HHS Public Access

Author manuscript

IEEE Trans Med Imaging. Author manuscript; available in PMC 2021 February 01.

Published in final edited form as:

IEEE Trans Med Imaging. 2020 February ; 39(2): 400–412. doi:10.1109/TMI.2019.2927436.

Real-time Dense Reconstruction of Tissue Surface from Stereo Optical Video

Haoyin Zhou [Member, IEEE], **Jayender Jagadeesan [Member, IEEE]**

Surgical Planning Laboratory, Brigham and Women's Hospital, Harvard Medical School, Boston, MA, 02115, USA.

Abstract

We propose an approach to reconstruct dense three-dimensional (3D) model of tissue surface from stereo optical videos in real-time, the basic idea of which is to first extract 3D information from video frames by using stereo matching, and then to mosaic the reconstructed 3D models. To handle the common low texture regions on tissue surfaces, we propose effective post-processing steps for the local stereo matching method to enlarge the radius of constraint, which include outliers removal, hole filling and smoothing. Since the tissue models obtained by stereo matching are limited to the field of view of the imaging modality, we propose a model mosaicking method by using a novel feature-based simultaneously localization and mapping (SLAM) method to align the models. Low texture regions and the varying illumination condition may lead to a large percentage of feature matching outliers. To solve this problem, we propose several algorithms to improve the robustness of SLAM, which mainly include (1) a histogram voting-based method to roughly select possible inliers from the feature matching results, (2) a novel 1-point RANSAC-based P*n*P algorithm called as DynamicR1PP*n*P to track the camera motion and (3) a GPU-based iterative closest points (ICP) and bundle adjustment (BA) method to refine the camera motion estimation results. Experimental results on *ex*- and *in vivo* data showed that the reconstructed 3D models have high resolution texture with an accuracy error of less than 2 mm. Most algorithms are highly parallelized for GPU computation, and the average runtime for processing one key frame is 76.3 ms on stereo images with 960 × 540 resolution.

Index Terms—

Surface Reconstruction; Stereo Matching; SLAM; GPU Parallel Computation; Stereo Imaging

I. Introduction

THE surgeon's visualization during surgery is typically limited to the anatomical tissue surface exposed to him/her through an optical imaging modality, such as a laparoscope, endoscope or microscope. As a result, intraoperative identification of the critical structures

zhouhaoyin@bwh.harvard.edu;.

Jayender Jagadeesan owns equity in Navigation Sciences, Inc. He is a co-inventor of a navigation device to assist surgeons in tumor excision that is licensed to Navigation Sciences.

Dr. Jagadeesans interests were reviewed and are managed by BWH and Partners HealthCare in accordance with their conflict of interest policies.

lying below the visual surface is difficult and could lead to inadvertent complications during the surgery. To solve this problem, many surgical navigation systems utilize models of tissue surface, internal structures and tumors segmented from preoperative MR/CT imaging for intraoperative guidance. However, direct registration between two-dimensional (2D) optical (microscopy, endoscopy or laparoscopy) videos and three-dimensional (3D) MR/CT images is difficult and highly non-trivial. To overcome the difficulty in registering the multimodal images, 3D information can be extracted from 2D optical videos, which is still an open problem and is especially challenging when the surface texture is low. In this paper, we propose a series of novel methods to reconstruct textured 3D models of tissue surfaces from stereo optical videos in real-time. The textures on the reconstructed tissue surface models have the same resolution as the input video frames, which can greatly facilitate surgical navigation for the following reasons: (1) During surgery, only a small area of the target tissue may be exposed and landmarks that can be automatically recognized are often invisible. In addition, blood or surgical smoke may occlude the target tissue. Hence, it is important to provide high resolution textures to help the clinicians to recognize the tissue from the reconstructed models and then perform manual registration. (2) Intuitive visual feedback as part of a surgical navigation system is also very important for tumor localization. And with high resolution textures, clinicians are able to visualize the *in vivo* scene from different angles intuitively.

Stereo optical imaging modalities have been widely used in the operating room to provide depth perception to the surgeon. In the past decade, many efficient stereo matching methods have been proposed to estimate depths of image pixels by establishing pixel-to-pixel correspondences between the stereo images, the results of which can be further refined to generate fine 3D models. Stereo matching methods can be roughly classified into global and local methods. Global methods use constraints on scan-lines or the whole image [1] [2], which are able to handle low texture regions by using explicit or implicit interpolation. However, global methods have high computational complexity and are inappropriate for real-time applications. In contrast, local methods only use constraints on a small number of pixels surrounding the pixel of interest [3], which are fast but are difficult to handle low texture regions. In this paper, we propose effective outliers removal, hole filling and smoothing methods as the post-processing steps for the local stereo matching methods, which have low computational complexity low and are appropriate for graphics processing unit (GPU) parallel computation.

Stereo matching-based 3D reconstruction is highly dependent on the texture of the observed object. However, the surface texture of tissues, such as lung and liver, is not rich enough to be observed at a distance due to the limited camera resolution and poor illumination condition. Another important reason to use a small camera-tissue distance is that the baseline of stereo imaging modalities is usually short, which result in large uncertainties when estimating large depths. However, due to the limited field of view, a small camera-tissue distance will lead to only a small area of the surface that can be reconstructed from the pair of stereo images, which is insufficient to perform accurate registration between pre- and intraoperative 3D models [4]. To solve the contradiction between the accuracy of 3D reconstruction and registration, we propose to scan the tissue surface at a close distance and perform stereo matching on the acquired stereo images, then mosaick the 3D models at

different time steps according to model alignment obtained by simultaneously localization and mapping (SLAM).

SLAM is one of the most important topics in the robotics navigation field, which aims to estimate the camera motion and reconstruct the surrounding environment in real-time [5] [6]. To date, SLAM methods have proven effective in reconstructing large environments and estimating long motions [7], hence it is a reasonable assumption that the accumulative errors of SLAM methods is minimal for the small *in vivo* environments. SLAM methods are often based on feature points matching to establish correspondences between video frames. However, for tissue surfaces with low and/or repeating texture under varying illumination conditions, feature matching is challenging [8] and a large percentage of matching outliers may cause failure of the SLAM methods. In order to overcome the difficulties in feature matching and improve the robustness of mosaicking, we first propose a novel histogram voting-based method to select possible inliers from the feature matching results. Then, using the selected possible inliers as the control points, we extend our previous work [9] and propose a novel perspective-*n*-points (P*n*P) algorithm called as DynamicR1PP*n*P to estimate the camera motion, which can remove incorrect and build new matches dynamically. Finally, we propose to integrate feature matching and iterative closest points (ICP)-based costs into an optimization method to refine the camera motion estimation results. The main algorithms involved in our SLAM framework are implemented in CUDA C++ and run on the GPU.

This paper is organized as follows: In Section II, we describe the process of the stereo matching method and provide the details of its GPU implementation. The SLAM-based model mosaicking method, including histogram voting-based inliers selection, DynamicR1PP*n*P and GPU-based BA+ICP, is introduced in Section III. Evaluation results on *ex vivo* and *in vivo* data are presented in Section IV. A discussion is presented in Section V.

A. Related Works

Optical video-based 3D reconstruction of tissue surfaces can improve the accuracy of intraoperative navigation [10]. Stereo matching is one of the most effective 3D reconstruction methods in surgical navigation applications [11], which estimates pixel disparities by comparing stereo images. Stereo matching is an important and active topic in the computer vision field and a large number of effective methods exist, and reader may refer to the Middlebury website [12] for the list of stereo matching methods. Stereo matching methods can be roughly classified into global and local methods. Global stereo matching [12], such as dynamic programming [13] and graph cuts [14], exploit nonlocal constraints to reduce sensitivity to regions that fail to match due to low texture and occlusions, which make explicit smoothness assumptions to solve an optimization problem. However, the high time complexity makes global stereo matching difficult to be real-time [15], hence most current real-time 3D reconstruction systems are based on local stereo matching.

Local stereo matching methods estimate disparities of pixels by computing matching matrices between small and local image patches. There exist many metrics to evaluate the similarity between two image patches [16]. The most straightforward one is window-based

matching costs, which compare the differences of squared image windows. Zero-mean normalized cross-correlation (ZNCC) [17] is one of the most effective window-based costs due to its good robustness to illumination changes. However, such squared window-based methods cannot handle pixels near object edges because they may belong to different surfaces. To overcome this problem, non-parametric matching costs, such as rank and census methods [18] and ordinal measures [19], were proposed to handle object boundaries. Another class of effective methods is based on support window methods [20], such as PatchMatch [3], which uses varying shape of the matching window. To achieve better accuracy, researchers propose to dynamically update the weights of pixels within the support window [21]. For our task, the needs of handling tissue edges or occlusion are not high because usually only one target tissue needs to be reconstructed and the surgeons may simply remove the instrument during the scan. Hence we use ZNCC matching in our method, which is fast on the GPU. Our main contribution of the stereo matching part is that we propose several effective post-processing steps to address the low texture problem, which can also be used for the refinement of other local stereo matching methods.

Many real-time stereo matching systems are based on ZNCC [17]. To achieve real-time performance, it is essential to reduce the number of candidate disparities for local stereo matching methods. For example, Bleyer et al [3] proposed an effective disparities searching strategy by first generating disparities for all pixels randomly, and then iteratively replacing the disparity of a pixel with that of its neighboring pixel if the new value suggests a better ZNCC matching. Stoyanov et al [22] [23] matched a sparse set of salient regions using stereo Lucas-Kanade and propagated the disparity around each matched region. They reported a 10Hz updating rate for images with 360 × 240 resolution. The development of GPU or FPGA [24] based parallel computational algorithms can greatly accelerate the image patch matching process [25]. Zollhöfer et al [26] reported a 100 Hz update rate for stereo images with 1280 × 1024 resolution using a NVIDIA Titan X GPU. Our CUDA C++ implementation achieves a 200 Hz updating rate for the 960 × 540 resolution and 100 candidate disparities, which is sufficient for our surface reconstruction system.

3D models generated by stereo matching are limited to the field of view, which may be too small for surgical guidance. Structure-from-motion (SfM) [27] or simultaneously localization and mapping (SLAM) [28] [29] [30] methods are able to align video frames at different time steps and generate a much larger synthetic field of view, which have been employed for 3D reconstruction of tissues. For example, Mountney et al [31] proposed to expand the field of view based on SLAM. Most SfM and SLAM methods only reconstruct sparse feature points, which poorly describe the surgical scene.

Dense SLAM methods have also been developed to generate dense tissue models in real-time. Totz et al [32] proposed an EKF-SLAM-based method for dense reconstruction. EKF-SLAM suffers from low accuracy and is difficult for representing loop closing. Recently, Mahmoud et al [33] proposed a monocular vision-based dense tissue 3D reconstruction method by using ORB-SLAM [6] to estimate the camera motion. However, because ORB-SLAM is based on ORB features and RANSAC+P3P [34] for camera motion tracking and loop closing, its robustness is not satisfying with low texture scenes. In this paper, we

propose novel camera motion tracking algorithms and a more robust SLAM framework to improve the robustness of camera pose estimation with low texture surfaces.

Another effective way to perform real-time dense reconstruction is to combine sparse SLAM and stereo vision, the idea of which is closely related to the famous KinectFusion work [35], which merges the raw depth map provided by Microsoft Kinect to generate the fine models. It is a natural idea to replace the depth map with the results of stereo matching. However, the most difficult part is to align the depth map by SLAM, and KinectFusion is based on the ICP method. However, due to the narrow field of view and the smooth surface of tissue, ICP-based alignment cannot achieve accurate registration in the tangential directions.

II. Stereo Matching

After stereo camera calibration, physical depthes of stereo image pixels can be directly computed from the disparities. We used the Matlab Computer Vision Toolbox to calibrate the stereo laparoscope and our C++ code to convert image disparities to physical depthes is equivalent to the Matlab ‘reconstructScene’ function. For local stereo matching methods, the estimation of disparities at low texture regions is difficult due to the lack of direct corresponding information between left and right images. However, low texture regions are common on tissue surfaces due to tissue optical properties, limited image resolution, poor image quality and poor illumination conditions. Most stereo matching methods rely on interpolation to propagate information from highly textured regions to low texture regions. For example, by interpolating between edges, a textureless flat wall can be reconstructed accurately. However, tissue surfaces have more complex shapes, and interpolation-based methods may not be accurate at distant regions. Hence, we do not seek to estimate disparities of all pixels in the stereo matching step, but rely on the subsequent mosaicking step to generate more complete and larger models of tissue surfaces.

To overcome the high time complexity drawback of global stereo matching methods and difficulty to handle low texture regions of local stereo matching methods, we propose a novel stereo matching framework as shown in Fig. 1 to enlarge the radius of constraints of local stereo matching. First, we employed the zero-mean normalized cross correlation (ZNCC) metric to evaluate similarities between local image patches to estimate disparities of pixels. Then, we developed a robust outliers removal and hole filling method to refine the ZNCC matching results. The first two steps provide discrete initial disparity values that are from the candidate disparities pool for the final refinement step, where we integrate the ZNCC metrics and the smoothing cost into a modified Laplacian smoothing framework. This method is able to build large connections among pixels when minimizing the cost function, and is easier to compute than conventional Gauss-Newton (GN) or Levenberg-Marquardt (LM) algorithms. It is worth clarifying that we are not implying that ZNCC is the best metrics, however since our stere matching methods are mostly post-processing steps, it is easy to replace ZNCC with other local matching metrics. The algorithms in our stereo matching method work in parallel with respect to each pixel, and are highly appropriate for GPU parallel computing.

A. ZNCC-based Local Matching

The most widely used local stereo matching method first generates disparities for all pixels randomly, and then iteratively replaces the disparity of a pixel with that of its neighboring pixel if the new value suggests a better matching [3]. This process has demonstrated high efficiency and even CPU-based serial computation can be real-time (2–3 Hz). Another advantage is that this type of method implicitly takes into account smoothing among pixels. However, in practice we found that this method is not suitable for the case of smooth tissue surface because pixels that have the same disparity are often distributed in a narrow belt, which makes it difficult to propagate a correct disparity value and many iterations are needed. In addition, these methods cannot make full use of the GPU parallel computing ability, because the propagation process can only be parallelized to W and H threads alternatively, where W and H are image width and height respectively.

Our stereo matching method is based on the ZNCC metrics to evaluate similarities between local image patches. In our experiments we use a window size of 11 × 11 pixels. To make full use of GPU parallel computing ability, we develop a brute force way by launching GPU threads for each pixel to test the candidate disparity values. To achieve higher computational speed, in the matching window we only use every other pixel values, which is distributed as a chessboard. The details of our GPU implementation are briefly described as follows: For images with a resolution of $W \times H$, our CUDA implementation launches H CUDA blocks and each CUDA block has W threads. We cache neighboring image rows into the GPU shared memory for each CUDA block to avoid the slow I/O speed of global memory. With a 960 × 540 resolution and 100 candidate disparity values, the runtime of our GPU-based ZNCC matching method is around 5ms.

B. Outliers Removal and Hole Filling

The initial ZNCC matching may result in a large amount of outliers. Our outliers removal and hole filling method is under a reasonable assumption that the tissue surface is relatively smooth. Hence, an inlier should have sufficient number of neighboring points that has smooth change of disparities. Denoting r as the detection radius, we detect along each 8-radial directions within radius r and check if the disparity of two neighboring points is smaller than a pre-defined threshold (= 2.5 in our experiments). If none of the 8-radial directions satisfies this smooth disparity assumption, the point will be recognized as an outlier and removed.

We developed two hole filling methods. For a left-image pixel that cannot find its corresponding right-image pixel, the first method searches along the pixel's 8-radial directions and the second method searches within a radius of the pixel. In our experiment the two radii for the hole filling methods are fixed, which are 50 and 20 pixels respectively. If sufficient number of neighboring points have a valid disparity value, then disparity of this pixel is filled according to interpolation. In the hole-filling step the iterations are performed within a radius, which avoids interpolation at distant areas.

However, when removing outliers, it is difficult to predefine a radius r for all cases. A small r may keep too many outliers and a large r may remove inliers. To removal outliers and

preserve as many inliers as possible, we propose to use an iterative process that alternately performs outlier removal and hole filling, as shown in Fig. 1(a). In this process we gradually enlarge r with a step Δr when detecting outliers. Hence, disparities that are removed may then be filled, and neighboring inliers will not be removed with larger r. In our experiments, the number of outliers removal and hole filling iterations is 3; the radius r is 10 pixels initially and increases at a step of $\Delta r = 10$ pixels.

C. Improved Laplacian Smoothing-based Refinement

Further step to refine the estimated disparities is necessary because (1) the initial disparities after the first two steps are discrete values that are directly selected or interpolated from the candidate disparities and (2) relationships among pixels are not fully considered. Our refinement method is based on Vollmer's improved Laplacian smoothing method [36], which is able to avoid model shrinking compared with standard Laplacian smoothing. We integrate a cost function that consists of the ZNCC metrics and the smoothing cost into this improved Laplacian framework to allow for dynamically updating the disparities. The details of our refinement step are as follows:

We denote the discrete disparity of a pixel i as o_i, which initially is equal to the disparity value after the first two steps. The smoothed disparities at the kth iteration are denoted as $d_i^{(k)}$. After an initialization $d_i^{(0)} := o_i$, the refinement method performs the following steps in the kth iteration:

$$d_i^{(k)} := \text{average}(o_j), \tag{1}$$

where j is the index of neighboring pixels of point i within a pre-defined radius, and we use the smoothing radius of 15 pixels in our experiments.

$$b_i := d_i^{(k)} - \alpha o_i - (1-\alpha) d_i^{(k-1)}, \tag{2}$$

where b_i is introduced to avoid model shrinking. $\alpha \in [0, 1]$ is a weighting coefficient and $\alpha = 0.1$ in our experiment. And then

$$d_i^{(k)} := d_i^{(k)} - \text{average}(b_j). \tag{3}$$

Equations (1), (2) and (3) are derived from Vollmer's Laplacian smoothing method, which generate continuous disparities d_i by smoothing discrete disparities o_i. We further propose to update the discrete disparities o_i in each iteration according to the minimization of a cost function that consists of the ZNCC metrics and the smoothing cost. Specifically, with an updated disparity $d_i^{(k)}$ in the iteration, we search within a disparity range $[d_i^{(k)} - 5, d_i^{(k)} + 5]$, and update the o_i to the disparity value that minimizes

$$o_i := \arg\min_{o_i^*} f_{\text{zncc}}(o_i^*) + \eta f_{\text{smooth}}(o_i^* - d_i), \tag{4}$$

where $f_{\text{zncc}}(o_i^*)$ is the ZNCC matching cost, which equals to the reciprocal of the ZNCC matching value when using a disparity o_i^*. $f_{\text{smooth}}(o_i^* - d_i) = (o_i^* - d_i)^2$ is the smoothing cost because d_i is the smoothed value of neighboring o_j. η is a coefficient. The size of matching window affects $f_{\text{zncc}}(o_i^*)$, and with a 11 × 11 pixels window, we use $\eta = 0.01$ in our experiments.

The advantage of using this improved Laplacian smooth framework is that it is able to naturally make use of the dynamically updated discrete disparities. This method is highly parallel to each pixel and suitable for GPU computation.

III. Model Mosaicking

We employ the truncated signed distance field (TSDF) method [37] to mosaic the raw 3D point cloud generated from pixel disparities results of stereo matching and the camera calibration parameters to obtain the extended 3D model of the tissue surface, as shown in Fig. 3. The prerequisite to perform TSDF is to align the raw 3D point cloud accurately, which is equivalent to the estimation of camera motion in this video-based 3D reconstruction problem. As shown in Fig. 4, conventional iterative closest points (ICP)-based model alignment is difficult to handle smooth tissue surfaces. Another way to align models is based on image feature points matching. However, due to the low texture and varying illumination condition, feature matching is challenging and a large amount of outliers may exist. To overcome these problems, we propose a novel SLAM method that consists of fast and robust algorithms to handle the large percentage of feature matching outliers in real-time.

The flow chart of our SLAM method is shown in Fig. 2, which mainly consists of three modules. The first module tracks the camera motion between adjacent video frames according to ORB feature matching [38], which is mainly based on a novel and robust P*n*P algorithm called DynamicR1PP*n*P. The second module aims to refine the camera motion estimation results at key frames and eliminate the accumulative error, which is based on the minimization of ICP and bundle adjustment (BA) costs. The third module performs TSDF-based model mosaicking and manages feature points. In the following section, we will introduce the details of the involved algorithms.

A. Histogram Voting-based Matching Inliers Preselection

Our SLAM system is based on the ORB feature [38], which is much faster to detect and match than the conventional SURF feature [39], and has been widely used in real-time SLAM systems, such as the ORB-SLAM method [6].

However, the low and/or repeating texture of the tissue surface and varying illumination condition may result in a large amount of incorrect feature matches. In practice we observed that the percentage of outliers may be larger than 85%, making the traditional RANSAC +P3P [40]-based outliers removal method slow. In addition, the small number of correct matches also decreases the accuracy of camera motion estimation. Hence, it is necessary to design algorithms to handle the large percentage of matching outliers.

ORB matching is performed between two adjacent video frames for camera motion tracking. Under a reasonable assumption that the camera motion, especially the roll angle, between adjacent video frames is minimal during the surface scan, we propose to utilize the displacements of matched ORB features between two adjacent images to roughly distinguish correct and incorrect ORB matches. Specifically, we denote the image coordinates of matched ORB features at two images as $[u_i^{(1)}, v_i^{(1)}]$ and $[u_i^{(2)}, v_i^{(2)}]$, $I = 1, \ldots, N$, where u and v are the x– and y– image coordinates in pixels. A correct match k should have a similar displacement $[u_i^{(2)} - u_i^{(1)}, v_i^{(2)} - v_i^{(1)}]$ with other correct matches. Hence, we first generate the histogram of $[u_i^{(2)} - u_i^{(1)}, v_i^{(2)} - v_i^{(1)}]$, and then consider the ORB matches that are close to bins with large histogram value more likely to be inliers, which will be assigned with higher priority to be the control points for the subsequent DynamicR1PPnP algorithm. It should be clarified that this histogram voting-based inliers preselection step may not be 100% correct, but it is fast and able to remove a large amount of outliers fast for the subsequent steps of the SLAM method.

B. DynamicR1PPnP

PnP methods, which aim to estimate the position and orientation of a calibrated camera from n known matches between 3D object points and their 2D image projections, have been widely used in SLAM systems for camera motion estimation. We propose to modify and improve our previous R1PPnP work [9] to handle the problem of small number of matching inliers in the task of tissue surface reconstruction. In this section, we first briefly introduce the original version of R1PPnP and then introduce our modification.

R1PPnP is based on the standard pin-hole camera model, which is

$$u_i = f\frac{x_i^c}{z_i^c}, v_i = f\frac{y_i^c}{z_i^c}, \tag{5}$$

where f is the camera focal length, $\mathbf{x}_i = [u_i, v_i, f]^T$ is the image homogeneous coordinate in pixels, and $\mathbf{X}_i^c = [x_i^c, y_i^c, z_i^c]^T$ is the real-world coordinate with respect to the camera frame. Hence, we have

$$\mathbf{X}_i^c = \lambda_i^* \mathbf{x}_i, \tag{6}$$

where $\lambda_i^* = z_i^c / f$ is the normalized depth of point i.

The relationship between the camera and world frame coordinate of point i is

$$\mathbf{X}_i^c = \mathbf{R}\mathbf{X}_i^w + \mathbf{t}, \tag{7}$$

where $\mathbf{R} \in SO(3)$ is the rotation matrix and $\mathbf{t} \in R^3$ is the translation vector. $\mathbf{R}$ and $\mathbf{t}$ are the variables that need to be estimated in the PnP problem. Selecting a point o as the control point, we have

$$\mathbf{X}_i^c - \mathbf{X}_o^c = \mathbf{R}\left(\mathbf{X}_i^w - \mathbf{X}_o^w\right),\ i \neq o\,. \tag{8}$$

Denoting $\mathbf{S}_i = \mathbf{X}_i^w - \mathbf{X}_o^w$, then, according to (6) and (8),

$$\lambda_i^* \mathbf{x}_i - \lambda_o^* \mathbf{x}_o = \mathbf{R}\mathbf{S}_i\,. \tag{9}$$

We divide both sides of (9) by the depth of the control point λ_o^*, and rewrite (9) as

$$\lambda_i \mathbf{x}_i - \mathbf{x}_o = \mu \mathbf{R}\mathbf{S}_i, \tag{10}$$

where $\lambda_i = \mu\lambda_i^*$ and $\mu = 1/\lambda_o^*$ is the scale factor. We have

$$\mathbf{t} = 1/\mu \mathbf{x}_o - \mathbf{R}\mathbf{X}_o^w\,. \tag{11}$$

which suggests that **t** can be computed from **R** and μ.

The geometrical relationships of R1PP*n*P is shown in Fig. 5. R1PP*n*P combines a re-weighting strategy and the 1-point RANSAC framework to reduce the effects of outliers. The 1-point RANSAC framework randomly selects one match as the control point o and then alternatively update **R**, μ and λ_i to minimize the cost function

$$f(\mathbf{R}, \mu, \lambda_i) = \sum_{i=1, i \neq o}^{N} w_i \|\lambda_i \mathbf{x}_i - \mathbf{x}_o - \mu \mathbf{R}\mathbf{S}_i\|^2, \tag{12}$$

where w_i is the weight of point i and is dynamically updated in the iteration process according to

$$w_i = \begin{cases} 1.0 & if\ e_i \leq H \\ H/e_i & if\ e_i > H \end{cases}, \tag{13}$$

where e_i suggests the reprojection error of point i with the current **R** and μ during iteration. H is the inliers threshold that points with final reprojection errors smaller than H are considered as inliers, and in our experiments we use $H = 5$ pixels. The reweighting rule (13) suggests that a point with a large reprojection error will have a small weight during the estimation of camera pose, as shown in Fig.6. Our experimental results in Ref. [9] showed that R1PP*n*P has state-of-the-art performance compared with conventional RANSAC+P3P methods to handle matching outliers.

In our SLAM method, we will use the preselected matches according to the histogram voting results as the control points o in the R1PP*n*P algorithm. However, when the number of feature matching inliers is small, R1PP*n*P or conventional RANSAC+P3P methods cannot estimate the camera motion accurately. Compared to obtaining correct matches, detecting consistent sets of feature points from two overlapped images is relatively easier. Based on this observation, we modified the R1PP*n*P method to dynamically update the feature

matching relationships as follows: The camera pose is updated in an iterative process and the re-projection of stored feature points is updated accordingly. When the distance between a current feature point i and the re-projected stored feature point j is small, it should be considered as a possible correct match and we will add this candidate match dynamically, and the weight $w_{i,j}$ is updated according to

$$w_{i,j} := \min\left(H/e_{i,j}, 1\right) \quad if\ e_{i,j} < \eta H, \tag{14}$$

where $\eta > 1$ is a coefficient and in our experiments we use $\eta = 2.0$. Then, we perform normalization by

$$w_{i,j} := w_{i,j} / \left(\sum_j w_{i,j} + w_i \right), \tag{15}$$

where w_i is the weight of the original matches provided by ORB feature matching.

In order to eliminate incorrect matches that happen accidently in the iteration process, we decrease the weight $w_{i,j}$ if it is a newly observed candidate correspondence.

$$w_{i,j} := \min((k - k_0)/T, 1) w_{i,j} \tag{16}$$

where k is the current iteration index, k_0 is the first iteration index when this candidate correspondence is observed. T is a pre-defined number and in our experiments we use $T = 5$.

C. Key Frame Decision

A frame is recognized as a key frame if it satisfies the following conditions: (1) There are at least 50 correct ORB matches when using DynamicR1PPnP, and (2) It has been at least 10 frames since the last key frame, or the difference between the camera pose of this frame and the last key frame is larger than a threshold. The camera pose difference is defined as

$$\|\mathbf{t}_{\text{diffence}}\| + 20\min\left(\|\mathbf{E}_{\text{diffence}}\|, 2\pi - \|\mathbf{E}_{\text{diffence}}\|\right), \tag{17}$$

where $\mathbf{E}_{\text{diffence}}$ and $\mathbf{t}_{\text{diffence}}$ are the differences between the Euler angles and translations respectively. In our experiments we may use different pose thresholds for different data because the number of video frames and the tissue scale varies. A large pose threshold suggests that key frames are distant from each other and due to illumination changes, the textures on the mosaic may not look very smooth, but in general this pose threshold that determines key frames is not sensitive.

D. Refinement of Camera Motion Estimation

In the camera motion tracking stage, the 3D coordinates of feature points are directly obtained from stereo matching. The estimated camera poses are not accurate enough and bundle adjustment (BA)-based refinement [41] is necessary. We also take into account the ICP-based distance between current stereo matching model and the existing model to improve the robustness of the SLAM method because feature matching with previous key frames may fail due to low texture.

We first try to match the ORB features of current key frames with those of previous key frames to eliminate accumulative error. Our SLAM algorithm stores the feature points of previous key frames for reducing the accumulative error. With the camera motion tracking results, we select several previous key frames that have enough overlapped areas with the current key frame as the candidates. Then, we perform ORB matching with the candidate previous key frames and perform DynamicR1PP*n*P to detect correct matches.

Then we apply the optimization method to refine the camera motion estimation results. At a key frame with index T, we refine the camera pose estimation results by minimizing the cost function

$$f_{\text{total}}(\mathbf{R}_t, \mathbf{t}_t, \mathbf{x}_i) = f_{\text{BA}}(\mathbf{R}_t, \mathbf{t}_t, \mathbf{x}_i) + \beta f_{\text{ICP}}(\mathbf{R}_T, \mathbf{t}_T), t \in \Omega \tag{18}$$

where Ω is the set of indices of video frames, which includes the current key frame T, all frames between the last key frame and current key frame T, and the matched previous key frames. $\mathbf{R}_t$ and $\mathbf{t}_t$ are the camera rotation and translation at video frame t respectively. x_i is the coordinate of feature point i. β is a weighting coefficient, which is dynamically adjusted according to the ratio of the number of feature points and ICP points. In our experiment we use $\beta = 0.1 \times$ number of feature points/number of ICP points.

The first term of (18), $f_{\text{BA}}(\cdot)$, is the standard local BA cost that aims to minimize the re-projection error, which only considers video frames that are included in Ω. In this term, we fix the pose of the last key frame and the feature points observed in the last key frame to avoid scale drift.

The second term of (18), $f_{\text{ICP}}(\cdot)$, aims to minimize the distance between the existing 3D model and the current stereo matching model at key frame T, which is

$$f_{\text{ICP}}(\mathbf{R}_T, \mathbf{t}_T) = \sum_i \rho\psi(\mathbf{n}_i(\mathbf{R}_T\mathbf{p}_i + \mathbf{t}_T - \mathbf{q}_i)), \tag{19}$$

where $\mathbf{p}_i$ are points of the existing model, and $\mathbf{q}_i$ are points of the current stereo matching model that has the same re-projection pixel coordinate with $\mathbf{R}_T\mathbf{p}_i + \mathbf{t}_T$. $\psi(\cdot)$ is Tukey's penalty function to handle outliers. $\rho = 1$ if $\mathbf{q}_i$ has a valid depth, otherwise $\rho = 0$. $\mathbf{n}_i$ is the normal direction of $\mathbf{q}_i$ obtained from the stereo matching point cloud, which allows the template to 'slide' along the tangent directions, as shown in Fig. 4.

To minimize the cost function (18), a GPU-based parallel Levenberg-Marquardt (LM) algorithm is developed. The equation in the standard LM algorithm to update the variables is

$$\left(\mathbf{J}^T\mathbf{J} + \lambda diag\left(\mathbf{J}^T\mathbf{J}\right)\right)\mathbf{x} = \mathbf{J}^T\mathbf{b}, \tag{20}$$

where x is the vector of variables, **J** is the Jacobian matrix and b is the residual vector. λ is a parameter that controls the updating step.

According to Eq. (18), the variables to be estimated consist of camera poses and coordinates of feature points. Since a feature point may exist in most of the recent video frames, the structure of the whole Jacobian matrix **J** is large and dense. In order to accelerate the

computation, we split the variables into two parts in our LM implementation and update the two parts of variables alternatively.

To update the coordinates of feature points, because each feature point is independent with respect to each other when the camera poses are fixed, we launch one GPU thread for each feature point and calculate the related Jacobian matrix and the residual re-projection errors.

We update the camera poses of different frames separately. Because only the camera pose at key frames considers the ICP term (19), hence the main parameters of Eq. (20) at key frames can be split to

$$\mathbf{J}^T\mathbf{J} = \mathbf{J}_{\mathrm{BA}}^T\mathbf{J}_{\mathrm{BA}} + \beta^2\mathbf{J}_{\mathrm{ICP}}^T\mathbf{J}_{\mathrm{ICP}}, \tag{21}$$

and

$$\mathbf{J}^T\mathbf{b} = \mathbf{J}_{\mathrm{BA}}^T\mathbf{b}_{\mathrm{BA}} + \beta^2\mathbf{J}_{\mathrm{ICP}}^T\mathbf{b}_{\mathrm{ICP}}. \tag{22}$$

We launch multiple parallel GPU threads to compute each row of **J**. Then, we perform Cholesky decomposition to solve (20).

E. GPU-based TSDF Mosaicking

The basic idea of TSDF is to take the average value of the 3D coordinates of an area if it is observed multiple times, which is more accurate than the results of a single observation. Raw 3D point cloud can be obtained from key video frames by using our stereo matching method. We incrementally mosaic the stereo matching results to generate the extended tissue surface model based on the camera motion estimation results of SLAM. The extended tissue surface models are also in the form of 3D point clouds. Because we aim to obtain high resolution textures to provide better surgical navigation, the extended surface model usually include millions of points and traditional volume-based TSDF method may take too large amount of computer memory. To avoid this problem, we store the 3D coordinate and the RGB color for each point in the GPU memory without using volume grids. To build correspondences between the extended surface model and the current stereo matching results, we project the extended surface model to the current imaging plane according to the camera pose estimation results. This rasterize process is performed by using GPU parallel computation and is fast. For each pixel with a valid depth value in the stereo matching results, the related point in the extended surface model is merged with the stereo matching results by using the TSDF method. Pixels that are not covered by the reprojection are considered as new points and will be added to the extended surface model.

Since the light source is often equipped at the tip of the imaging modality, hence the image edge is often darker than the central area. In order to generate smooth texture of the model, we also use TSDF-like merging to update the RGB color of points. During RGB color merging, the TSDF updating weight is 1.0 if the point is at the center area of the image, and decreases as it approaches the image edge.

IV. Experiments

The source code was implemented in CUDA C++ and executed on a Dell desktop with an Intel Xeon X5472 3.00 GHz CPU and NIVIDA Titan X GPU. We used OpenCV to read the recorded videos and the results were visualized by the Visualization Toolkit (VTK). We collected *ex*- and *in vivo* stereo videos for the evaluation of our method, and details of the videos are provided in Tab. I and II, which includes the video length, number of frames, average tissue-camera distance, average camera motion speed, size of the tissue and the number of points of the reconstructed models.

A. Ex Vivo Experiments

We first qualitatively tested our 3D reconstruction method on phantoms and *ex vivo* tissues, including porcine stomachs and livers. We used a KARL-STORZ stereo laparoscope (model number TipCam 26605AA) with a resolution of 960 × 540 to capture stereo videos and performed the proposed 3D reconstruction method on the videos. The candidate disparity values for performing ZNCC matching are between −20 and 80 pixels. Details of the videos are provided in Tab. I. The results of our *ex vivo* qualitative experiments are shown in Fig. 8. Since down-sampling is not included in the reconstruction process, the obtained 3D models have the same resolution as the input image, which usually include millions of points and are able to provide rich details of the surface texture. Our results qualitatively look promising and accurate. We also employed ORB-SLAM2 [6] for comparison, which is one of the most famous open-source SLAM methods. In order to handle low texture, the key parameters of ORB-SLAM2 were set as follows: the number of feature points is 3000 per image, and the threshold for detecting FAST corner is 1. As shown in Fig. 8(a) and (c), ORB-SLAM2 succeeded in reconstructing the sparse environment and tracking the camera motion. However ORB-SLAM2 tracking lost in cases shown in Fig. 8(b) and (d) due to the low texture.

In order to evaluate the quantified accuracy of our 3D reconstruction method, we used the CT imaging of tissues as the gold standard. In this experiment, CT scans of four ex-vivo porcine livers and three kidneys were obtained (Siemens Somatom, Erlangen Germany) with a 0.6 mm resolution at our hospital, and we used the 3D Slicer software to segment the tissue models from the CT images, as shown in Fig. 9. We captured stereo videos of the tissues with the KARL-STORZ stereo laparoscope, the details of which are in Tab. II. Surfaces of livers and kidneys are very smooth and have low textures, but the proposed method was still able to reconstruct the 3D models, as shown in Fig. 7. To quantify accuracy, we registered the 3D reconstructed model with the CT segmentation results by first manually selecting landmarks, such as tissue tips, edge points and other recognizable points, and then refining the registration with the ICP algorithm. As shown in Fig. 7 (a), the root mean square errors (RMSE) with the liver cases are 1.3, 1.1, 1.4 and 2.0 mm respectively. The fourth liver case has a relatively larger error because we used an entire piece of liver and the video was captured at a larger camera-tissue distance. The results on porcine kidneys are shown in Fig. 7 (b), the RMSE of which are 1.0, 1.0 and 1.1 mm respectively. The histograms of errors are also provided in Fig. 7, which show that most points have an error of less than 2mm. It is worth noting that there are multiple sources of errors, including 3D reconstruction error, CT

resolution error, CT segmentation error and registration error that contribute to the obtained RMSE in this experiment. In addition, because the livers and kidneys were placed on a textureless plastic sheet and part of the sheet were also included in the 3D reconstructed model, which is difficult to be totally removed (see the tissue edges in the distance maps of Fig. 7), so the quantified error may also include a small amount of the background. Therefore, it is a reasonable assumption that the actual error of our 3D reconstruction method is smaller than the reported RMSE.

B. In Vivo Experiments

To further evaluate the performance of our surface reconstruction method in real-world surgical scenarios, we obtained intraoperative videos from various stereo imaging modalities during surgeries performed in our hospital and online videos. The details of the videos are provided in Tab. I. The videos were captured under an Institution Review Board approved protocol. Patient consent was waived since the analysis was performed retrospectively and no clinical decisions were affected.

For the first set of experiments, we obtained intraoperative stereo microscope images during a neurosurgery case. The dual channel output from a Carl-Zeiss microscope was captured using an Epiphan video capture card (DVI2PCI Duo) using the 3D Slicer software [42]. Five image frames with resolution 720 × 480 with small overlap between the frames were used to create a high-resolution mosaic of the surgical cavity. The results of the stereo reconstruction and mosaicking algorithms are shown in Fig. 10. In this experiment, we simply set the pose threshold to determine key frames to a small number hence all five images were used as key frames. Such a high-resolution mosaicking of the neurosurgery cavity could conceivably be used to register the intraoperative or diagnostic MRI to the mosaicked stereo reconstruction of the surgical cavity to identify remnant brain tumor during surgery. Due to the too small number of images, we did not run ORB-SLAM2 for this case.

For the next set of experiments, we obtained high resolution stereo laparoscopy images of the kidney during a robot-assisted partial nephrectomy case. The video was obtained from the dual channel DVI output of the master console of the Intuitive da Vinci Xi robot. The video has the resolution of 1024 × 768, and was captured using two Epiphan video capture cards (DVI2PCI Duo) and a simple video capture program implemented using OpenCV. Prior to tumor resection, the surgeon scanned the exposed kidney surface using a stereo laparoscope. The 3D reconstructed model of the kidney surface and the tumor is shown in Fig. 11. This model could further be registered to the diagnostic CT or MRI to plan the extent of surgical resection intraoperatively. This experiments also showed that our method can handle tissue motion caused by respiration, which is because respiration often cause the entire tissue to move but the deformation is relatively minimal. Since the time to scan the tissue surface is short, the tissue motion may not significant.

In the third set of experiments, we obtained intraoperative stereo laparoscopy images from a urethroplasty procedure. Prior to resecting the urethral constriction, the urethra was exposed to identify the extent of the constriction. Thereafter, the authors scanned the exposed surgical area using a stereo laparoscope (Karl Storz Inc., model TipCam 26605AA) by moving the laparoscope slowly along the urethra. The interlaced video was captured and

recorded using a video capture program in OpenCV. Fig. 12 shows the results of the surface mosaicking algorithms of the exposed urethra. The figure shows a high-resolution 3D mosaicked surface model of the urethra and the surrounding structures. The fourth set of experiments were conducted with the same stereo laparoscope and the data was collected during a spine surgery, as shown in Fig. 13. The spine bone was scanned by the Karl Storz stereo laparoscope after it was exposed. The estimated camera trajectories are smooth, which qualitatively prove that our method is accurate.

As shown in Fig. 14, the last in-vivo experiment was conducted on the Hamlyn data[1], which was captured within a porcine abdomen by using a stereo laparoscope. The length of this video is longer (≈ 35s) than other videos, and the smooth camera trajectory shown in Fig. 14 demonstrated that our method is able to work on such relatively long videos.

We also tested ORB-SLAM2 on the collected *in vivo* data, which performed well on most cases because the texture is generally richer than our *ex vivo* data. However, in the spine experiment we observed that ORB-SLAM2 failed to track the camera motion during the scan (see Fig. 13).

Experiments with in-vivo data demonstrated that our approach can be applied to stereo optical videos obtained from different types of imaging modalities, and has potential for the 3D reconstruction of different types of tissues in varying lighting and surgical conditions. The reconstructed surface could be used for further registration to diagnostic or intraprocedural volumetric CT/MRI imaging.

C. Runtime

We report the average runtime of the main steps of the proposed 3D reconstruction method in Tab. III, which is the average results of 1,000 key frames on 960 × 540 laparoscopy videos. The average computational time to process a key frame is 76.3 ms, which suggests that the proposed method is real-time.

V. Conclusions

In this paper, we have proposed a series of algorithms to solve the problem of tissue surface reconstruction, and mainly addressed the difficulties caused by low texture. The main novelties of this paper are as follows: (1) We have proposed effective post-processing steps for the local stereo matching method to enlarge the radius of constraint, and these steps are appropriate for GPU computation. (2) We have combined a histogram voting-based inliers pre-selection method and a novel DynamicR1PP*n*P algorithm that is robust to feature matching outliers to handle the camera motion tracking problem in the SLAM system. Traditional SLAM systems, such as ORB-SLAM, usually utilize RANSAC + P3P methods for camera motion tracking, which cannot work robustly when the number of inliers is too small. The methods proposed in this paper can greatly improve the robustness of the SLAM system. Experimental results on *ex*- and *in vivo* videos captured using different types of imaging modalities have demonstrated the feasibility of our methods, and the obtained

[1] http://hamlyn.doc.ic.ac.uk/vision/data/Dataset1/stereo.avi

models have high quality textures and the same resolution as the input videos. We have also introduced the CUDA implementation details to accelerate the computation with the GPU and enable real-time performance.

One limitation of this work is that we assume a static environment during the scan, hence this method is mainly suitable for surgeries on tissues with minimal deformation, such as the cases in our *in vivo* experiments or other surgeries on bony structures. But such minimal deformation cases are common, which makes our method valuable for practical applications.

Acknowledgment

This work was supported by the National Institute of Biomedical Imaging and Bioengineering of the National Institutes of Health through Grant Numbers R01EB025964, P41EB015898, P41RR019703, and a Research Grant from Siemens-Healthineers USA. We appreciate the generous help of Drs. Jiping Wang, Matthew Ingham, Steven Chang, Jairam Eswara, Carleton Eduardo Corrales, Sarah Frisken and Alexandra Golby in collecting *in vivo* data.

References

[1]. Yang Q, “Stereo matching using tree filtering,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 4, pp. 834–846, 2015. [PubMed: 26353297]

[2]. Hirschmuller H, “Stereo processing by semiglobal matching and mutual information,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 2, pp. 328–341, 2008. [PubMed: 18084062]

[3]. Bleyer M, Rhemann C, and Rother C, “Patchmatch stereo-stereo matching with slanted support windows.” in BMVC, vol. 11, 2011, pp. 1–11.

[4]. Mountney P, Stoyanov D, and Yang G-Z, “Three-dimensional tissue deformation recovery and tracking,” IEEE Signal Processing Magazine, 2010.

[5]. Mur-Artal R, Montiel JMM, and Tardos JD, “Orb-slam: A versatile and accurate monocular slam system,” IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147–1163, 2015.

[6]. Mur-Artal R and Tardós JD, “Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras,” IEEE Transactions on Robotics, vol. 33, no. 5, pp. 1255–1262, 2017.

[7]. Engel J, Schöps T, and Cremers D, “Lsd-slam: Large-scale direct monocular slam,” in ECCV. Springer, 2014, pp. 834–849.

[8]. Puerto-Souza GA and Mariottini G-L, “A fast and accurate feature-matching algorithm for minimally-invasive endoscopic images,” IEEE Transactions on Medical Imaging, vol. 32, no. 7, pp. 1201–1214, 2013. [PubMed: 23335663]

[9]. Zhou H, Zhang T, and Jagadeesan J, “Re-weighting and 1-point ransac-based pnp solution to handle outliers,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018.

[10]. Maier-Hein L, Mountney P, Bartoli A et al., “Optical techniques for 3d surface reconstruction in computer-assisted laparoscopic surgery,” Medical Image Analysis, vol. 17, no. 8, pp. 974–996, 2013. [PubMed: 23837969]

[11]. Hirschmuller H and Scharstein D, “Evaluation of stereo matching costs on images with radiometric differences,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 9, pp. 1582–1599, 2009. [PubMed: 19574620]

[12]. Scharstein D and Szeliski R, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” International Journal of Computer Vision, vol. 47, no. 1–3, pp. 7–42, 2002.

[13]. Birchfield S and Tomasi C, “Depth discontinuities by pixel-to-pixel stereo,” International Journal of Computer Vision, vol. 35, no. 3, pp. 269–293, 1999.

[14]. Boykov Y, Veksler O, and Zabih R, “Fast approximate energy minimization via graph cuts,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 11, pp. 1222–1239, 2001.

[15]. Wang Z-F and Zheng Z-G, “A region based stereo matching algorithm using cooperative optimization,” in CVPR. IEEE, 2008, pp. 1–8.

[16]. Brown MZ, Burschka D, and Hager GD, “Advances in computational stereo,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 8, pp. 993–1008, 2003.

[17]. Di Stefano L, Mattoccia S, and Tombari F, “Zncc-based template matching using bounded partial correlation,” Pattern Recognition Letters, vol. 26, no. 14, pp. 2129–2134, 2005.

[18]. Zabih R and Woodfill J, “Non-parametric local transforms for computing visual correspondence,” in European Conference on Computer Vision. Springer, 1994, pp. 151–158.

[19]. Bhat DN and Nayar SK, “Ordinal measures for image correspondence,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 4, pp. 415–423, 1998.

[20]. Zhang K, Lu J, and Lafruit G, “Cross-based local stereo matching using orthogonal integral images,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 19, no. 7, pp. 1073–1079, 2009.

[21]. Yoon K-J and Kweon IS, “Adaptive support-weight approach for correspondence search,” IEEE Transactions on Pattern Analysis and Machine Intelligence, no. 4, pp. 650–656, 2006. [PubMed: 16566513]

[22]. Stoyanov D, Scarzanella MV, Pratt P, and Yang G-Z, “Real-time stereo reconstruction in robotically assisted minimally invasive surgery,” in MICCAI. Springer, 2010, pp. 275–282.

[23]. Chang P-L, Stoyanov D, Davison AJ et al., “Real-time dense stereo reconstruction using convex optimisation with a cost-volume for image-guided robotic surgery,” in MICCAI. Springer, 2013, pp. 42–49.

[24]. Jin S, Cho J, Dai Pham X et al., “Fpga design and implementation of a real-time stereo vision system,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 20, no. 1, pp. 15–26, 2010.

[25]. Röhl S, Bodenstedt S, Suwelack S et al., “Dense gpu-enhanced surface reconstruction from stereo endoscopic images for intraoperative registration,” Medical Physics, vol. 39, no. 3, pp. 1632–1645, 2012. [PubMed: 22380395]

[26]. Zollhöfer M, Nießner M, Izadi S et al., “Real-time non-rigid reconstruction using an rgb-d camera,” ACM Transactions on Graphics, vol. 33, no. 4, p. 156, 2014.

[27]. Sun D, Liu J, Linte CA et al., “Surface reconstruction from tracked endoscopic video using the structure from motion approach,” in Augmented Reality Environments for MICCAI. Springer, 2013, pp. 127–135.

[28]. Grasa OG, Bernal E, and Casado S, “Visual slam for handheld monocular endoscope,” IEEE Transactions on Medical Imaging, vol. 33, no. 1, pp. 135–146, 2014. [PubMed: 24107925]

[29]. Mountney P and Yang G-Z, “Motion compensated slam for image guided surgery,” in MICCAI. Springer, 2010, pp. 496–504.

[30]. Chen L, Tang W, John NW et al., “Slam-based dense surface reconstruction in monocular minimally invasive surgery and its application to augmented reality,” Computer Methods and Programs in Biomedicine, vol. 158, pp. 135–146, 2018. [PubMed: 29544779]

[31]. Mountney P and Yang G, “Dynamic view expansion for minimally invasive surgery using simultaneous localization and mapping,” in IEEE EMBC. IEEE, 2009, pp. 1184–1187.

[32]. Totz J, Mountney P, Stoyanov D, and Yang G-Z, “Dense surface reconstruction for enhanced navigation in mis,” in MICCAI. Springer, 2011, pp. 89–96.

[33]. Mahmoud N, Collins T, Hostettler A et al., “Live tracking and dense reconstruction for hand-held monocular endoscopy,” IEEE Transactions on Medical Imaging, 2018.

[34]. Lepetit V, Moreno-Noguer F, and Fua P, “Epnp: An accurate o(n) solution to the pnp problem,” International Journal of Computer Vision, vol. 81, no. 2, p. 155, 2009.

[35]. Newcombe RA, Izadi S, and Hilliges O, “Kinectfusion: Real-time dense surface mapping and tracking,” in ISMAR. IEEE, 2011, pp. 127–136.

[36]. Vollmer J, Mencl R, and Mueller H, “Improved laplacian smoothing of noisy surface meshes,” in Computer Graphics Forum, vol. 18, no. 3. Wiley Online Library, 1999, pp. 131–138.

[37]. Curless B and Levoy M, “A volumetric method for building complex models from range images,” in Computer Graphics and Interactive Techniques. ACM, 1996, pp. 303–312.

[38]. Rublee E, Rabaud V, Konolige K, and Bradski G, “Orb: An efficient alternative to sift or surf,” in ICCV. IEEE, 2011, pp. 2564–2571.

[39]. Bay H, Tuytelaars T, and Van Gool L, “Surf: Speeded up robust features,” in ECCV. Springer, 2006, pp. 404–417.

[40]. Kneip L, Scaramuzza D, and Siegwart R, “A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation,” 2011.

[41]. Triggs B, McLauchlan PF, Hartley RI, and Fitzgibbon AW, “Bundle adjustmenta modern synthesis,” in International Workshop on Vision Algorithms. Springer, 1999, pp. 298–372.

[42]. Pieper S, Halle M, and Kikinis R, “3d slicer,” in Biomedical Imaging: Nano to Macro, 2004. IEEE International Symposium on. IEEE, 2004, pp. 632–635.

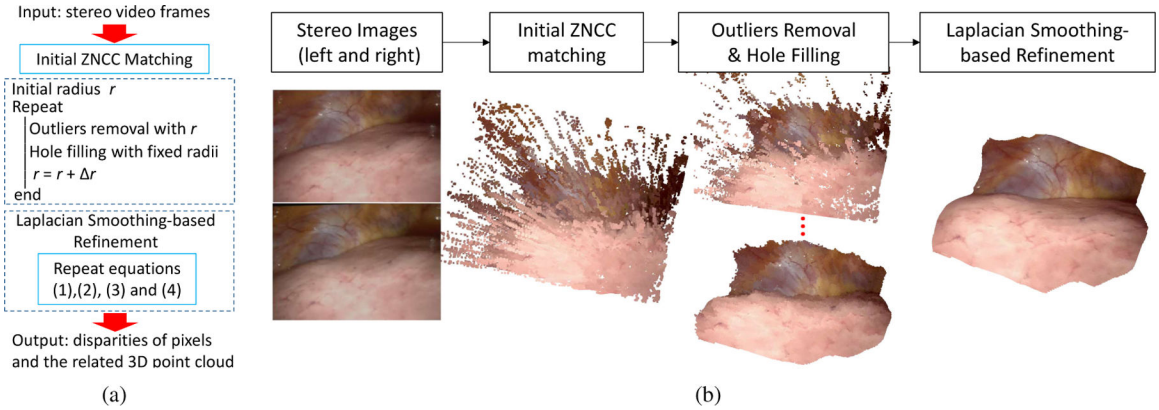


Fig. 1.

(a) The flow chart of our stereo matching method. (b) An intuitive example to show the stereo matching process with a pair of stereo laparoscopic images captured during a lung surgery at our hospital, the texture on the tissue surface is low.

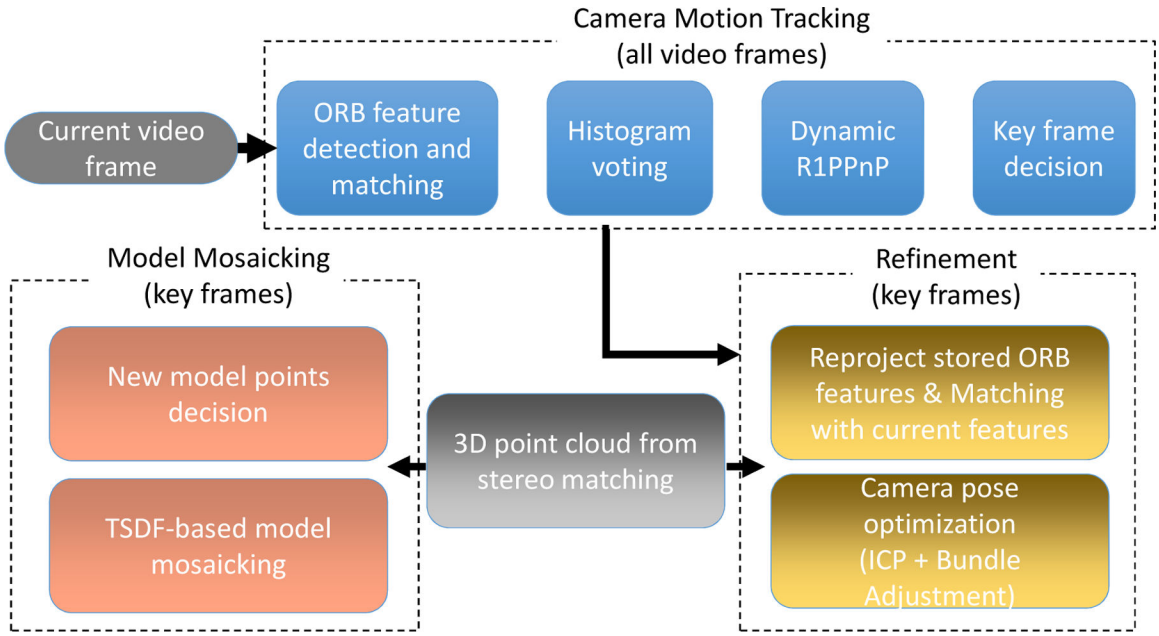


Fig. 2.
The flow chart of the SLAM method.

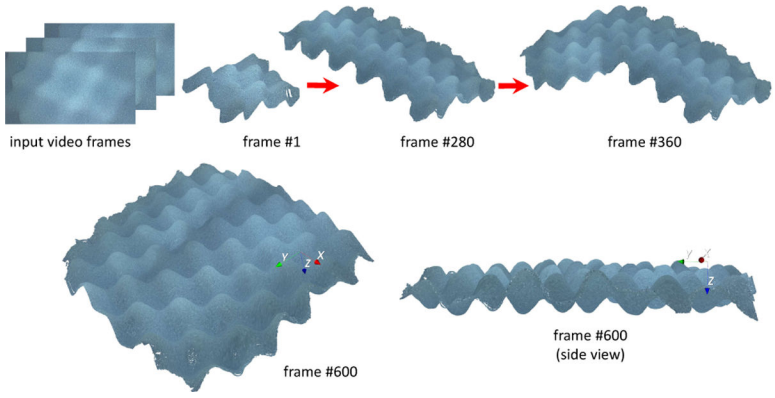


Fig. 3.
An example of our model mosaicking process with a phantom.

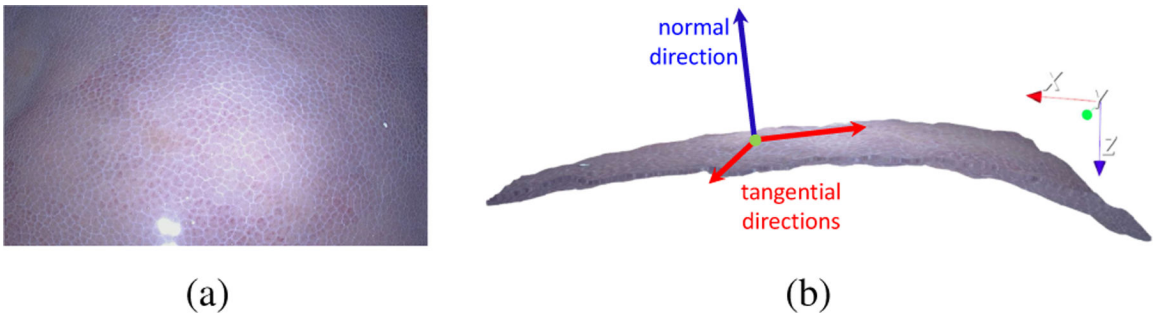


Fig. 4.

ICP-based model alignment may not work well due to the smooth tissue surface and the narrow field of view. We use the laparoscopies images of the liver surface as an example. To observe the texture clearly, the laparoscope should be close to the liver. (a) The obtained image has a narrow field of view. (b) The reconstructed 3D point cloud is small and smooth. Hence ICP-based alignment cannot find good constraints in the tangential directions, but is accurate in the normal direction.

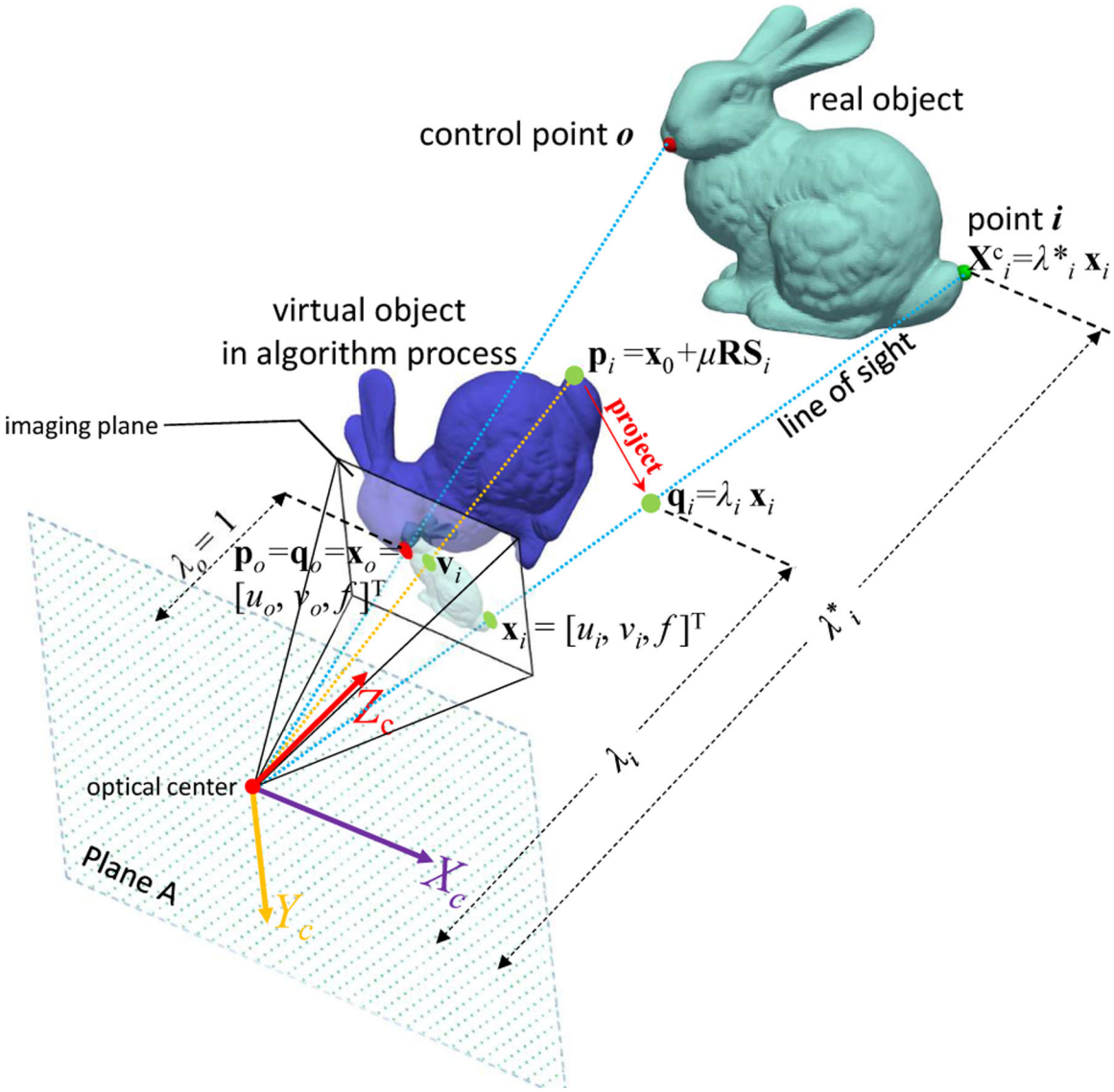


Fig. 5.

Demonstration of the geometrical relationships in the R1PP*n*P algorithm with a bunny model. The mouth point is used as the control point *o* and the tail point is used to exemplify the geometrical relationships.

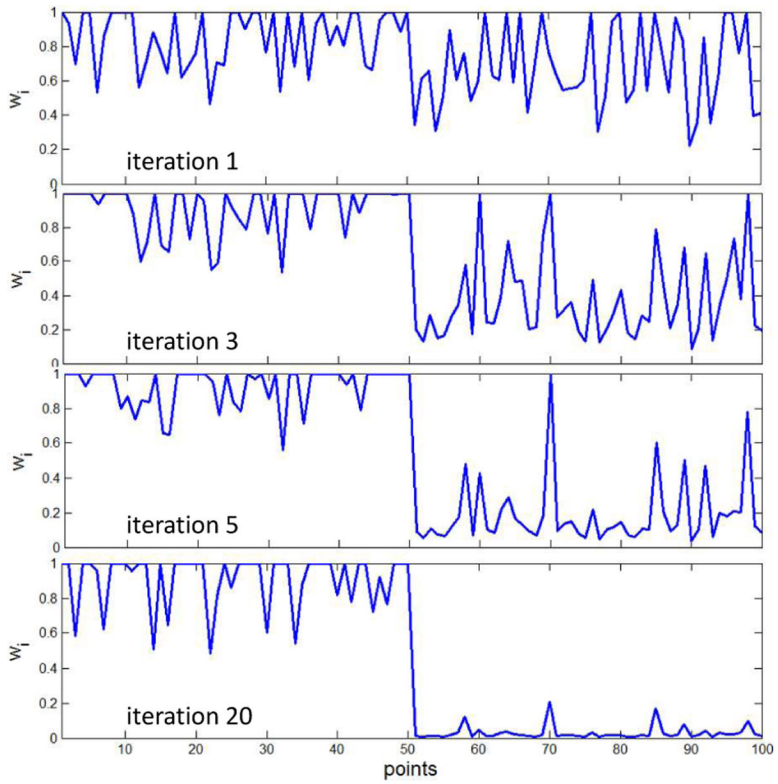
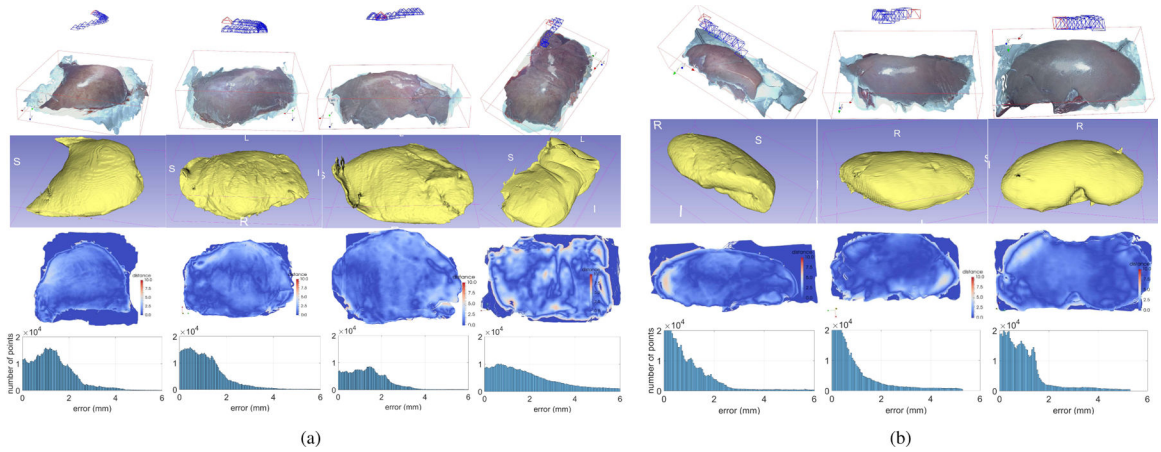


Fig. 6.

An example of the changes of weights w_i in the R1PPnP iterations. In this example, the first 50 matches are inliers and the others are outliers. With the iteration, the weights of outliers decreases and their effects on camera motion estimation are reduced.



(a) (b)

Fig. 7.

Quantified accuracy evaluation on stereo laparoscopy videos of ex-vivo porcine tissues. The 3D reconstruction results were compared with the CT segmentation results. The first row shows the 3D reconstruction results, the second row shows the CT image segmentation results, the third row shows the distance map after registration, and the last row shows the histogram of the errors. (a) Porcine livers, the RMSEs are 1.3, 1.1, 1.4 and 2.0 mm respectively. (b) Porcine kidneys, the RMSEs are 1.0, 1.0 and 1.1 mm respectively.

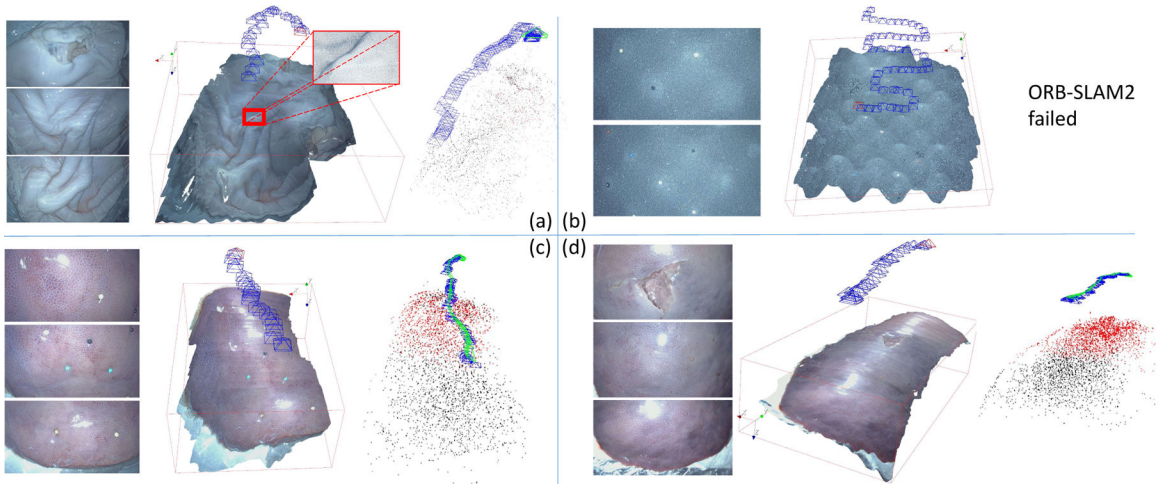


Fig. 8.

Qualitative results on stereo laparoscopy videos of phantoms and *ex vivo* porcine tissues. The reconstructed tissues and the estimated camera motion (blue triangles) at key frames are shown in this figure. (a) A porcine stomach. (b) A phantom. (c)-(d) Porcine livers. A small region of the reconstructed model in (a) is enlarged to demonstrate the dense point cloud. For each case, from left to right are image samples (only images from the left camera are shown but both left and right images are used in our method), the reconstruction results of our method and the results of ORB-SLAM2. ORB-SLAM2 tracking failure occurred in cases in (b) and (d) due to the low texture.

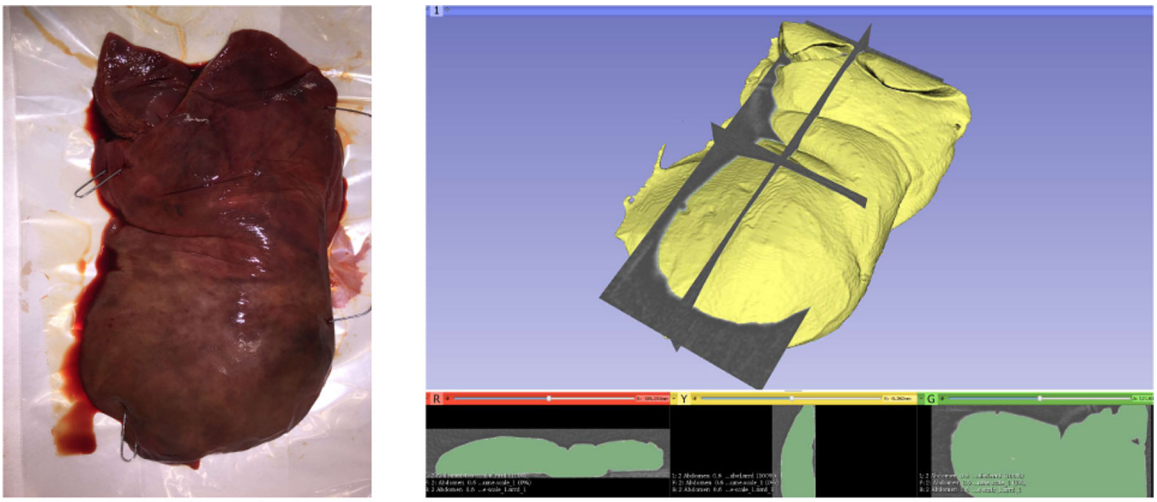


Fig. 9.

We used CT imaging of ex-vivo porcine tissues for quantified evaluation. (a) An ex-vivo porcine liver. (b) 3D Slicer-based segmentation of the obtained CT imaging.

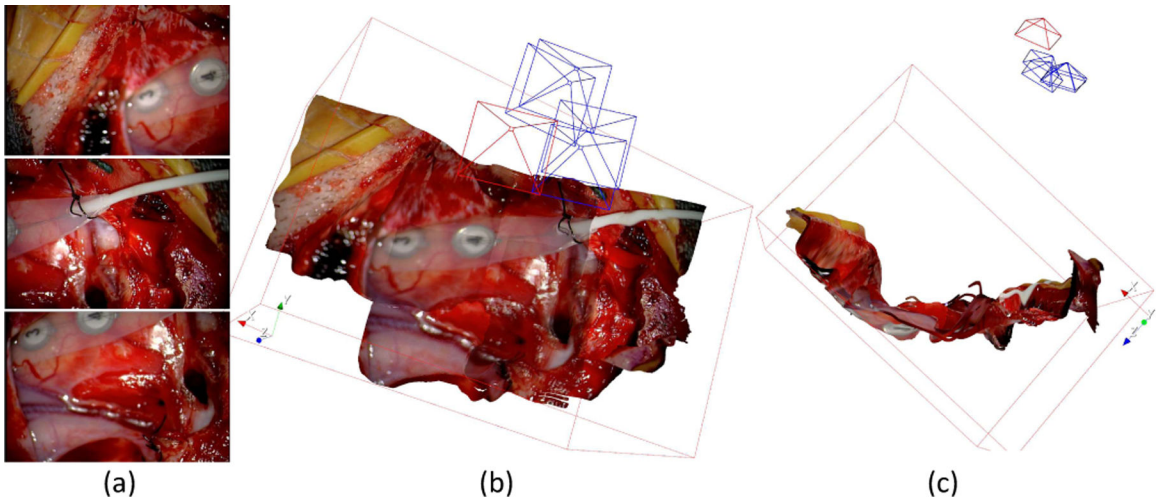


Fig. 10.

Experiments on stereo microscopic images captured during a neurosurgery at our hospital. (a) Samples of input images, and only images from the left camera are shown. (b)-(c) Our results.

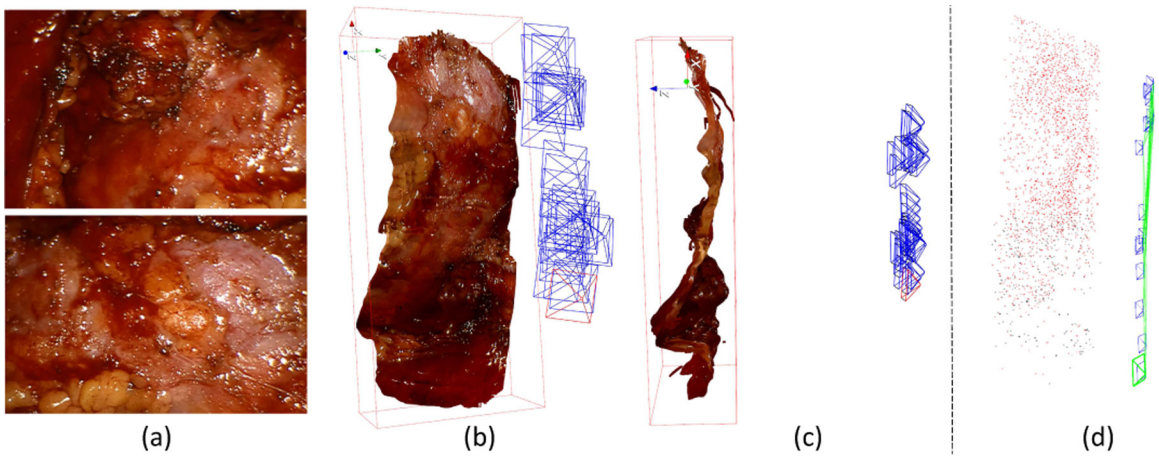


Fig. 11.

Experiments on stereo laparoscopy videos captured during a robotic kidney surgery at our hospital. The kidney surface and the tumor are shown in the images. (a) Samples of input left camera images. (b)-(c) Our results. (d) ORB-SLAM2 results. Due to respiration, the camera motion with respect to the kidney is more complex.

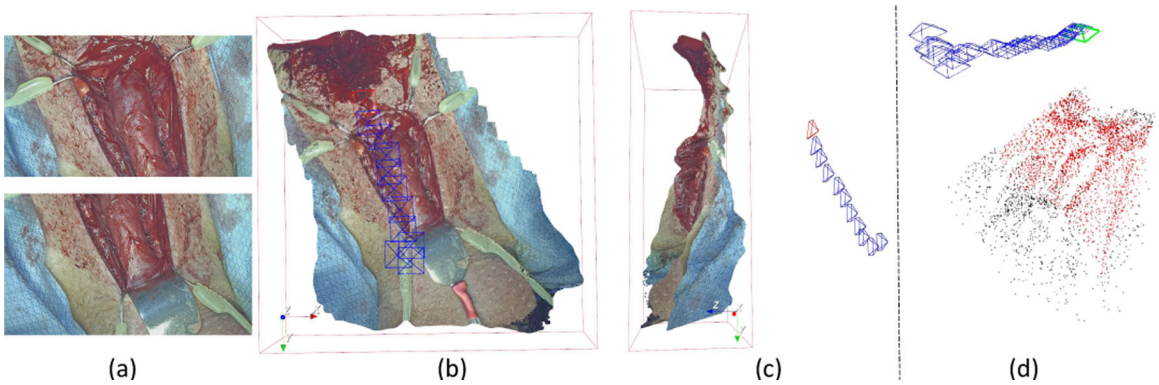


Fig. 12.

Reconstruction results of the urethra. We scanned the structures with the KARL STORZ stereo laparoscope during the surgery. (a) Samples of input left camera images. (b)-(c) Our results. (d) ORB-SLAM2 results.

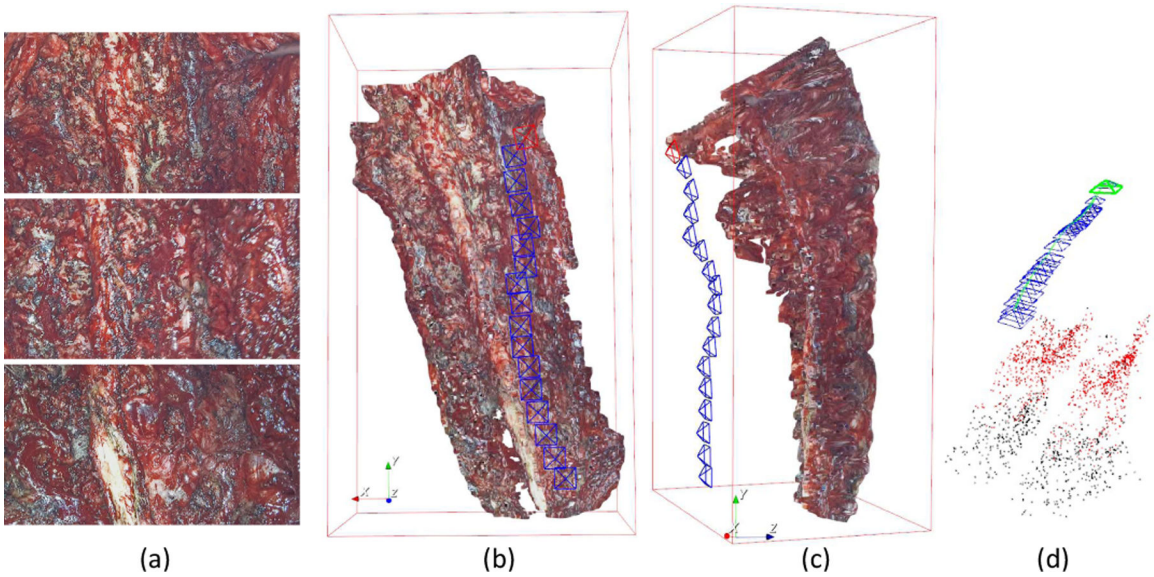


Fig. 13.

Reconstruction results of the spine. We scanned the structures with the KARL STORZ stereo laparoscope during the surgery. (a) Samples of input left camera images. (b)-(c) Our results. (d) ORB-SLAM2 results (tracking failed).

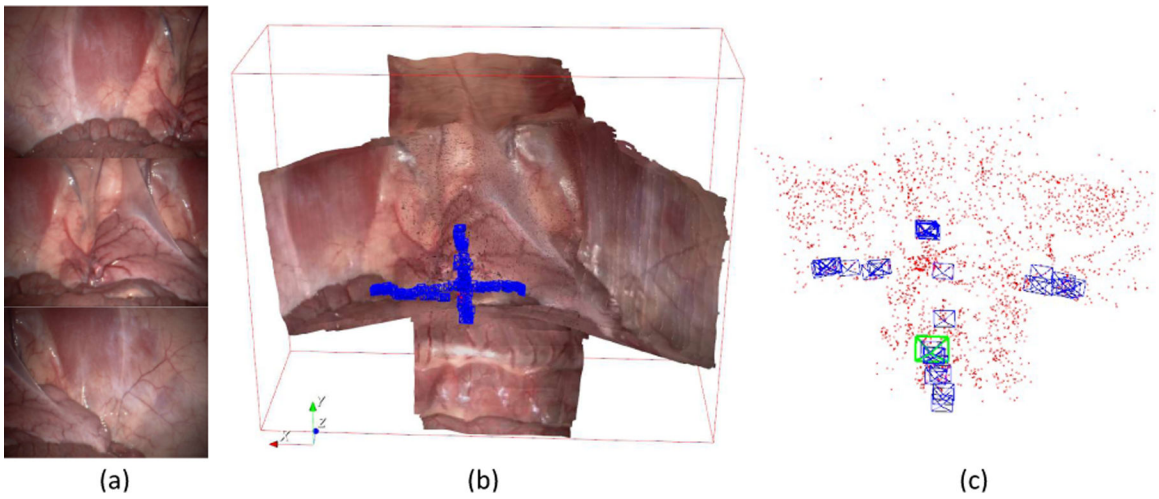


Fig. 14.

Experiments on in-vivo porcine abdomen videos from the Hamlyn datasets. (a) Samples of input left camera images. (b) Our results. (c) ORB-SLAM2 results.

Table I

Parameters of Qualitative Experiments

	exvivo stomach (Fig.8(a))	**exvivo phantom (Fig.8(b))**	**exvivo liver1 (Fig.8(c))**	**exvivo liver2 (Fig.8(d))**	**invivo neurosurgery (Fig.10)**	**invivo kidney (Fig.11)**	**invivo urethral (Fig.12)**	**invivo spine (Fig.13)**	**invivo Hamlyn (Fig.14)**
video length (s)	19.5	48.3	15.5	28.7	-	3.4	12.4	6.1	35
number of frames	293	725	232	430	5	86	186	91	961
resolution (pixels)	960×540	960×540	960×540	960×540	720×480	1024×768	960×540	960×540	320×240
average tissue-camera distance (mm)	87.3	129	67.2	71.6	76.0	92.0	97.7	95.3	49.9
average speed (mm/s)	10.7	10.0	6.2	4.4	-	17.2	6.4	27.3	3.2
bounding box length (mm)	199	274	137	164	80	153	161	252	125
number of model points ($\times 10^6$)	2.1	2.6	1.9	1.8	0.8	1.1	1.2	2.2	0.2
key frames threshold (Eq. (17)):	10.0	10.0	10.0	10.0	1e-6	5.0	10.0	10.0	3.0

IEEE Trans Med Imaging. Author manuscript; available in PMC 2021 February 01.

Table II

Parameters of Quantitative Experiments (Fig.7)

	liver1	liver2	liver3	liver4	kidney1	kidney2	kidney3
video length (s)	25.2	14.3	10.9	15.7	7.3	6.2	4.3
number of frames	378	215	164	236	109	93	64
resolution (pixels)	960×540	960×540	960×540	960×540	960×540	960×540	960×540
average tissue-camera distance (mm)	107.9	82.7	77.1	173.5	84.5	85.9	87.8
average speed (mm/s)	3.5	8.5	6.4	4.7	8.6	5.2	6.9
bounding box length (mm)	163	136	121	249	181	137	134
number of model points ($\times 10^6$)	1.2	1.2	0.9	0.7	0.7	0.7	0.7
key frames threshold (Eq.(17)):	10.0	10.0	10.0	10.0	10.0	10.0	10.0

Table III

Average Runtime of 3D Reconstruction(ms)

video reading	12.7
stereo matching	15.9
ORB matching and histogram voting	15.2
DynamicR1PPnP	6.1
Refinement	24.6
TSDF	2.2
Total	**76.3**