

Globally Optimal Fetoscopic Mosaicking Based on Pose Graph Optimisation with Affine Constraints

Liang Li, Sophia Bano, Jan Deprest, Anna L David, Danail Stoyanov, and Francisco Vasconcelos

Abstract—Fetoscopic laser ablation surgery could be guided using a high-quality panorama of the operating site, representing a map of the placental vasculature. This can be achieved during the initial inspection phase of the procedure using image mosaicking techniques. Due to the lack of camera calibration in the operating room, it has been mostly modelled as an affine registration problem. While previous work mostly focuses on image feature extraction for visual odometry, the challenges related to large-scale reconstruction (re-localisation, loop closure, drift correction) remain largely unaddressed in this context. This letter proposes using pose graph optimisation to produce globally optimal image mosaics of placental vessels. Our approach follows the SLAM framework with a front-end for visual odometry and a back-end for long-term refinement. Our front-end uses a recent state-of-the-art odometry approach based on vessel segmentation, which is then managed by a key-frame structure and the bag-of-words (BoW) scheme to retrieve loop closures. The back-end, which is our key contribution, models odometry and loop closure constraints as a pose graph with affine warpings between states. This problem in the special Euclidean space cannot be solved by existing pose graph algorithms and available libraries such as G2O. We model states on affine Lie group with local linearisation in its Lie algebra. The cost function is established using Mahalanobis distance with the vectorisation of the Lie algebra. Finally, an iterative optimisation algorithm is adopted to minimise the cost function. The proposed pose graph optimisation is first validated on simulation data with a synthetic trajectory that has different levels of noise and different numbers of loop closures. Then the whole system is validated using real fetoscopic data that has three sequences with different numbers of frames and loop closures. Experimental results validate the advantage of the proposed method compared with baselines.

Index Terms—Surgical robotics, pose graph optimisation, fetoscopic camera, image mosaicking, affine Lie group

I. INTRODUCTION

IMAGE mosaicking or stitching is a classic problem in robotics and computer vision for field-of-view expansion.

Manuscript received: February 24th, 2021; Revised: May 27th, 2021; Accepted: 12th, July, 2021. This paper was recommended for publication by Editor Pietro Valdastri upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the Wellcome/EPSC Centre for Interventional and Surgical Sciences (WEISS) at UCL (203145Z/16/Z), EPSC (EP/P027938/1, EP/R004080/1, NS/A000027/1), the H2020 FET (GA863146) and Wellcome [WT101957]. Danail Stoyanov is supported by a Royal Academy of Engineering Chair in Emerging Technologies (CiET1819/2/36) and an EPSC Early Career Research Fellowship (EP/P012841/1).

L. Li, S. Bano, D. Stoyanov and F. Vasconcelos are with the Wellcome/EPSC Centre for Interventional and Surgical Sciences (WEISS) and Department of Computer Science, University College London, London, UK.

J. Deprest is with WEISS, and Department of Development and Regeneration, University Hospital KU Leuven, Leuven, Belgium., and the Institute of Women's Health, UCL, London, UK

AL. David is with WEISS, and EGA Institute for Women's Health, University College London, and National Institute for Health Research University College London Hospitals Biomedical Research Centre.

This has relevant applications in image guided minimally invasive surgery [1], where endoscopic cameras typically have a very narrow field-of-view when compared to the entire operating site, and therefore image mosaicking can offer a global map of the targeted anatomy to guide the surgeon. This is the case of fetoscopic laser ablation surgery for treating twin-to-twin transfusion syndrome (TTTS) [2]. It is a minimally invasive procedure for complicated identical twin pregnancies with a single shared placenta and is guided by a fetoscopic camera and ultrasound. The objective of the procedure is to stop unbalanced blood transfusion from one twin to another, which is achieved by photocoagulating anastomoses on the placenta surface with a laser fiber exiting the fetoscope [3]. In this context, image mosaicking would provide a panorama of the placental surface where a map of the entire vasculature network and the targeted anastomoses are visible. This can be achieved during the initial phase of the procedure, where the operating site is being inspected by the surgeon with the fetoscopic camera. Additionally, the image mapping and localisation techniques that mosaicking relies on can potentially drive the autonomous navigation of robotic fetoscopic instruments currently under development [4]–[6].

The main problem is to estimate the pose of every image with respect to a single reference frame. Mosaicking methods can be based on direct (photometric) [7] or indirect (feature-based) [8]–[11] methods. While feature-based methods are limited by the scene environment and require distinctive landmarks, direct methods are computationally heavier to run. Considering the two factors, semi-direct algorithms have also been proposed [12], which combine the advantages of both.

There are several mosaicking challenges that are particular to fetoscopic imaging such as fluid immersion, floating particles, moving light source and reflections, close-range and often out of focus frames. Therefore, the ongoing development of mosaicking techniques in this domain diverges from the state-of-the-art in computer vision literature. Most of the research in this domain has concentrated on addressing the short-term challenges in sequential video frame registration and therefore most of the existing methods rely on visual odometry that eventually accumulates drift error. The clinical need for reliable and consistent mosaics of the entire operating site has therefore not been met with the existing approaches.

This letter aims at addressing the long-term challenges of loop closure and global map consistency in longer fetoscopic sequences. We propose a graph-based globally optimal image mosaicking algorithm and emphasise its usage in fetoscopy. The overall framework is inspired by SLAM, containing a visual odometry front-end and a global optimisation back-end.

The front-end relies on the approach from Bano et. al. [13], that performs vessel segmentation followed by registration of vessel probability maps. A keyframe library is then established based on sparse sampling of visual odometry frames. Loop closures are detected with a Bag-of-Words (BoW) scheme followed by direct image registration. To globally refine odometry and loop closures, the back-end follows closely the G2O framework [14]. The problem is formulated as pose graph optimisation, with the pose of every image as vertex, constraints of odometry and loop closure as edges. Following previous literature on fetoscopy mosaicking, the state of each vertex is a 6 DoF affine matrix. This is different from available pose-graph approaches that work in $SE(2)$ or $SIM(2)$. Instead, we propose to construct the cost function and derive the solution in affine Lie group, *i.e.*, a combination of $GL(2)$ and 2-DOF translation. The linearisation of the state is its Lie algebra, with which the Jacobian and operation of plus can be conducted when updating the state during optimisation. The computation between Lie group and Lie algebra is achieved by exponential and logarithm maps. Diagram of the whole system is summarized in Fig. 1. The contributions of this letter can be summarised as:

1. We propose a SLAM-based fetoscopic mosaicking framework with the full pipeline from the front-end to the back-end where we use pose graph consistency instead of photometric error to optimise the map.
2. We propose an explicit form of Lie group-based pose graph optimisation algorithm on 2D affine space. This not only tackles the fetoscopic mosaicking problem, but is also generalisable to other problems in computer vision and robotics¹.
3. We test our complete mosaicking algorithm on real fetoscopic sequences, showing its advantage compared to the baseline [13] without back-end optimisation. Additionally, the affine graph-optimisation formulation is further validated with simulation data.

The rest of this letter is organised as follows. Section II discusses the related work in fetoscopic mosaicking and pose graph optimisation. Section III gives a detailed description of the proposed method. Section IV presents the results on simulation and real-world data. Finally, Section V concludes this letter.

II. RELATED WORK

Fetoscopic mosaicking aims at obtaining a panorama of the fetal vasculature on the chorionic surface of the placenta from an intraoperative video sequence. The early methods are based on classic point feature detection and matching such as SURF [15] and SIFT [16] and have been tested on synthetic data. However, these methods have low repeatability in more realistic experiments, and explicit detection of vessel features is suggested as a better option [17]. None of the above methods has been shown to work reliably with in-vivo fetoscopic data. In this setting, most breakthroughs have been achieved with direct methods. This includes pixel-wise gradient alignment

[18], a deep learning approach for direct homography regression [19], [20], and more recently, registration of segmented placental vessels [13]. While this last method shows significant progress, it ignores accumulative drift errors that would eventually occur in long videos. Only a small portion of the above methods considers long-term global refinement. A bundle-adjustment approach was proposed in [15], however, it relies on SURF features, which are not reliable with in-vivo data. A direct bundle-adjustment approach was proposed in [18], however, both the retrieval of similar frames and the global optimisation over an entire video are computationally heavy and thus face challenges in an online setting. To the best of our knowledge, a more efficient global refinement solution based on keyframe management and graph pose optimisation, such as Graph SLAM [21], has never been attempted for fetoscopic mosaicking, despite its widespread usage by the robotics community. This would allow a lightweight global optimisation of large mosaics that is image free and does not require the modelling of landmarks.

Pose graph optimisation methods in SLAM can be classified into two categories: filtering-based and error function-based. Herein, we only discuss the error function-based method which is the most relevant to this letter. G2O [14] is one representative method in this category. It minimises Mahalanobis distance of all the constraints provided by odometry and loop closures. Gauss-Newton or Levenberg-Marquardt (LM) algorithms are used with an initial guess linearised by its first-order Taylor expansion. In [22], the authors simplified the computation by solving the rotation part first, then rectified the translation part in 2D case. Robustness to outliers of loop closures was investigated in [23], [24]. The convex relaxation of the optimisation was addressed in [25]. A smoother error function is exploited for the pose graph optimisation in [26]. The asynchronous property of distributed pose graph optimisation was investigated in [27] for multi-robot systems. All these methods work in $SE(2)$, $SE(3)$ and $SIM(3)$, which are the common kinematics spaces in robotics. Uncalibrated camera projections are commonly modelled as a projective transformation $P(3)$. In in-vivo fetoscopy however, due to uncalibrated distortions from the lens and water refraction [28], approximating it with an affine approximation renders iterative estimation more stable as indicated in [13], [18]. Therefore we aimed to derive an algorithm that works on 2D affine space to optimise the mosaicking directly without camera calibration. In this letter, we combine fetoscopic mosaicking and pose graph optimisation to make an efficient and optimal panorama generation algorithm.

III. METHODOLOGY

This section details the full pipeline of the proposed method from the front-end to the back-end.

A. The front-end odometry

Odometry is estimated with the method presented in [13]. We briefly introduce it in this section to make this letter self-contained. It consists of two parts: feature extraction and odometry estimation. The extracted features are segmented

¹The code and data are publicly available on github with the following link <https://github.com/LiangLiUCL/Affine-Pose-Graph-Optimisation.git>

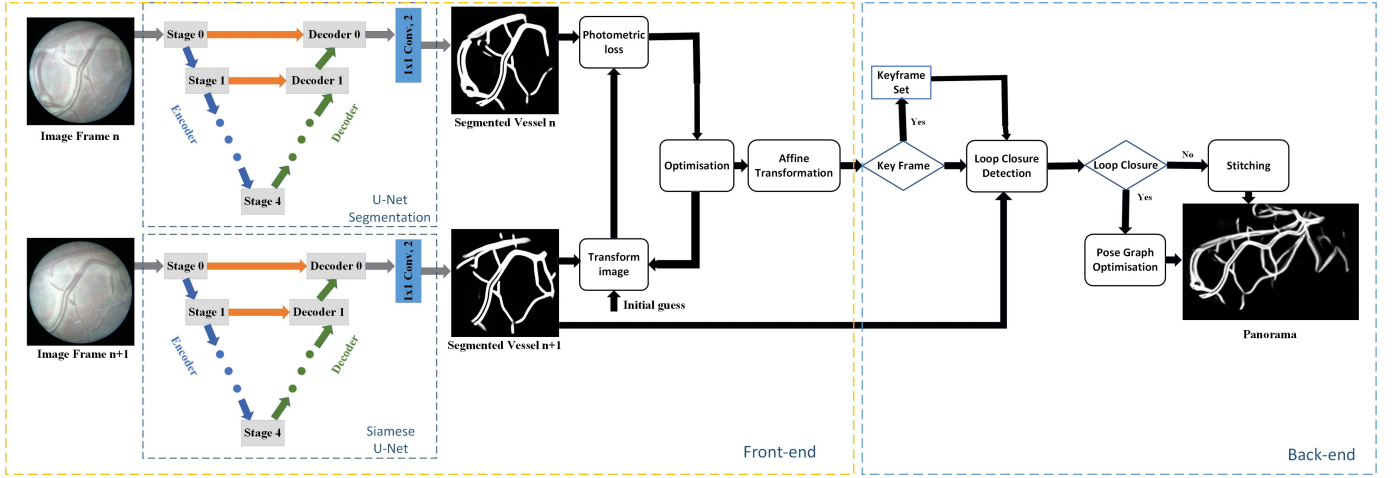


Fig. 1. Diagram of the proposed fetoscopic mosaicking algorithm. It has two parts: front-end and back-end. The main contribution of this work is to develop a back-end to make the obtained panorama globally optimal given all the odometry and loop closure measurements.

vessels on the chorionic placenta surface, obtained using U-Net [29] which has achieved good performance in many biomedical image segmentation tasks. One favourable property of U-Net is that it generalises well in cases where the training dataset is relatively small [29]. The output of U-Net is a probability map of each pixel being part of a vessel.

The odometry estimation is based on the vessel probability map of consecutive frames as that in [13]. It is assumed that the transformation between two frames is affine rather than projective as suggested in [18], as in our case the camera calibration parameters are not available. The image registration is based on the pyramidal Lucas-Kanade framework [30], where the least-squares error between the target image and the warped source image is minimised. There is no analytic solution to this problem, and therefore an iterative optimisation method is adopted. Here, the LM algorithm is used as it controls the convergence better than Gauss-Newton by using a damping factor. The diagram of the front-end can be seen on the left side of Fig. 1.

B. Loop closure detection

In order to deal with large-scale trajectories, we select a subset of all odometry frames as keyframes, and only these will be considered for subsequent optimisation. The first frame is always a keyframe and afterwards keyframes are defined when the movement (translation) estimated by the odometry is beyond a threshold. For every new vessel segmentation image, we determine if there is a loop closure by comparing it against previous keyframes. The keyframes are constructed as clusters using BoW with SURF features, from which a similarity score between frames can be computed². Then the score between the current frame and every keyframe from previous upto 50 frames is computed. As usually the possibility of loop closure

²Please note that we mentioned before that fetoscopic mosaicking based on local point feature matching methods such as SURF is not reliable, however, here we are just using them as a lightweight approach to detect globally similar frames and discarding their inaccurate location within the image for any motion estimation purposes

within 50 frames is small. It may be just caused by slow motion of the camera. Moreover, short loop closures contribute little for the global optimisation. If the score is larger than a threshold (*e.g.*, 0.1), this keyframe will be selected as a loop closure candidate. Finally, the transformation between the current frame and the loop closure candidate is computed using the odometry method described in section III-A. When the obtained transformation is smaller than a threshold, the loop closure will be added. This is because when a real loop closure is detected, the transformation should not be too large, *i.e.*, the current frame and the keyframe should be close to each other.

C. The back-end optimisation

Both odometry estimation between consecutive frames and loop closures between non-adjacent frames should be taken into consideration to generate a consistent map. This problem is modelled as a directed graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, where \mathbf{V} is the set of vertices (pose of the image) and \mathbf{E} is the set of edges (constraints of odometry and loop closures). Every edge is associated with its uncertainty, *i.e.*, the covariance matrix Σ . We do not take the landmarks (vessels) into consideration, and therefore the back-end optimisation does not need to use any image appearance information. The flowchart of the back-end optimisation can be seen on the right side of Fig. 1. An illustration of the pose graph is shown in Fig. 2.

As mentioned in Section III-A, the vertex of the pose graph is represented by a 2D affine matrix. We denote the i -th vertex as:

$$\mathbf{x}_i = \begin{bmatrix} \mathbf{A}_i & \mathbf{t}_i \\ \mathbf{0} & 1 \end{bmatrix} \quad (1)$$

where $\mathbf{A}_i \in GL(2)$ is a 2×2 non-singular matrix on the general linear group, *i.e.*, $|\mathbf{A}_i| \neq 0$, and \mathbf{t}_i is a 2×1 vector denotes translation. The affine matrix group is a Lie group as it is a smooth manifold whose elements satisfy the group axioms with closure under matrix multiplication, identity matrix as the identity, matrix inverse and associativity. We define the plus operation \oplus on the Lie group as left matrix multiplication, *i.e.*,

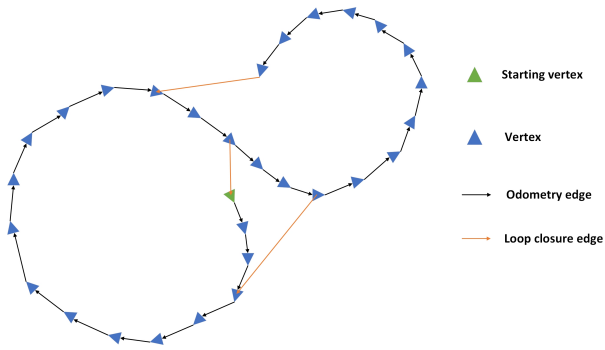


Fig. 2. An illustration of the pose graph. In our case, the vertex is the pose of the image, the odometry is estimated by image registration between consecutive frames, and the loop closure is detected based on similarity score and transformation.

$\mathbf{x}_i \oplus \mathbf{x}_j = \mathbf{x}_j \mathbf{x}_i$, while the minus operation \ominus is defined as $\mathbf{x}_i \ominus \mathbf{x}_j = \mathbf{x}_j^{-1} \mathbf{x}_i$. Odometry edges between the i -th node and the $i+1$ -th node \mathbf{z}_i and loop closure edges between the i -th node and the j -th node \mathbf{z}_{ij} , $|i - j| \neq 1$ are all on the affine Lie group. Given a graph with n vertices, assuming there is an edge between the i -th node and the j -th node, $0 < i, j \leq n$, $i \neq j$, the error constraint by this edge is:

$$\mathbf{e}_{ij} = \mathbf{e}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij}) = (\mathbf{x}_j \ominus \mathbf{x}_i) \ominus \mathbf{z}_{ij} = \mathbf{z}_{ij}^{-1} \mathbf{x}_i^{-1} \mathbf{x}_j \quad (2)$$

where the error \mathbf{e}_{ij} is still a 3×3 matrix on the affine Lie group. The affine matrix has 6-DOF, we need to define the vectorisation of the matrix to construct a cost function. The Lie algebra is a local linearisation of the Lie group, which is a suitable tool in this case. The Lie algebra generators of the 2D affine group are:

$$\begin{aligned} \mathbf{g}_1 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{g}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{g}_3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\ \mathbf{g}_4 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{g}_5 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{g}_6 = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned} \quad (3)$$

The Lie algebra is a weighted sum of the generators:

$$\mathbf{g} = [\omega]_{\times} = \sum_{i=1}^6 \omega_i \mathbf{g}_i, \quad \omega = [\omega_1, \omega_2, \dots, \omega_6]^T \quad (4)$$

The element \mathbf{T} on the affine Lie group $AF(2)$ and its Lie algebra ω^\wedge on $\mathfrak{aff}(2)$ is converted through exponential map and logarithm map:

$$\begin{aligned} \mathbf{T} &= \exp(\omega^\wedge) = \sum_{n=0}^{\infty} \frac{1}{n!} (\omega^\wedge)^n \\ \omega^\wedge &= \log(\mathbf{T}) = \sum_{n=1}^{\infty} (-1)^{n+1} \frac{(\mathbf{T} - \mathbf{I})^n}{n} \end{aligned} \quad (5)$$

where the operator \wedge takes an element on \mathbb{R}^6 and transforms it to an element of $\mathfrak{aff}(2)$ shown in equation (4). In the remainder of this letter we also define the operator \vee as the inverse of \wedge . Thus, the affine Lie group and the vector space are related.

The computation of both maps are by the definition in equation (5) with appropriate orders. Since the transformation between consecutive frames or the loop closures is not large, *i.e.*, the image does not change dramatically in the next time step or when a loop closure is detected, we can define the new error function in the vector space as:

$$\tilde{\mathbf{e}}_{ij}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij}) = \log(\mathbf{e}_{ij})^\vee = \log(\mathbf{z}_{ij}^{-1} \mathbf{x}_i^{-1} \mathbf{x}_j)^\vee \quad (6)$$

Then the cost function can be defined as:

$$\mathbf{f}(\mathbf{x}) = \sum_{i,j \in \mathcal{C}} \mathbf{f}_{ij} = \sum_{i,j \in \mathcal{C}} \tilde{\mathbf{e}}_{ij}^\top \boldsymbol{\Omega}_{ij} \tilde{\mathbf{e}}_{ij} \quad (7)$$

where the 6×6 matrix $\boldsymbol{\Omega}_{ij}$ is the information matrix, *i.e.*, the inverse of the covariance, \mathcal{C} is the set of edges, $\mathbf{x} = [\mathbf{x}_1^\top, \mathbf{x}_2^\top, \dots, \mathbf{x}_n^\top]^\top$ is the state to be estimated. As the Lie algebra at different point of the manifold is different, it can not be updated directly in the vector space. Instead, it should be updated iteratively by adding a perturbation around the current value in every step. In this way, it can be approximated as a least squares problem in every iterative step. For the error function in equation (2), a small perturbation on the Lie algebra and taking the first-order Taylor approximation will result in

$$\begin{aligned} \tilde{\mathbf{e}}_{ij}(\mathbf{x} \oplus \exp(\delta \boldsymbol{\xi}^\wedge)) &= \mathbf{z}_{ij}^{-1} \mathbf{x}_i^{-1} \exp(\boldsymbol{\xi}_i^\wedge)^{-1} \exp(\boldsymbol{\xi}_j^\wedge) \mathbf{x}_j \\ &\simeq \tilde{\mathbf{e}}_{ij}(\mathbf{x}) + \mathbf{J}_{ij} \delta \boldsymbol{\xi} \end{aligned} \quad (8)$$

where \mathbf{J}_{ij} is the Jacobian of $\tilde{\mathbf{e}}_{ij}$ with respect to $\delta \boldsymbol{\xi}$. It can be derived to obtain an approximation form with the Lie group theory and the adjoint operation. However, it does not have the elegant form as that in $SE(3)$ or other Euclidean, orthogonal spaces. In this letter, we do not derive it but use a numerical method to compute it:

$$\begin{aligned} \mathbf{J}_{ij} &= \left. \frac{\partial \tilde{\mathbf{e}}_{ij}(\mathbf{x} \oplus \exp(\delta \boldsymbol{\xi}^\wedge))}{\partial \delta \boldsymbol{\xi}} \right|_{\delta \boldsymbol{\xi} = \mathbf{0}} \\ &\simeq \left. \frac{\tilde{\mathbf{e}}_{ij}(\mathbf{x} \oplus \exp(\delta \boldsymbol{\xi}^\wedge)) - \tilde{\mathbf{e}}(\mathbf{x})}{\delta \boldsymbol{\xi}} \right|_{\delta \boldsymbol{\xi} \rightarrow \mathbf{0}} \end{aligned} \quad (9)$$

At every iterative step, we add a very small perturbation to the current estimated state to compute $\mathbf{J}_{ij} \in \mathbb{R}^{6 \times 6}$. Then following the framework of G2O [14], we can have:

$$\begin{aligned} \tilde{\mathbf{f}}_{ij}(\mathbf{x} \oplus \exp(\delta \boldsymbol{\xi}^\wedge)) &= \tilde{\mathbf{e}}_{ij}(\mathbf{x} \oplus \exp(\delta \boldsymbol{\xi}^\wedge))^\top \boldsymbol{\Omega}_{ij} \tilde{\mathbf{e}}_{ij}(\mathbf{x} \oplus \exp(\delta \boldsymbol{\xi}^\wedge)) \\ &\simeq (\tilde{\mathbf{e}}_{ij}(\mathbf{x}) + \mathbf{J}_{ij} \delta \boldsymbol{\xi})^\top \boldsymbol{\Omega}_{ij} (\tilde{\mathbf{e}}_{ij}(\mathbf{x}) + \mathbf{J}_{ij} \delta \boldsymbol{\xi}) \\ &= \underbrace{\tilde{\mathbf{e}}_{ij}(\mathbf{x})^\top \boldsymbol{\Omega}_{ij} \tilde{\mathbf{e}}_{ij}(\mathbf{x})}_{c_{ij}} + 2 \underbrace{\tilde{\mathbf{e}}_{ij}(\mathbf{x})^\top \boldsymbol{\Omega}_{ij} \mathbf{J}_{ij}}_{\mathbf{b}_{ij}} \delta \boldsymbol{\xi} \\ &\quad + \delta \boldsymbol{\xi}^\top \underbrace{\mathbf{J}_{ij}^\top \boldsymbol{\Omega}_{ij} \mathbf{J}_{ij}}_{\mathbf{H}_{ij}} \delta \boldsymbol{\xi} \end{aligned} \quad (10)$$

By adding all the constraints in equation (7), we can get the perturbed cost function as:

$$\begin{aligned} \tilde{\mathbf{f}}(\mathbf{x} \oplus \exp(\delta \boldsymbol{\xi}^\wedge)) &= \sum_{i,j \in \mathcal{C}} \tilde{\mathbf{f}}_{ij} \\ &\simeq \sum_{i,j \in \mathcal{C}} (c_{ij} + 2\mathbf{b}_{ij} \delta \boldsymbol{\xi} + \delta \boldsymbol{\xi}^\top \mathbf{H}_{ij} \delta \boldsymbol{\xi}) \\ &= c + 2\mathbf{b}^\top \delta \boldsymbol{\xi} + \delta \boldsymbol{\xi}^\top \mathbf{H} \delta \boldsymbol{\xi} \end{aligned} \quad (11)$$

where $c = \sum c_{ij}$ is the sum of c_{ij} presented in equation (10), $\mathbf{b} = \sum \mathbf{b}_{ij}^\top$ is the sum of \mathbf{b}_{ij} presented in equation (10), and $\mathbf{H} = \sum \mathbf{H}_{ij}$ is the Hessian matrix. Equation (11) is a linear function and has an analytic solution. In this letter, we use the LM algorithm to solve the equation:

$$(\mathbf{H} + \lambda \mathbf{I})\delta\xi^* = -\mathbf{b} \quad (12)$$

where λ is called the damping factor that is controlled dynamically to improve the convergence. The updated state at this step is:

$$\mathbf{x}^* = \mathbf{x} \oplus \exp(\delta\xi^\wedge) \quad (13)$$

The algorithm repeats equation (6) at the updated point to compute $\hat{\mathbf{e}}_{ij}$, equation (9) at the updated point to compute \mathbf{J}_{ij} , equation (10) and (11) to compute \mathbf{b} , \mathbf{H} , equation (12) and (13) to update the state until the convergence criteria is fulfilled. The back-end algorithm is based on the framework of G2O [14], we define our own affine Lie group class, the operation, exponential map, logarithm map, *etc.*, the new vertex type and new edge type to make it work on the 2D affine pose graph optimisation.

IV. EXPERIMENTS

This section presents the results both on simulation data and real in-vivo fetoscopic sequences.

A. Simulation

In order to verify the performance of the proposed pose graph optimisation and give quantitative results, we create simulated data on $SIM(2)$, *i.e.*, with rotation, translation and scale, which is a special case of the 2D affine motion. The state can be represented by an affine matrix:

$$\mathbf{x}_i = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}$$

where $s \in \mathbb{R}^+$ is the scale, \mathbf{R} is a 2D rotation matrix, and $\mathbf{t} \in \mathbb{R}^{2 \times 1}$ is the translation vector. The robot/camera starts at $(-100, 0)$ point, rotates with respect to the origin at a constant angular velocity. Meanwhile, the scale s decreases gradually from 1. This makes the ground truth trajectory a spiral curve. It rotates five laps and there are fifty nodes in every lap, which means the graph has 250 vertices in total. The ground truth of the pose is 4-DOF, *i.e.*, one scale, one rotation and two translation parameters. Nevertheless, we add noise in all six dimensions of the affine group, *i.e.*, the coefficient on Lie algebra from ω_1 to ω_6 of the noise in equation (4) are all non-zero. We test the proposed graph optimisation algorithm both with dense and sparse loop closures. In the dense case, for the first three laps, we add loop closures between the i -th vertex to the $i + 49$ -th, $i + 50$ -th, and $i + 51$ -th vertex, where $0 < i \leq 150$. For the last two laps, we add loop closures between the i -th vertex to the $i + 49$ -th and $i + 50$ -th vertex, where $150 < i \leq 200$. So, there are 550 loop closures in total. Then we decrease the number of loop closures gradually to 350, 91, and until the very sparse case 21. Noise for the loop closure is the same as that in the odometry edges. We test the algorithm with different levels of noise to validate

TABLE I
A COMPARISON OF THE AVERAGE POSITION ERROR BEFORE AND AFTER OPTIMISATION WITH DIFFERENT LEVELS OF NOISE AND DIFFERENT NUMBERS OF LOOP CLOSURES

LC	Noise	Case 1		Case 2		Case 3	
		GLN 0.005	TN 0.01	GLN 0.008	TN 0.02	GLN 0.01	TN 0.03
	Bef. Opt.	17.99		28.78		36.05	
550	Aft. Opt.	1.18		1.51		1.54	
350	Aft. Opt.	0.78		1.13		1.33	
91	Aft. Opt.	1.71		2.60		3.01	
21	Aft. Opt.	1.97		3.15		3.92	

GLN: General linear noise TN: Translation noise LC: Loop closure

its robustness. We first add a small amount of noise to the ground truth (0.01 for translation, *i.e.*, \mathbf{g}_3 and \mathbf{g}_4 , and 0.005 for the general linear part, *i.e.*, \mathbf{g}_1 , \mathbf{g}_2 , \mathbf{g}_5 and \mathbf{g}_6 for every edge), which can simulate the condition where the odometry estimation is accurate. The result of the dense case is shown in Fig. 3 (a), from which we can see that after optimisation the trajectory is much closer to the ground truth. For the second case, the added noise is larger than that in the first case (0.02 for translation and 0.008 for the general linear component). The result is presented in Fig. 3 (b), from which we can see that the proposed pose graph optimisation is still able to correct the pose and trajectory given the loop closures even though the trajectory estimated by odometry has a larger drift. Finally, for the third simulation, we increase the noise further (0.03 for translation, and 0.01 for the general linear part), and the result is shown in Fig. 3 (c). We can see remarkable improvement after optimisation as that in the first two cases though the trajectory before optimisation is much different compared to the ground truth. The quantitative results of all the sparse and dense cases in terms of loop closures are presented in Table I, where we show the average position error, *i.e.*, the sum of the position error for every vertex divided by the total number of vertices. It shows that after optimisation the error becomes much smaller in all these cases. When the number of loop closure decreases to 350, the errors after optimisation are even smaller than those that have 550 loop closures. This is due to the reason that when the loop closures are too dense, the error of all the loop closures cannot be ignored. When it becomes a little sparse, the error brought by the loop closure becomes smaller. Thus, the optimisation result becomes better. But when the density of loop closures decreases significantly to 91 or 21, the constraints are not as good as the first two cases. So, the error after optimisation increases a little bit. This simulation demonstrates effectiveness and convergence of the proposed pose graph optimisation.

B. Fetoscopic mosaicking results

We also test the proposed method using real in-vivo fetoscopic data. We use an extended version of the publicly available data from [13]. While the original dataset includes short sequences from 6 different surgeries, valid loop closures were detected in 3 of them (sequences 1, 2, 3 in [13]), and therefore the results in this section will focus on these. We

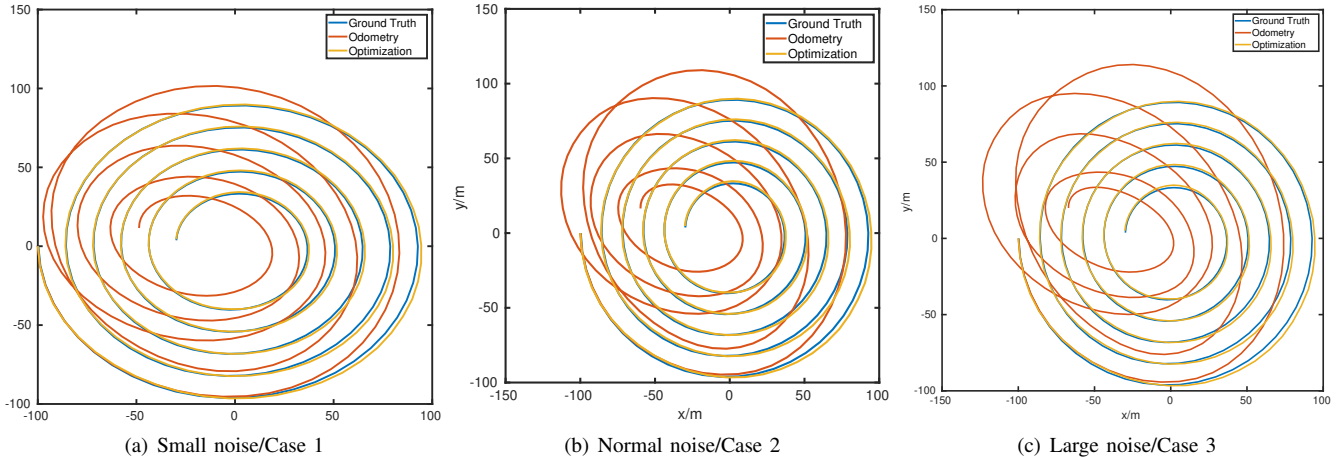


Fig. 3. Results on the simulation data with different levels of noise in the front-end. In (a), the noise of general linear part is 0.005, and the noise of the translation is 0.01 for every odometry edge. We can see that the trajectory after optimisation is very close to the ground truth. In (b), the noise increases with 0.008 for the general linear part and 0.02 for translation. In (c), the noise for the general linear part and translation are 0.01 and 0.03 respectively. Though the odometry is farther from the ground truth in the last two cases, the proposed method can still bend the trajectories back.

TABLE II
NUMBER OF FRAMES AND LOOP CLOSURES OF THE THREE FETOSCOPIC SEQUENCES

Sequence	Sequence 1	Sequence 2	Sequence 3
Number of frames	400	300	150
Number of loop closures	19	13	1

also extend sequence 2 and 3 with 100 additional frames to better visualise the effect of incremental drift. The 100 frames are added to the original 'end frame' of that in [13], where it is the point when vessel-based mosaicking fails. The number of frames and detected loop closures are presented in Table II. The uncertainties of the odometry edges as well as the loop closures are based on their similarity scores. Sequence 1 and 2 both have more than 10 loop closures detected, and the loops are entangled. While sequence 3 only has one loop closure detected. As results in [13] have already demonstrated that vessel-based mosaicking outperforms equivalent approaches based on RGB data, herein we focus on comparing vessel mosaics before and after pose graph optimisation. We also compare the proposed method with the deep sequential mosaicking (DSM) presented in [19], which is an end-to-end method. Since we do not have the ground-truth motion of these sequences, we only provide a qualitative analysis.

The result of sequence 1 is shown in Fig. 4 (a) - (d). Here we present the vessel panorama and the trajectory before and after optimisation. From Fig. 4 (a), we can see that the odometry estimated by image registration is already reliable with no noticeable drift, which is in accordance with the analysis in [13]. Though the improvement after back-end optimisation is marginal (Fig. 4 (b)), we can still notice that the result is slightly refined. Especially when comparing the parts in the dash cyan circle in Fig. 4 (a) and Fig. 4 (b), we can see that small vessel discontinuities and breaks are eliminated after optimisation. Fig. 4 (d) shows the estimated trajectories before

and after optimisation. They are the same until a loop closure is detected. Most of the loop closures are around $(-500, 150)$ where the fetoscope moves very little over 100 frames. The optimisation corrects the small drift during this time and the difference is observable afterwards. From the result of this sequence, we observe that if the front-end is accurate, the back-end optimisation does not destroy the mosaicking. In contrast, it can improve the result marginally. While Fig. 4 (c) shows that the DSM-based mosaicking has lots of drift even compared to the vessel-based mosaicking before optimisation.

The result of sequence 2 is shown in Fig. 4 (e) - (h). The improvement after optimisation is more obvious in this case, as the odometry is not as good as in sequence 1. The segmentation result has some occlusions and false negative segmentation areas. The cyan circle in Fig. 4 (e) and Fig. 4 (f) shows that the vessel after optimisation becomes connected and more consistent. In Fig. 5, the fetoscopic images stitched only by the front-end [13] and optimised by the proposed method are presented. It demonstrates that the proposed back-end optimisation improves the mosaicking quality substantially when there are odometry errors due to occasional vessel segmentation failures. From Fig. 4 (h), we can see that the trajectories before and after optimisation are much different, which means the correction by the back-end is large. The result of sequence 3 is shown in Fig. 4 (i) - (l). The most significant difference is in the left side of the stitched panorama. From Fig. 4 (i), we can see a noticeable drift in the front-end estimations that makes the same vessel appear twice in different parts of the mosaic. The back-end optimisation corrects this error and improves the alignment. Fig. 4 (l) also indicates that the trajectory after optimisation noticeably improves the mosaics compared with that estimated by the front-end. Similar to sequence 1, the vessel-based method outperforms the DSM-based mosaicking of the last two sequences significantly no matter before or after optimisation. If there are heavy occlusions, some frames and estimation of the front-end segmentation may have large

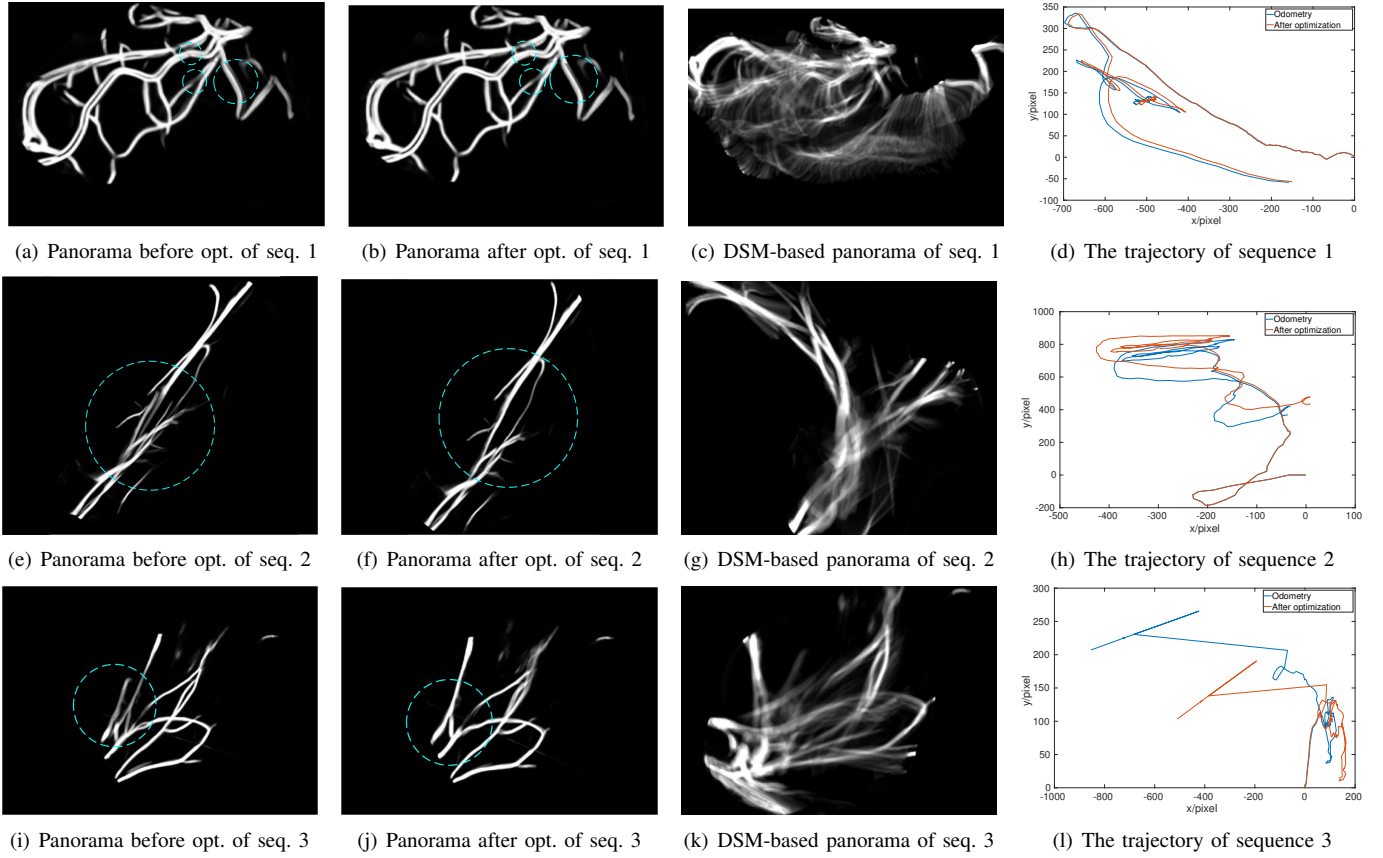


Fig. 4. The result of fetoscopic mosaicking, with the first row of sequence 1, the second row of sequence 2, and the third row of sequence 3. The vessel panoramas before and after back-end optimisation are shown in first and second columns respectively. The vessel panoramas based on DSM are shown in the third column. And the fourth column shows the trajectories before and after optimisation. We can see that the panorama after optimisation is less blurry or become connected in the cyan dash circle. And the vessel-based mosaicking is much better than the end-to-end method.

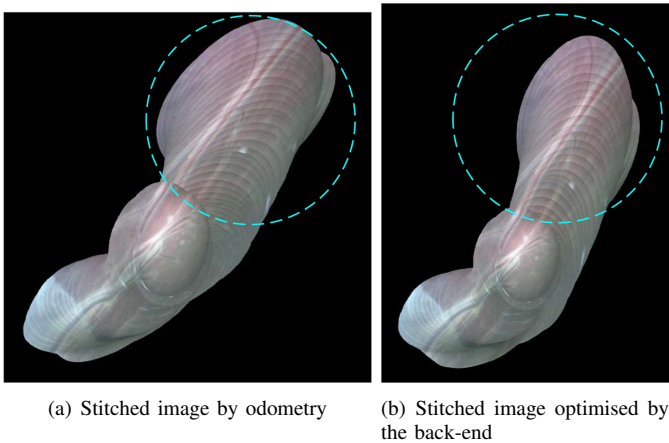


Fig. 5. The stitched image of sequence 2 using vessel only by the front-end (a), and by the proposed method (b). From the cyan dash circle, it can be seen that the shape after optimisation is much different.

errors or even fail. In this case, the back-end has two possible performances. First, it can drive the trajectory/mosaicking to a better result but still has noticeable errors. Or second, it may converge to the local minima because of the poor initial guess provided by the front-end. In future work, we plan to investigate how to identify outliers in both odometry

and loop closure. As that of SLAM for mobile robots, the proposed method works if there are loop closures, and the error bound (noise) of the front-end is within a reasonable range. For example, for sequence 4 to 6 of [13], there is no loop closure detected, then it does not make sense to optimise the mosaicking using the proposed back-end.

V. CONCLUSIONS

This letter presents a globally optimal fetoscopic mosaicking algorithm, based on the full pipeline of SLAM from the front-end to the back-end. From the theoretical perspective, we generalise pose graph optimisation to affine transformations between edges and implement a 2D affine extension to the G2O pose graph framework. While we focus on fetoscopic mosaicking, this self-contained extension can potentially find application in any other robotics or computer vision domains where affine models are relevant. From the practical perspective, the presented results extend the recent state-of-the-art in fetoscopic mosaicking by combining vessel segmentation with global consistency optimisation. Further work is necessary to extend the current approach to map the entire chorionic placental surface and finally tackle the targeted clinical application. This includes a re-localisation module for circumstances when vessel segmentation completely fails due

to either significant and prolonged vascular occlusions or vessels simply not being present in the camera view. Rectifying the significant lens radial distortion of the fetoscopic camera would remove mosaic inconsistencies that are impossible to resolve with affine warpings. While traditional calibration methods are unfeasible in the operating room, recent advances in self-calibration should be investigated [31]. Additionally, the visual appearance of anastomoses will change during the procedure due to laser photocoagulation. Dynamically updating the corresponding mapping areas in real-time is still an open and interesting problem that requires investigation. We also expect that experimentation with motion tracked cameras can provide more insight into validating this approach as well as enabling potential data-fusion approaches to mosaicking.

REFERENCES

- [1] T. Bergen and T. Wittenberg, "Stitching and surface reconstruction from endoscopic image sequences: a review of applications and methods," *IEEE J. Biomed. Heal. Informatics*, vol. 20, no. 1, pp. 304–321, 2014.
- [2] A. Baschat, R. H. Chmait, J. Deprest, E. Gratacós, K. Hecher, E. Kontopoulos, R. Quintero, D. W. Skupski, D. V. Valsky, and Y. Ville, "Twin-to-twin transfusion syndrome (tts)," *J. Perinat. Medicine*, vol. 39, no. 2, pp. 107–112, 2011.
- [3] J. Deprest, D. Van Schoubroeck, P. Van Ballaer, H. Flageole, F. A. Van Assche, and K. Vandenberghe, "Alternative technique for nd: Yag laser coagulation in twin-to-twin transfusion syndrome with anterior placenta," *Ultrasound Obstet. Gynecol. The Off. J. Int. Soc. Ultrasound Obstet. Gynecol.*, vol. 11, no. 5, pp. 347–352, 1998.
- [4] G. Dwyer, F. Chadebecq, M. T. Amo, C. Bergeles, E. Maneas, V. Pawar, E. Vander Poorten, J. Deprest, S. Ourselin, P. De Coppi, *et al.*, "A continuum robot and control interface for surgical assist in fetoscopic interventions," *IEEE Robotics Autom. Lett.*, vol. 2, no. 3, pp. 1656–1663, 2017.
- [5] T. Vandebroek, M. Ourak, C. Gruijthuijsen, A. Javaux, J. Legrand, T. Vercauteren, S. Ourselin, J. Deprest, and E. Vander Poorten, "Macro-micro multi-arm robot for single-port access surgery," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, pp. 425–432, IEEE, 2019.
- [6] M. A. Ahmad, M. Ourak, C. Gruijthuijsen, J. Deprest, T. Vercauteren, and E. Vander Poorten, "Deep learning-based monocular placental pose estimation: towards collaborative robotics in fetoscopy," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 15, no. 9, pp. 1561–1571, 2020.
- [7] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Europ. Conf. on Computer Vis.*, pp. 834–849, Springer, 2014.
- [8] K.-i. Okumura, S. Raut, Q. Gu, T. Aoyama, T. Takaki, and I. Ishii, "Real-time feature-based video mosaicking at 500 fps," in *Proc. IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, pp. 2665–2670, IEEE, 2013.
- [9] T.-Z. Xiang, G.-S. Xia, X. Bai, and L. Zhang, "Image stitching by line-guided local warping with global similarity constraint," *Pattern Recognit.*, vol. 83, pp. 481–497, 2018.
- [10] J. Zheng, Y. Wang, H. Wang, B. Li, and H.-M. Hu, "A novel projective-consistent plane based image stitching method," *IEEE Trans. on Multimed.*, vol. 21, no. 10, pp. 2561–2575, 2019.
- [11] C.-H. Chang, C.-N. Chou, and E. Y. Chang, "CLKN: Cascaded lucas-kanade networks for image alignment," in *Proc. IEEE Conf. on Computer Vis. and Pattern Recog.*, pp. 2213–2221, 2017.
- [12] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect visual odometry for monocular and multicamera systems," *IEEE Trans. on Robotics*, vol. 33, no. 2, pp. 249–265, 2016.
- [13] S. Bano, F. Vasconcelos, L. M. Shepherd, E. Vander Poorten, T. Vercauteren, S. Ourselin, A. L. David, J. Deprest, and D. Stoyanov, "Deep placental vessel segmentation for fetoscopic mosaicking," in *Proc. Int. Conf. on Medical Image Comput. and Computer-Assisted Interv.*, pp. 763–773, Springer, 2020.
- [14] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, " g^2o : A general framework for graph optimization," in *Proc. IEEE Int. Conf. on Robotics and Autom.*, pp. 3607–3613, IEEE, 2011.
- [15] M. Reeff, F. Gerhard, P. Cattin, and S. Gábor, "Mosaicking of endoscopic placenta images," in *INFORMATIK 2006 – Informatik für Menschen, Band 1* (C. Hochberger and R. Liskowsky, eds.), (Bonn), pp. 467–474, Gesellschaft für Informatik e.V., 2006.
- [16] P. Daga, F. Chadebecq, D. I. Shakir, L. C. G.-P. Herrera, M. Tella, G. Dwyer, A. L. David, J. Deprest, D. Stoyanov, T. Vercauteren, *et al.*, "Real-time mosaicking of fetoscopic videos using SIFT," in *Proc. Medical Imaging 2016: Image-Guided Procedures, Robotic Interv., and Model.*, vol. 9786, p. 97861R, International Society for Optics and Photonics, 2016.
- [17] F. Gaisser, S. H. Peeters, B. A. Lenseigne, P. P. Jonker, and D. Oepkes, "Stable image registration for in-vivo fetoscopic panorama reconstruction," *J. Imaging*, vol. 4, no. 1, p. 24, 2018.
- [18] L. Peter, M. Tella-Amo, D. I. Shakir, G. Attilakos, R. Wimalasundera, J. Deprest, S. Ourselin, and T. Vercauteren, "Retrieval and registration of long-range overlapping frames for scalable mosaicking of in vivo fetoscopy," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 13, no. 5, pp. 713–720, 2018.
- [19] S. Bano, F. Vasconcelos, M. T. Amo, G. Dwyer, C. Gruijthuijsen, J. Deprest, S. Ourselin, E. Vander Poorten, T. Vercauteren, and D. Stoyanov, "Deep sequential mosaicking of fetoscopic videos," in *Proc. Int. Conf. on Medical Image Comput. and Computer-Assisted Interv.*, pp. 311–319, Springer, 2019.
- [20] S. Bano, F. Vasconcelos, M. Tella-Amo, G. Dwyer, C. Gruijthuijsen, E. Vander Poorten, T. Vercauteren, S. Ourselin, J. Deprest, and D. Stoyanov, "Deep learning-based fetoscopic mosaicking for field-of-view expansion," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 15, no. 11, pp. 1807–1816, 2020.
- [21] S. Thrun and M. Montemerlo, "The graph SLAM algorithm with applications to large-scale mapping of urban structures," *The Int. J. Robotics Res.*, vol. 25, no. 5-6, pp. 403–429, 2006.
- [22] L. Carlone, R. Aragues, J. A. Castellanos, and B. Bona, "A fast and accurate approximation for planar pose graph optimization," *The Int. J. Robotics Res.*, vol. 33, no. 7, pp. 965–987, 2014.
- [23] N. Sünderhauf and P. Protzel, "Towards a robust back-end for pose graph SLAM," in *Proc. IEEE Int. Conf. on Robotics and Autom.*, pp. 1254–1261, IEEE, 2012.
- [24] F. Wu and G. Beltrame, "Cluster-based penalty scaling for robust pose graph optimization," *IEEE Robotics Autom. Lett.*, vol. 5, no. 4, pp. 6193–6200, 2020.
- [25] L. Carlone and G. C. Calafiore, "Convex relaxations for pose graph optimization with outliers," *IEEE Robotics Autom. Lett.*, vol. 3, no. 2, pp. 1160–1167, 2018.
- [26] I. Aloise and G. Grisetti, "Chordal based error function for 3-d pose-graph optimization," *IEEE Robotics Autom. Lett.*, vol. 5, no. 1, pp. 274–281, 2019.
- [27] Y. Tian, A. Koppel, A. S. Bedi, and J. P. How, "Asynchronous and parallel distributed pose graph optimization," *IEEE Robotics Autom. Lett.*, vol. 5, no. 4, pp. 5819–5826, 2020.
- [28] F. Chadebecq, F. Vasconcelos, R. Lacher, E. Maneas, A. Desjardins, S. Ourselin, T. Vercauteren, and D. Stoyanov, "Refractive two-view reconstruction for underwater 3d vision," *Int. J. Comput. Vis.*, pp. 1–17, 2019.
- [29] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. on Medical Image Computing and Computer-assisted Intervention*, pp. 234–241, Springer, 2015.
- [30] J.-Y. Bouguet *et al.*, "Pyramidal implementation of the affine Lucas Kanade feature tracker description of the algorithm," *Intel Corp.*, vol. 5, no. 1-10, p. 4, 2001.
- [31] M. Lopez, R. Mari, P. Gargallo, Y. Kuang, J. Gonzalez-Jimenez, and G. Haro, "Deep single image camera calibration with radial distortion," in *Proc. IEEE/CVF Conf. on Computer Vis. and Pattern Recog.*, pp. 11817–11825, 2019.