d surface approximation of the bladder using structure-from-motion for enhanced cystoscopy based on phantom data," *Biomedical Engineering / Biomedizinische Technik*, vol. 63, pp. 461 – 466, 2018.

[27] T.-B. Phan, D.-H. Trinh, D. Lamarque, D. Wolf, and C. Daul, "Dense optical flow for the reconstruction of weakly textured and structured surfaces: Application to endoscopy," in *2019 IEEE International Conference on Image Processing (ICIP)*, Sep. 2019, pp. 310–314.

[28] A. R. Widya, Y. Monno, M. Okutomi, S. Suzuki, T. Gotoda, and K. Miki, "Whole stomach 3d reconstruction and frame localization from monocular endoscope video," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 7, pp. 1–10, 2019.

[29] A. R. Widya, Y. Monno, M. Okutomi, S. Suzuki, T. Gotoda, and K. Miki, "Stomach 3d reconstruction based on virtual chromoendoscopic image generation," *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 1848–1852, 2020.

[30] T.-B. Phan, D.-H. Trinh, D. Wolf, and C. Daul, "Optical flow-based structure-from-motion for the reconstruction of epithelial surfaces," *Pattern Recognit.*, vol. 105, no. 107391, p. 107391, Sep. 2020.

[31] C. Harris and M. Stephens, "A combined corner and edge detector," in *In Proc. of Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[32] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proceedings of the 9th European Conference on Computer Vision - Vol-*

ume Part I, ser. ECCV'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 430–443.

[33] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[34] R. Arandjelovic, "Three things everyone should know to improve object retrieval," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ser. CVPR '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 2911–2918.

[35] A. Bursuc, G. Tolias, and H. Jégou, "Kernel local descriptors with implicit rotation matching," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, ser. ICMR '15. New York, NY, USA: ACM, 2015, pp. 595–598.

[36] J. Dong and S. Soatto, "Domain-size pooling in local descriptors: Dsp-sift," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5097–5106, 2014.

[37] J.-W. Bian, Y.-H. Wu, J. Zhao, Y. Liu, L. Zhang, M.-M. Cheng, and I. Reid, "An evaluation of feature matchers for fundamental matrix estimation," in *British Machine Vision Conference (BMVC)*, 2019.

[38] J. L. Schonberger, H. Hardmeier, T. Sattler, and M. Pollefeys, "Comparative evaluation of hand-crafted and learned local features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1482–1491.

[39] Y. Tian, B. Fan, and F. Wu, "L2-net: Deep learning of discriminative patch descriptor in euclidean space," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 661–669.

[40] Z. Luo, T. Shen, L. Zhou, S. Zhu, R. Zhang, Y. Yao, T. Fang, and L. Quan, "Geodesc: Learning local descriptors by integrating geometry constraints," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 168–183.

[41] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, "Working hard to know your neighbor's margins: Local descriptor learning loss," in *Advances in Neural Information Processing Systems*, 2017, pp. 4826–4837.

[42] E. Tola, V. Lepetit, and P. Fua, "Daisy: An efficient dense descriptor applied to wide-baseline stereo," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 5, pp. 815–830, 2009.

[43] C. B. Choy, J. Gwak, S. Savarese, and M. Chandraker, "Universal correspon-

dence network," in *Advances in Neural Information Processing Systems*, 2016, pp. 2414–2422.

[44] H. Liao, W. Lin, J. Zhang, J. Zhang, J. Luo, and S. K. Zhou, "Multiview 2d/3d rigid registration via a point-of-interest network for tracking and triangulation," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019, pp. 12 638–12 647.

[45] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, "D2-net: A trainable cnn for joint detection and description of local features," in *Proceedings of the 2019 IEEE / CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[46] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 224–236.

[47] P. Truong, S. Apostolopoulos, A. Mosinska, S. Stucky, C. Ciller, and S. D. Zanet, "Glampoints: Greedily learned accurate match points," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 10 732–10 741.

[48] S. Leonard, A. Sinha, A. Reiter, M. Ishii, G. L. Gallia, R. H. Taylor *et al.*, "Evaluation and stability analysis of video-based navigation system for functional

endoscopic sinus surgery on in vivo clinical data," *IEEE_J_MI*, vol. 37, no. 10, pp. 2185–2195, Oct. 2018.

[49] O. G. Grasa, E. Bernal, S. Casado, I. Gil, and J. Montiel, "Visual slam for handheld monocular endoscope," *IEEE transactions on medical imaging*, vol. 33, no. 1, pp. 135–146, 2013.

[50] N. Mahmoud, I. Cirauqui, A. Hostettler, C. Doignon, L. Soler, J. Marescaux, and J. M. M. Montiel, "Orbslam-based endoscope tracking and 3d reconstruction," in *CARE@MICCAI*, 2016.

[51] L. Qiu and H. Ren, "Endoscope navigation and 3d reconstruction of oral cavity by visual slam with mitigated data scarcity," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2018, pp. 2278–22 787.

[52] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[53] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[54] C. Campos, R. Elvira, J. J. Gómez, J. M. M. Montiel, and J. D. Tardós, "ORB-

SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM," *arXiv preprint*, 2020.

[55] I. Khan, "Robust sparse and dense nonrigid structure from motion," *IEEE Transactions on Multimedia*, vol. 20, no. 4, pp. 841–850, April 2018.

[56] J. Lamarca, S. Parashar, A. Bartoli, and J. Montiel, "Defslam: Tracking and mapping of deforming scenes from monocular sequences," *IEEE Transactions on robotics*, vol. 37, no. 1, pp. 291–303, 2020.

[57] J. Song, J. Wang, L. Zhao, S. Huang, and G. Dissanayake, "Mis-slam: Real-time large-scale dense deformable slam system in minimal invasive surgery based on heterogeneous computing," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4068–4075, 2018.

[58] S. H. N. Jensen, M. E. B. Doest, H. Aanæs, and A. Del Bue, "A benchmark and evaluation of non-rigid structure from motion," *International Journal of Computer Vision*, vol. 129, no. 4, pp. 882–899, 2021.

[59] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8958–8966.

[60] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle ad-

justment - a modern synthesis," in *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ser. ICCV '99. Berlin, Heidelberg: Springer-Verlag, 1999, p. 298–372.

[61] Y. Lou, N. Snavely, and J. Gehrke, "MatchMiner: Efficient spanning structure mining in large image collections," in *Computer Vision – ECCV 2012*. Springer Berlin Heidelberg, 2012, pp. 45–58.

[62] M. Havlena and K. Schindler, "VocMatch: Efficient multiview correspondence for structure from motion," in *Computer Vision – ECCV 2014*. Springer International Publishing, 2014, pp. 46–60.

[63] J. L. Schönberger, A. C. Berg, and J.-M. Frahm, "PAIGE: PAirwise image geometry encoding for improved efficiency in Structure-from-Motion," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 1009–1018.

[64] J. Heinly, J. L. Schönberger, E. Dunn, and J.-M. Frahm, "Reconstructing the world* in six days," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 3287–3295.

[65] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[66] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[67] C. Beder and R. Steffen, "Determining an initial image pair for fixing the scale of a 3D reconstruction from an image sequence," in *Lecture Notes in Computer Science*, ser. Lecture notes in computer science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 657–666.

[68] Y. Zheng, Y. Kuang, S. Sugimoto, K. Åström, and M. Okutomi, "Revisiting the PnP problem: A fast, general and optimal solution," in *2013 IEEE International Conference on Computer Vision*, Dec. 2013, pp. 2344–2351.

[69] M. Bujnak, Z. Kukelova, and T. Pajdla, "A general solution to the P4P problem for camera with unknown focal length," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Jun. 2008.

[70] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment — a modern synthesis," in *Vision Algorithms: Theory and Practice*. Springer Berlin Heidelberg, 2000, pp. 298–372.

[71] F. Lu and R. Hartley, "A fast optimal algorithm for L2 triangulation," in *Computer Vision – ACCV 2007*. Springer Berlin Heidelberg, 2007, pp. 279–288.

[72] Li, "A practical algorithm for L triangulation with outliers," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 0, Jun. 2007, pp. 1–8.

[73] C. Aholt, S. Agarwal, and R. Thomas, "A QCQP approach to triangulation," in *Computer Vision – ECCV 2012*. Springer Berlin Heidelberg, 2012, pp. 654–667.

[74] G. P. Meyer, "An alternative probabilistic interpretation of the huber loss," *ArXiv*, vol. abs/1911.02088, 2019.

[75] J. J. Moré, "The Levenberg-Marquardt algorithm: Implementation and theory," in *Numerical Analysis*. Springer Berlin Heidelberg, 1978, pp. 105–116.

[76] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 11–19.

[77] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "Normface: l 2 hypersphere embedding for face verification," in *Proceedings of the 25th ACM international conference on Multimedia*. ACM, 2017, pp. 1041–1049.

[78] S. Honari, P. Molchanov, S. Tyree, P. Vincent, C. Pal, and J. Kautz, "Improving landmark localization with semi-supervised learning," in *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1546–1555.

[79] D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 20, no. 2, pp. 215–232, 1958.

[80] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[81] C. Strecha, W. Von Hansen, L. Van Gool, P. Fua, and U. Thoennessen, "On benchmarking camera calibration and multi-view stereo for high resolution imagery," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. Ieee, 2008, pp. 1–8.

[82] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, pp. 8026–8037, 2019.

[83] H. Robbins, "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 2007.

[84] L. N. Smith, "Cyclical learning rates for training neural networks," in *2017*

IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2017, pp. 464–472.

[85] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[86] G. A. Puerto-Souza and G. L. Mariottini, "Hierarchical multi-affine (hma) algorithm for fast and accurate feature matching in minimally-invasive surgical images," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 2007–2012.

[87] P. Moulon, P. Monasse, R. Perrot, and R. Marlet, "Openmvg: Open multiple view geometry," in *International Workshop on Reproducible Research in Pattern Recognition*. Springer, 2016, pp. 60–74.

[88] A. Baumberg, "Reliable feature matching across widely separated views," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, vol. 1, 2000, pp. 774–781 vol.1.

[89] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[90] C. Bregler, A. Hertzmann, and H. Biermann, "Recovering non-rigid 3D shape

from image streams," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 690–696, 2000.

[91] Y. Dai, H. Li, and M. He, "A simple prior-free method for non-rigid structure-from-motion factorization," *Int. J. Comput. Vis.*, vol. 107, no. 2, pp. 101–122, 2014.

[92] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96*. New York, New York, USA: ACM Press, 1996.

[93] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *2016 4th Int. Conf. 3D Vis.*, Oct. 2016, pp. 239–248.

[94] M. Visentini-Scarzanella, T. Sugiura, T. Kaneko, and S. Koto, "Deep monocular 3d reconstruction for assisted navigation in bronchoscopy," *International journal of computer assisted radiology and surgery*, vol. 12, no. 7, pp. 1089–1099, 2017.

[95] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[96] F. Mahmood and N. J. Durr, "Deep learning and conditional random fields-based depth estimation and topographical reconstruction from conventional endoscopy," *Medical image analysis*, vol. 48, pp. 230–243, 2018.

[97] S.-P. Yang, J.-J. Kim, K.-W. Jang, W.-K. Song, and K.-H. Jeong, "Compact stereo endoscopic camera using microprism arrays," *Optics letters*, vol. 41, no. 6, pp. 1285–1288, 2016.

[98] M. Simi, M. Silvestri, C. Cavallotti, M. Vatteroni, P. Valdastri, A. Menciassi *et al.*, "Magnetically activated stereoscopic vision system for laparoendoscopic single-site surgery," *IEEE_J_MECH*, vol. 18, no. 3, pp. 1140–1151, 2013.

[99] R. Garg, V. K. BG, G. Carneiro, and I. Reid, "Unsupervised cnn for single view depth estimation: Geometry to the rescue," in *Comput. Vis. ECCV*. Springer, 2016, pp. 740–756.

[100] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *2017 IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 6612–6619.

[101] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *2018 IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1983–1992.

[102] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5667–5675.

[103] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Adv. Neural Inf. Process. Syst. 27*. Curran Associates, Inc., 2014, pp. 2366–2374.

[104] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proceedings of the fourth Eurographics symposium on Geometry processing*, vol. 7, 2006.

[105] M. Turan, Y. Y. Pilavci, I. Ganiyusufoglu, H. Araujo, E. Konukoglu, and M. Sitti, "Sparse-then-dense alignment-based 3d map reconstruction method for endoscopic capsule robots," *Machine Vision and Applications*, vol. 29, no. 2, pp. 345–359, 2018.

[106] H. N. Tokgozoglu, E. M. Meisner, M. Kazhdan, and G. D. Hager, "Color-based hybrid reconstruction for endoscopy," in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, June 2012, pp. 8–15.

[107] A. Karargyris and N. Bourbakis, "Three-dimensional reconstruction of the di-

gestive wall in capsule endoscopy videos using elastic video interpolation," *IEEE Transactions on Medical Imaging*, vol. 30, no. 4, pp. 957–971, April 2011.

[108] Q. Zhao, T. Price, S. Pizer, M. Niethammer, R. Alterovitz, and J. Rosenman, "The endoscopogram: A 3d model reconstructed from endoscopic video frames," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, and W. Wells, Eds. Cham: Springer International Publishing, 2016, pp. 439–447.

[109] N. Mahmoud, A. Hostettler, T. Collins, L. Soler, C. Doignon, and J. Montiel, "Slam based quasi dense reconstruction for minimally invasive surgery scenes," *arXiv preprint*, 2017.

[110] R. Ma, R. Wang, S. Pizer, J. Rosenman, S. K. McGill, and J.-M. Frahm, "Real-time 3d reconstruction of colonoscopic surfaces for determining missing regions," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 573–582.

[111] R. J. Chen, T. L. Bobrow, T. Athey, F. Mahmood, and N. J. Durr, "Slam endoscopy enhanced by adversarial depth prediction," *arXiv preprint*, 2019.

[112] X. Liu, A. Sinha, M. Unberath, M. Ishii, G. Hager, R. Taylor *et al.*, "Self-supervised learning for dense depth estimation in monocular endoscopy," in *OR*

2.0 Context Aware Oper. Theaters Comput. Assist. Robot. Endosc. Clin. Image Based Proced. Skin Image Anal. Springer Verlag, 2018, pp. 128–138.

[113] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. 2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1. Washington, DC, USA: IEEE Computer Society, 2005, pp. 539–546.

[114] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, vol. 1, no. 10, p. e3, 2016.

[115] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, vol. 2. Cambridge, MA, USA: MIT Press, 2015, pp. 2017–2025.

[116] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods," *Mach. Learn.*, vol. 110, no. 3, pp. 457–506, Mar. 2021.

[117] D. R. Canelhas, "Truncated signed distance fields applied to robotics," Ph.D. dissertation, School of Science and Technology, 2017.

[118] C. Zach, T. Pock, and H. Bischof, "A globally optimal algorithm for robust tv-l 1 range image integration," in *ICCV*. IEEE, 2007, pp. 1–8.

[119] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *ACM siggraph computer graphics*, vol. 21, no. 4, pp. 163–169, 1987.

[120] S. Billings and R. Taylor, "Generalized iterative most likely oriented-point (g-imlop) registration," *IJCARS*, vol. 10, no. 8, pp. 1213–1226, 2015.

[121] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[122] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 4, pp. 349–359, Oct 1999.

[123] F. Cazals and J. Giesen, "Delaunay triangulation based surface reconstruction," in *Effective computational geometry for curves and surfaces*. Springer, 2006, pp. 231–276.

[124] R. Wang, S. M. Pizer, and J.-M. Frahm, "Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5555–5564.

[125] X. Huang, G. Mei, J. Zhang, and R. Abbas, "A comprehensive survey on point cloud registration," *CoRR*, vol. abs/2103.02690, 2021.

[126] A. Kovnatsky, M. M. Bronstein, A. M. Bronstein, K. Glashoff, and R. Kimmel, "Coupled quasi-harmonic bases," in *Computer Graphics Forum*, vol. 32, no. 2pt4. Wiley Online Library, 2013, pp. 439–448.

[127] Q. Huang, F. Wang, and L. Guibas, "Functional map networks for analyzing and exploring large shape collections," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, pp. 1–11, 2014.

[128] Y. Aflalo, A. Dubrovina, and R. Kimmel, "Spectral generalized multi-dimensional scaling," *International Journal of Computer Vision*, vol. 118, no. 3, pp. 380–392, 2016.

[129] D. Eynard, E. Rodola, K. Glashoff, and M. M. Bronstein, "Coupled functional maps," in *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE, 2016, pp. 399–407.

[130] O. Burghard, A. Dieckmann, and R. Klein, "Embedding shapes with green's functions for global shape matching," *Computers & Graphics*, vol. 68, pp. 1–10, 2017.

[131] E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers, "Partial

functional correspondence," in *Computer Graphics Forum*, vol. 36, no. 1. Wiley Online Library, 2017, pp. 222–236.

[132] R. Litman and A. M. Bronstein, "Learning spectral descriptors for deformable shape correspondence," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 1, pp. 171–180, 2013.

[133] T. Windheuser, M. Vestner, E. Rodola, R. Triebel, and D. Cremers, "Optimal intrinsic descriptors for non-rigid shape analysis," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.

[134] D. Boscaini, J. Masci, S. Melzi, M. M. Bronstein, U. Castellani, and P. Vandergheynst, "Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks," in *Computer Graphics Forum*, vol. 34, no. 5. Wiley Online Library, 2015, pp. 13–23.

[135] D. Boscaini, J. Masci, E. Rodolà, M. M. Bronstein, and D. Cremers, "Anisotropic diffusion descriptors," in *Computer Graphics Forum*, vol. 35, no. 2. Wiley Online Library, 2016, pp. 431–441.

[136] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3dmatch: Learning local geometric descriptors from rgb-d reconstructions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1802–1811.

[137] M. Khoury, Q.-Y. Zhou, and V. Koltun, "Learning compact geometric features," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 153–161.

[138] H. Deng, T. Birdal, and S. Ilic, "Ppfnet: Global context aware local features for robust 3d point matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 195–205.

[139] ——, "Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 602–618.

[140] Z. Gojcic, C. Zhou, J. D. Wegner, and A. Wieser, "The perfect match: 3d point cloud matching with smoothed densities," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5545–5554.

[141] X. Bai, Z. Luo, L. Zhou, H. Fu, L. Quan, and C.-L. Tai, "D3feat: Joint learning of dense detection and description of 3d local features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[142] D. Boscaini, J. Masci, E. Rodolà, and M. Bronstein, "Learning shape correspondence with anisotropic convolutional neural networks," in *Advances in neural information processing systems*, 2016, pp. 3189–3197.

[143] O. Litany, T. Remez, E. Rodola, A. Bronstein, and M. Bronstein, "Deep functional maps: Structured prediction for dense shape correspondence," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5659–5667.

[144] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model cnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5115–5124.

[145] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[146] N. Donati, A. Sharma, and M. Ovsjanikov, "Deep geometric functional maps: Robust feature learning for shape correspondence," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8592–8601.

[147] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "3d-coded: 3d correspondences by deep deformation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 230–246.

[148] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the

kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3354–3361.

[149] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015.

[150] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?" in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018, pp. 2483–2493.

[151] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016.

[152] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *CoRR*, vol. abs/1607.08022, 2016.

[153] Y. Wu and K. He, "Group normalization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[154] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[155] A. Ortiz, C. Robinson, D. Morris, O. Fuentes, C. Kiekintveld, M. M. Hassan, and N. Jojic, "Local context normalization: Revisiting local normalization," in *Pro-*

ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.

[156] H. Nam and H.-E. Kim, "Batch-instance normalization for adaptively style-invariant neural networks," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018, pp. 2558–2567.

[157] P. Luo, J. Ren, Z. Peng, R. Zhang, and J. Li, "Differentiable learning-to-normalize via switchable normalization," in *International Conference on Learning Representations*, 2019.

[158] W. Shao, T. Meng, J. Li, R. Zhang, Y. Li, X. Wang, and P. Luo, "Ssn: Learning sparse switchable normalization via sparsestmax," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[159] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 922–928.

[160] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.

[161] X. Roynard, J.-E. Deschaud, and F. Goulette, "Classification of Point Cloud for Road Scene Understanding with Multiscale Voxel Deep Network," in *10th workshop on Planning, Perceptionand Navigation for Intelligent Vehicules PPNIV'2018*, Madrid, Spain, Oct. 2018, preprint.

[162] B. Graham, M. Engelcke, and L. Van Der Maaten, "3d semantic segmentation with submanifold sparse convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9224–9232.

[163] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3075–3084.

[164] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[165] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017, pp. 5099–5108.

[166] M. Atzmon, H. Maron, and Y. Lipman, "Point convolutional neural networks by

extension operators," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, Jul. 2018.

[167] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," in *Advances in neural information processing systems*, 2018, pp. 820–830.

[168] P. Hermosilla, T. Ritschel, P.-P. Vázquez, À. Vinacua, and T. Ropinski, "Monte carlo convolution for learning on non-uniformly sampled point clouds," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 6, pp. 1–12, 2018.

[169] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[170] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on riemannian manifolds," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, December 2015.

[171] J. Huang, H. Zhang, L. Yi, T. Funkhouser, M. Niessner, and L. J. Guibas, "Texturenet: Consistent local parametrizations for learning from high-resolution

signals on meshes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[172] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or, "Meshcnn: a network with an edge," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–12, 2019.

[173] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[174] L. Jiang, H. Zhao, S. Liu, X. Shen, C.-W. Fu, and J. Jia, "Hierarchical point-edge interaction network for point cloud semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[175] G. Li, M. Muller, A. Thabet, and B. Ghanem, "Deepgcns: Can gcns go as deep as cnns?" in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[176] J. Schult, F. Engelmann, T. Kontogianni, and B. Leibe, "Dualconvmesh-net: Joint geodesic and euclidean convolutions on 3d meshes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[177] H. Lei, N. Akhtar, and A. Mian, "Spherical kernel for efficient graph convolution on 3d point clouds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[178] P. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[179] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, May 2001, pp. 145–152.

[180] A. W. Fitzgibbon, "Robust registration of 2d and 3d point sets," *Image and Vision Computing*, vol. 21, no. 13, pp. 1145–1153, 2003, british Machine Vision Computing 2001.

[181] S. D. Billings, E. M. Boctor, and R. H. Taylor, "Iterative Most-Likely Point Registration (IMLP): A Robust Algorithm for Computing Optimal Shape Alignment," *PLoS ONE*, vol. 10, no. 3, p. e0117688, Mar. 2015.

[182] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp." in *Robotics: science and systems*, vol. 2, no. 4. Seattle, WA, 2009, p. 435.

[183] O. Duchenne, F. Bach, I.-S. Kweon, and J. Ponce, "A tensor-based algorithm for

high-order graph matching," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 12, pp. 2383–2395, 2011.

[184] F. Zhou and F. De la Torre, "Factorized graph matching," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 127–134.

[185] L. Livi and A. Rizzi, "The graph matching problem," *Pattern Analysis and Applications*, vol. 16, no. 3, pp. 253–283, 2013.

[186] F. Zhou and F. De la Torre, "Factorized graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 9, pp. 1774–1789, 2016.

[187] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010.

[188] A. Rasoulian, R. Rohling, and P. Abolmaesumi, "Group-wise registration of point sets for statistical shape models," *IEEE Transactions on Medical Imaging*, vol. 31, no. 11, pp. 2025–2034, 2012.

[189] J. Fan, J. Yang, D. Ai, L. Xia, Y. Zhao, X. Gao, and Y. Wang, "Convex hull indexed gaussian mixture model (ch-gmm) for 3d point set registration," *Pattern Recognition*, vol. 59, pp. 126–141, 2016, compositional Models and Structured Learning for Visual Recognition.

[190] G. D. Evangelidis, D. Kounades-Bastian, R. Horaud, and E. Z. Psarakis, "A Generative Model for the Joint Registration of Multiple Point Sets," in *Computer Vision – ECCV 2014*, ser. Lecture Notes in Computer Science, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 109–122.

[191] O. Enqvist, K. Josephson, and F. Kahl, "Optimal correspondences from pairwise constraints," in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 1295–1302.

[192] N. Dym, H. Maron, and Y. Lipman, "Ds++: A flexible, scalable and provably tight relaxation for matching problems," *arXiv preprint*, 2017.

[193] X. Huang, J. Zhang, Q. Wu, L. Fan, and C. Yuan, "A coarse-to-fine algorithm for matching and registration in 3d cross-source point clouds," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 2965–2977, 2017.

[194] H. M. Le, T.-T. Do, T. Hoang, and N.-M. Cheung, "Sdrsac: Semidefinite-based randomized approach for robust point cloud registration without correspondences," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 124–133.

[195] J. P. Iglesias, C. Olsson, and F. Kahl, "Global optimality for point set registra-

tion using semidefinite programming," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[196] H. Yang, J. Shi, and L. Carlone, "Teaser: Fast and certifiable point cloud registration," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 314–333, 2020.

[197] J. Li, Q. Hu, and M. Ai, "Point cloud registration based on one-point ransac and scale-annealing biweight estimation," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–14, 2021.

[198] H. Maron, N. Dym, I. Kezurer, S. Kovalsky, and Y. Lipman, "Point registration via efficient convex relaxation," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–12, 2016.

[199] H. Deng, T. Birdal, and S. Ilic, "3D local features for direct pairwise registration," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2019.

[200] W. Lu, G. Wan, Y. Zhou, X. Fu, P. Yuan, and S. Song, "Deepvcp: An end-to-end deep neural network for point cloud registration," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 12–21.

[201] G. D. Pais, S. Ramalingam, V. M. Govindu, J. C. Nascimento, R. Chellappa, and P. Miraldo, "3dregnet: A deep neural network for 3d point registration,"

in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 7193–7203.

[202] G. Riegler, A. Osman Ulusoy, and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3577–3586.

[203] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2088–2096.

[204] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, "O-cnn: Octree-based convolutional neural networks for 3d shape analysis," *ACM Transactions On Graphics (TOG)*, vol. 36, no. 4, pp. 1–11, 2017.

[205] Z. J. Yew and G. H. Lee, "3dfeat-net: Weakly supervised local 3d features for point cloud registration," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 630–646.

[206] Y. Wang and J. M. Solomon, "Deep closest point: Learning representations for point cloud registration," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3523–3532.

[207] J. Zhou, M. Wang, W. Mao, M. Gong, and X. Liu, "Siamesepointnet: A siamese point network architecture for learning 3d shape descriptor," in *Computer Graphics Forum*, vol. 39, no. 1. Wiley Online Library, 2020, pp. 309–321.

[208] Z. J. Yew and G. H. Lee, "Rpm-net: Robust point matching using learned features," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 824–11 833.

[209] C. Choy, W. Dong, and V. Koltun, "Deep global registration," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2514–2523.

[210] W. Yuan, B. Eckart, K. Kim, V. Jampani, D. Fox, and J. Kautz, "Deepgmr: Learning latent gaussian mixture models for registration," in *European Conference on Computer Vision*. Springer, 2020, pp. 733–750.

[211] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, "Pointnetlk: Robust & efficient point cloud registration using pointnet," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7163–7172.

[212] X. Huang, G. Mei, and J. Zhang, "Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 366–11 374.

[213] H. Zhu, C. Cui, L. Deng, R. C. Cheung, and H. Yan, "Elastic net constraint-based tensor model for high-order graph matching," *IEEE transactions on cybernetics*, 2019.

[214] S. D. Billings, A. Sinha, A. Reiter, S. Leonard, M. Ishii, G. D. Hager, and R. H. Taylor, "Anatomically constrained video-ct registration via the v-imlop algorithm," in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2016*, S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, and W. Wells, Eds. Cham: Springer International Publishing, 2016, pp. 133–141.

[215] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle *et al.*, "The cancer imaging archive (tcia): maintaining and operating a public information repository," *Journal of digital imaging*, vol. 26, no. 6, pp. 1045–1057, 2013.

[216] A. Sinha, A. Reiter, S. Leonard, M. Ishii, G. D. Hager, and R. H. Taylor, "Simultaneous segmentation and correspondence improvement using statistical modes," in *Medical Imaging 2017: Image Processing*, M. A. Styner and E. D. Angelini, Eds., vol. 10133, International Society for Optics and Photonics. SPIE, 2017, pp. 377 – 384.

[217] S. Valette, J. M. Chassery, and R. Prost, "Generic remeshing of 3d triangular

meshes with metric-dependent discrete voronoi diagrams," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 2, pp. 369–381, 2008.

[218] Z. Gojcic, C. Zhou, J. D. Wegner, and A. Wieser, "The perfect match: 3d point cloud matching with smoothed densities," in *Proceedings of the IEEE / CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5545–5554.

[219] S. Ehsan, A. F. Clark, N. U. Rehman, and K. D. McDonald-Maier, "Integral images: Efficient algorithms for their computation and storage in resource-constrained embedded vision systems," *Sensors*, vol. 15, no. 7, pp. 16 804–16 830, 2015.

[220] Z. Liu, H. Hu, Y. Cao, Z. Zhang, and X. Tong, "A closer look at local aggregation operators in point cloud analysis," *ECCV*, 2020.

[221] L. McInnes, J. Healy, N. Saul, and L. Großberger, "Umap: Uniform manifold approximation and projection," *Journal of Open Source Software*, vol. 3, no. 29, p. 861, 2018.

[222] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-Scale direct monocular SLAM," *Lect. Notes Comput. Sci.*, vol. 8690 LNCS, no. PART 2, pp. 834–849, 2014.

[223] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid,

and J. J. Leonard, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” *IEEE Trans. Rob.*, vol. 32, no. 6, pp. 1309–1332, 2016.

[224] K. Tateno, F. Tombari, I. Laina, and N. Navab, “CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 6565–6574, 2017.

[225] R. Mur-Artal and J. D. Tardós, “ORB-SLAM2 an Open-Source SLAM system for monocular stereo.pdf,” *IEEE Trans. Rob.*, vol. 33, no. 5, pp. 1255–1262, 2017.

[226] R. Li, S. Wang, and D. Gu, “Ongoing evolution of visual SLAM from geometry to deep learning: Challenges and opportunities,” *Cognit. Comput.*, vol. 10, no. 6, pp. 875–889, Dec. 2018.

[227] M. Bloesch, J. Czarnowski, R. Clark, S. Leutenegger, and A. J. Davison, “CodeSLAM - learning a compact, optimisable representation for dense visual SLAM,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2560–2568, 2018.

[228] T. Laidlow, J. Czarnowski, and S. Leutenegger, “DeepFusion: Real-time dense 3D reconstruction for monocular SLAM using single-view depth and gradient

predictions," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May, pp. 4068–4074, 2019.

[229] L. Tiwari, P. Ji, Q.-H. Tran, B. Zhuang, S. Anand, and M. Chandraker, "Pseudo rgb-d for self-improving monocular slam and depth prediction," in *European Conference on Computer Vision*, 2020.

[230] W. N. Greene and N. Roy, "Metrically-Scaled monocular SLAM using learned scale factors," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2020, pp. 43–50.

[231] J. Czarnowski, T. Laidlow, R. Clark, and A. J. Davison, "DeepFactors: Real-Time probabilistic dense monocular SLAM," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 721–728, 2020.

[232] C. Wang, M. Oda, Y. Hayashi, T. Kitasaka, H. Honma, H. Takabatake, M. Mori, H. Natori, and K. Mori, "Visual slam for bronchoscope tracking and bronchus reconstruction in bronchoscopic navigation," in *Medical Imaging 2019: Image-Guided Procedures, Robotic Interventions, and Modeling*, vol. 10951. International Society for Optics and Photonics, 2019, p. 109510A.

[233] C. Xie, T. Yao, J. Wang, and Q. Liu, "Endoscope localization and gastrointestinal feature map construction based on monocular SLAM technology," *J. Infect. Public Health*, pp. 4–11, 2019.

[234] R. Ma, R. Wang, Y. Zhang, S. Pizer, S. K. McGill, J. Rosenman, and J.-M. Frahm, "RNNSLAM: Reconstructing the 3D colon to visualize missing regions during a colonoscopy," *Med. Image Anal.*, vol. 72, p. 102100, May 2021.

[235] N. Yang, L. v. Stumberg, R. Wang, and D. Cremers, "D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1281–1292.

[236] C. Tang and P. Tan, "Ba-net: Dense bundle adjustment network," *arXiv preprint*, 2018.

[237] X. Wei, Y. Zhang, Z. Li, Y. Fu, and X. Xue, "Deepsfm: Structure from motion via deep bundle adjustment," in *European conference on computer vision*. Springer, 2020, pp. 230–247.

[238] H. Zhan, C. S. Weerasekera, J. W. Bian, and I. Reid, "Visual odometry revisited: What should be learnt?" *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4203–4210, 2020.

[239] N. Mahmoud, I. Cirauqui, A. Hostettler, C. Doignon, L. Soler, J. Marescaux, and J. M. M. Montiel, "Orbslam-based endoscope tracking and 3d reconstruction," in *Computer-Assisted and Robotic Endoscopy*, T. Peters, G.-Z. Yang, N. Navab,

K. Mori, X. Luo, T. Reichl, and J. McLeod, Eds. Cham: Springer International Publishing, 2017, pp. 72–83.

[240] N. Mahmoud, T. Collins, A. Hostettler, L. Soler, C. Doignon, and J. Montiel, "Live tracking and dense reconstruction for handheld monocular endoscopy," *IEEE Transactions on Medical Imaging*, vol. 38, pp. 79–89, 2019.

[241] M. Turan, E. P. Örnek, N. Ibrahimli, C. Giracoglu, Y. Almalioglu, M. Yanik, and M. Sitti, "Unsupervised odometry and depth learning for endoscopic capsule robots," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1801–1807, 2018.

[242] M. Turan, Y. Almalioglu, H. Araujo, E. Konukoglu, and M. Sitti, "A non-rigid map fusion-based direct SLAM method for endoscopic capsule robots," *Int J Intell Robot Appl*, vol. 1, no. 4, pp. 399–409, Nov. 2017.

[243] J. Song, L. Zhao, S. Huang, and G. Dissanayake, "An observable time series based slam algorithm for deforming environment," *ArXiv*, vol. abs/1906.08563, 2019.

[244] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Lect. Notes Comput. Sci.*, vol. 9351, pp. 234–241, 2015.

[245] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 85–100.

[246] M. Bosse, G. Agamennoni, I. Gilitschenski *et al.*, *Robust estimation and applications in robotics*. Now Publishers, 2016.

[247] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2794–2802.

[248] M. Avi-Aharon, A. Arbelle, and T. R. Raviv, "Deephist: Differentiable joint and color histogram layers for image-to-image translation," 2020.

[249] J. Blanco, "A tutorial on se (3) transformation parameterizations and on-manifold optimization," *University of Malaga, Tech. Rep*, no. 3, 2010.

[250] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "ISAM2: Incremental smoothing and mapping using the bayes tree," *Int. J. Rob. Res.*, vol. 31, no. 2, pp. 216–235, 2012.

[251] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, October 2012.

[252] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 7244–7251.

[253] J. Kopf, X. Rong, and J.-B. Huang, "Robust consistent video depth estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1611–1621.