MIS-SLAM: Real-time Large Scale Dense Deformable SLAM System in Minimal Invasive Surgery Based on Heterogeneous Computing

Jingwei Song, Jun Wang, Liang Zhao, Shoudong Huang and Gamini Dissanayake[*][†]

February 5, 2019

Abstract

Real-time simultaneously localization and dense mapping is very helpful for providing Virtual Reality and Augmented Reality for surgeons or even surgical robots. In this paper, we propose MIS-SLAM: a complete real-time large scale dense deformable SLAM system with stereoscope in Minimal Invasive Surgery (MIS) based on heterogeneous computing by making full use of CPU and GPU. Idled CPU is used to perform ORB-SLAM for providing robust global pose. Strategies are taken to integrate modules from CPU and GPU. We solved the key problem raised in previous work, that is, fast movement of scope and blurry images make the scope tracking fail. Benefiting from improved localization, MIS-SLAM can achieve large scale scope localizing and dense mapping in real-time. It transforms and deforms current model and incrementally fuses new observation while keeping vivid texture. In-vivo experiments conducted on publicly available datasets presented in the form of videos demonstrate the feasibility and practicality of MIS-SLAM for potential clinical purpose.

1 INTRODUCTION

Comparing with open surgery, Minimally Invasive Surgery (MIS) brings shortcomings such as lack of field of view, poor localization of scope and fewer surrounding information. Limited by these, surgeons are required to perform the intra-operations in narrow space with elongated tools and without direct 3D vision [1]. To overcome these problems, surgeons spend large amount of time training to be familiar with doing operations under scopes.

[*] All the authors are from Centre for Autonomous Systems, University of Technology, Sydney, P.O. Box 123, Broadway, NSW 2007, Australia

[†] Email: jingwei.song@student.uts.edu.au, wangjun@radi.ac.cn, {Liang.Zhao; Shoudong.Huang; Gamini.Dissanayake}@uts.edu.au

SLAM (simultaneous localization and mapping) is a technique applied in robotics for pose estimation and environment mapping. Efforts have been devoted to exploit the feasibility of applying SLAM to localize the scope and reconstruct a sparse or even dense soft-tissue surface. [2] and [3] adopt conventional feature based SLAM, these are extended Kalman filter (EKF) SLAM and Parallel Tracking and Mapping (PTAM). They improved EKF and PTAM by using threshold strategies to separate rigid and non-rigid feature points. [4][5][6] exploit and tune a complete and widely used large scale SLAM system named ORB-SLAM [7]. They analyze and prove that ORB-SLAM is also suitable for scope localization in MIS. In [5], a quasi-dense map is generated off-line based on pose imported from ORB-SLAM. Many researchers adopt feature points for localization and sparse mapping. Contrary to feature based SLAM, Du et al. [8] adopts dense matching SLAM which employed a special optical flow namely Deformable Lucas-Kanade for tracking tissue surface. Aside from SLAM, other approaches also contribute greatly to enable augmented reality (AR) and virtual reality (VR) in MIS. [9] proposes an approach to recover 3D geometry from stereo images. A structure from motion pipeline [10] is proposed for partial 3D surgical scene reconstruction and localization. [11] and [12] extract whole tissue surface from stereo or monocular images. All these approaches contribute significantly to MIS. However, they still don't provide a real-time complete and robust solution for localizing scope while reconstructing dense deformable soft-tissue surfaces. All the SLAM techniques mentioned above focus on monocular scope and fail to solve the problem of missing scale, thus making localization not practical.

To broaden surgeons' field of view, 3D laparoscopy, or binoculars is applied to generate two images from different viewing point so that a 3D geometry based on parallax is created in surgeons' mind for better understanding of the environment. Recently, similar stereo vision is adopted by some AR devices for enhancing MIS procedures. Therefore, it will be very helpful if stereo vision related approaches in computer vision community could be integrated, extended and improved to recover deformable shape in real-time while estimating the pose of the camera. In our previous work, we proposed dynamic reconstruction system of deformable soft-tissue with stereo scope [13]. A warping field based on the embedded deformation nodes approach is introduced with 3D dense shapes recovered from stereo images. With the help of general-purpose computing on graphics processing units (GPGPU), all the processes are achieved in real-time. Mentioned in [13], the first and most important challenge to the pipeline is the fast movement of scope. Fast movement not only makes visual odometry unstable but also causes blurry images and worse registrations. This issue has also been reported in [2] and [3].

Inspired by the researches [4] [5] [6] which demonstrate the robustness of camera pose estimation from ORB-SLAM, we figure out ORB-SLAM is suitable to be improved and coupled with dense deformable SLAM. In this paper, we propose MIS-SLAM based on our preliminary work [13] with the following major improvements: (1) We proposed a heterogeneous framework to make full use of both GPU (dense deformable SLAM) and CPU (ORB-SLAM) to recover the dense deformed 3D structure of the soft-tissues in MIS scenario. Computational power of CPU is fully exploited to run an improved ORB-SLAM to provide complementary information to GPU modules. (2) Modules from GPU and CPU are deeply integrated to boost performance

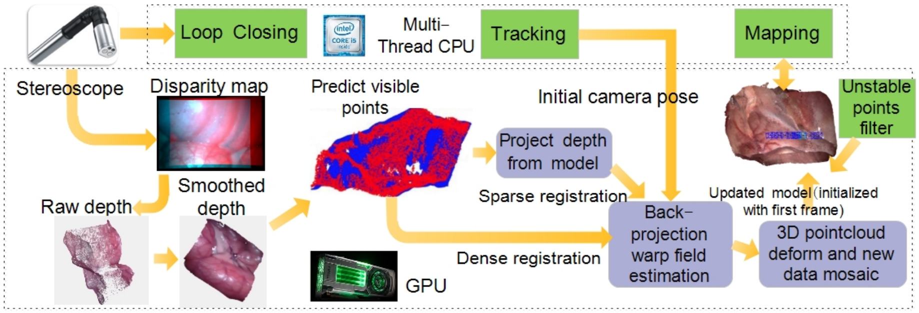

Figure 1: The framework of MIS-SLAM. CPU is responsible for ORB-SLAM, uploading features, global pose and start a visualization module. GPU processes depth estimation, registration, fusion and visualization.

and enhance the efficiency. Sparse ORB features as well as global pose are uploaded to GPU. (3) We upgrade former model point storage system and fusion management strategy to enhance large scale soft-tissue reconstructing. Comparing with truncated signed distance function (TSDF) widely used in computer vision community [14] [15] and [16], point cloud management in MIS-SLAM notably reduces memory as well as boosts the performance. (4) Real-time visualization is achieved on GPU end. MIS-SLAM can process large scale surface reconstruction in just one desktop in real-time. We suggest readers to view the associate video to fully appreciate the live capabilities of the system.

2 Technical Details

2.1 Overview of MIS-SLAM

The architecture we adopt can be divided as **initial tracking** and **deformable tracking and dense mapping**. The initial tracking is achieved with an improved ORB-SLAM algorithm on CPU end. Deformable tracking and dense mapping is implemented on GPU end.

In the initial tracking step, ORB-SLAM is first launched on CPU; ORB features and global pose are uploaded from CPU to GPU global memory. This initial global pose significantly increases robustness of the system.

In the deformable tracking and dense mapping step, after receiving initial global pose from CPU end, it first initializes the model with the first estimated depth. Each time when a new observation is acquired, the matched ORB features are uploaded to GPU. Potential visible points are extracted from the model and projected on 2D depth images. A registration process is performed to estimate optimum global pose as well as non-rigid warping field. Live model is then deformed to current shape according to this transformation and fused with the new observation. We make use of the feature called ‘Graphic Interoperability’ in Compute Unified Device Architecture (CUDA) to directly visualize model from GPU side. Fig. 1 demonstrates the pipeline of these processes.

Realizing the point cloud generated from stereo images are much less reliable than depth perception sensors, we modify and update our previous approach of generating point cloud [13] with more properties. Each point stores six domains: coordinate $\boldsymbol{v}_i$, normal $\boldsymbol{n}_i$, weight ω_i, color $\boldsymbol{C_i}$, time stamp t_i and a boolean variable stability $\boldsymbol{S_i}$. We update original visible points selection approach to have better model to depth registration (Algorithm 1). We add t_i and $\boldsymbol{S_i}$ and introduce model filtering technique to have smooth model with less noisy points (Algorithm 2 and 3).

2.2 Depth estimation from stereo images

Efficient Large-scale Stereo (ELAS) [17] is adopted as the depth estimation method. ELAS has been widely proved to achieve good result in surgical vision [18]. Fig. 2 shows the example of original depth and smoothed depth.

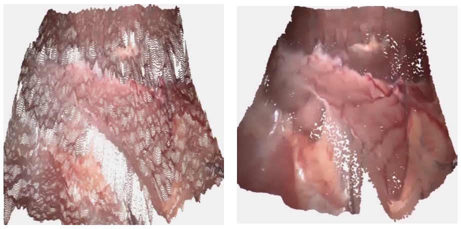
Original depth
Smoothed depth

Figure 2: Examples of depth and smoothed depth.

2.3 Sparse key correspondences and camera pose estimation

The main issue in previous work [13] is the inaccuracy of global scope pose leading to instability of the pipeline. The deformation graph based approach is a typical model-to-frame visual odometry process lacking additional mechanics to ensure global pose tracking robustness. Without good camera pose initialization, dense mapping inevitably suffers from drift or lost tracking. To improve the robustness of the system, idled CPU is fully exploited to run ORB-SLAM for providing good initial pose for enhancing robustness. ORB-SLAM module provides the ORB features which are fully exploited on GPU. This strategy save computational powers on GPU: (1) Dense Speeded Up Robust Feature (SURF) extraction and matching step in original approach [13] is therefore not needed as we upload matched ORB features. (2) Visual Odometry and Random sample consensus (RANSAC) on GPU end in [13] is replaced with initial pose and ORB features from ORB-SLAM on CPU end.

2.4 Deformation

The basic idea of deformation graph is weighted average of locally rigid rotation and transformation defined by neighboring deformation nodes, which are sparsely and evenly scattered in space. Each source point is transformed to its target position by several nearest embedded deformation (ED) nodes which are defined by position $\boldsymbol{g_j} \in \mathbb{R}^3$, affine matrix $\boldsymbol{A_j} \in \mathbb{R}^{3\times 3}$ and translation vector in $\boldsymbol{t_j} \in \mathbb{R}^3$. Practically, we downsampled the reconstructed model to get nodes and initialize $\boldsymbol{A_j}$ with identity matrix and $\boldsymbol{t_j}$ with zero vector. We would like to address nodes are used for describing deformation and is irrelevant of model. For any given vertex $\boldsymbol{v}_i$, deformed position $\tilde{\boldsymbol{v}}_{\boldsymbol{i}}$ is defined by the ED nodes as:

$$\tilde{\boldsymbol{v}}_{\boldsymbol{i}} = \boldsymbol{R}\sum_{j=1}^{k} w_j(\boldsymbol{v_i})[\boldsymbol{A_j}(\boldsymbol{v_i}-\boldsymbol{g_j})+\boldsymbol{g_j}+\boldsymbol{t_j}]+\boldsymbol{T} \tag{1}$$

where k denotes the number of neighboring node. $w_j(\boldsymbol{v_i})$ is quantified weight for transforming $\boldsymbol{v_i}$ exerted by each related ED node. $\boldsymbol{R}$ and $\boldsymbol{T}$ denote rigid rotation and translation. We confine the number of nearest nodes by defining the weight in Eq. 2. Deformation of each point in the space is limited locally by setting the weight as:

$$w_j(\boldsymbol{v_i}) = (1-||\boldsymbol{v_i}-\boldsymbol{g_j}||/d_{max}) \tag{2}$$

where d_{max} is the maximum distance of the vertex to $k+1$ nearest ED node. Please note that all the poses are in the coordinate of the first frame.

2.5 Energy function

Following our previous formulation [13], two new terms are added to ensure robustness of global pose. The objective function formulated is composed of six terms: Rotation, Regularization, the point-to-plane distances between the visible points and the target scan, sparse key points correspondence and global pose (new terms) as:

$$\begin{aligned}\underset{\boldsymbol{R},\boldsymbol{T},\boldsymbol{A_1},\boldsymbol{t_1}...\boldsymbol{A_m},\boldsymbol{t_m}}{\operatorname{argmin}} \quad & w_{rot}E_{rot}+w_{reg}E_{reg}+w_{data}E_{data}+w_{corr}E_{corr} \\ & +w_rE_r+w_pE_p\end{aligned} \tag{3}$$

where m is the number of ED nodes. We follow [19] to constrain deformation graph from unreasonable deformation with two constraints **Rotation** and **Regularization**. All m nodes follows the two constraints.

Rotation. E_{rot} sums the rotation error of all the matrix in the following form:

$$E_{rot} = \sum_{j=1}^{m} Rot(\boldsymbol{A_j}) \tag{4}$$

$$\begin{aligned}Rot(\boldsymbol{A_j}) = (\boldsymbol{c_1}\cdot\boldsymbol{c_2})^2+(\boldsymbol{c_1}\cdot\boldsymbol{c_3})^2+(\boldsymbol{c_2}\cdot\boldsymbol{c_3})^2+ \\ (\boldsymbol{c_1}\cdot\boldsymbol{c_1}-1)^2+(\boldsymbol{c_2}\cdot\boldsymbol{c_2}-1)^2+(\boldsymbol{c_3}\cdot\boldsymbol{c_3}-1)^2\end{aligned} \tag{5}$$

where $\boldsymbol{c_1}$, $\boldsymbol{c_2}$ and $\boldsymbol{c_3}$ are the column vectors of the affine matrix $\boldsymbol{A_j}$.

Regularization. This term is to prevent divergence of the neighboring nodes exerts on the overlapping space. For details, please refer to [13].

$$E_{reg} = \sum_{i=1}^{m} \sum_{j=1}^{k} \alpha_{ij} ||\boldsymbol{A_j}(\boldsymbol{g_i} - \boldsymbol{g_j}) + \boldsymbol{g_j} + \boldsymbol{t_j} - (\boldsymbol{g_i} + \boldsymbol{t_i})||^2 \tag{6}$$

where α_{ij} is the weight calculated by the Euclidean distance of the two ED nodes. We follow [19] by uniformly setting α_{ij} to 1.

Data Term. We follow Algorithm 1 to find registrations of model points and minimize point to plane distance of all the registered points. For each model point $\boldsymbol{v_i}$, if it is registered to depth, it is assumed to be a visible point. In Algorithm 1, ε_d and ε_n are thresholds for measuring distance and angle. $P(\cdot)$ is projecting 3D point to 2D pixel, $\Gamma(\cdot)$ is lifting 2D pixel into 3D space, $H(\cdot)$ is converting 2D pixel from depth into 3D normal.

After extracting registered visible points, we adopt back-projection approach as a model-to-scan registration strategy that penalizes misalignment of the predicted visible points $\boldsymbol{v}_i$ $(i \in \{1,...,N\})$ and current depth scan $\mathbb{D}$. Data Term is sum of point-to-plane errors in the form of:

$$E_{data} = \sum_{i=1}^{N} (\boldsymbol{H}(P(\tilde{\boldsymbol{v}}_i))^T (\tilde{\boldsymbol{v}}_{\boldsymbol{i}} - \Gamma(P(\tilde{\boldsymbol{v}}_i)))^2 \tag{7}$$

Point-to-plane distance is calculated by multiplying corresponding normal to the pixel in depth $\mathbb{D}(\cdot)$ with normal $H(\cdot)$. $\tilde{\boldsymbol{v}}_{\boldsymbol{i}}$ is the deformed position of point $\boldsymbol{v}_i$.

As described in [16], back-projection and point-to-plane strategies make full use of the input depth image so that the Jacobians can be calculated in regularized 2D space which leads to fast convergence and robustness to outliers.

Correspondence. Similar to previous work [13], we also utilize RGB information for enhancing robustness. We first track frame-to-frame feature points and minimize the Euclidean distance between pair-wise sparse key points generated from features described in Section 2.3 in the following form. We substitute previous Dense SURF with ORB features uploaded from ORB-SLAM.

$$E_{corr} = ||\tilde{\boldsymbol{V}}_{\boldsymbol{i}} - \boldsymbol{V_i}|| \tag{8}$$

where $\tilde{\boldsymbol{V}}_i$ and $\boldsymbol{V}_i$ are the 3D points of current frame and deformed points from last frame of ORB features.

Global Pose. We add this new term with regard to previous formulation [13]. It is measured by the variations of rotation and transformation. First frame is fixed as the coordinate origin. We use Euclidean distance and Euler angles to define the difference between optimized global pose (orientation $\tilde{\boldsymbol{R}}_i$ and position $\tilde{\boldsymbol{P}}_i$) and global pose (orientation $\boldsymbol{R}_i$ and position $\boldsymbol{P}_i$) generated by ORB-SLAM. It is presented in the following form:

$$E_r = ||\tilde{\boldsymbol{R}}_{\boldsymbol{i}} - \boldsymbol{R_i}|| \qquad E_p = ||\tilde{\boldsymbol{P}}_{\boldsymbol{i}} - \boldsymbol{P_i}|| \tag{9}$$

2.6 Optimization

We adopt Algorithm 1 to find visible point set $\mathbb{V}$ for optimization. We follow our previous strategy [13] using Levenberg-Marquardt (LM) to solve the nonlinear optimization problem. The efficiency is almost the same as [13] because only 6 more variables (Global orientation and translation) are added.

Algorithm 1: Model points to depth image registration

```
Input: Point cloud state in last frame (position $\boldsymbol{v}_i$ normal $\boldsymbol{n}_i$)
       Depth map in current state $\mathbb{D}_n$
       Distance threshold of two points $\varepsilon_d$
       Normal angle threshold of two normals $\varepsilon_n$
Output: Visible points set $\mathbb{V}_n$ regarding to depth $\mathbb{D}_n$
foreach Model point $\boldsymbol{v_i}$ do
    if $\mathbb{D}(P(\boldsymbol{v}_i)) \neq null$ then
        if ($\|\boldsymbol{v}_i - \Gamma(P(\boldsymbol{v}_i))\| < \varepsilon_d$ and $\boldsymbol{n_i} \cdot \boldsymbol{H}(P(\boldsymbol{v}_i)) > cos(\varepsilon_n)$)
         then
            Add $\boldsymbol{v}_i$ to $\mathbb{V}_n$
        end
    end
end
```

2.7 Model update with new observation

Inspired by [20], we add new properties (time step and stability) to point management. We fuse model with depth following Algorithm 2. After that we follow Algorithm 3 to remove noisy model points.

The basic idea of Algorithm 2 is building three different groups of point cloud. The original model is classified into registered (Group 1) and unregistered (Group 2) with regard to depth image. Points in Group 1 are fused with depth image. After which pixel from depth image that's not registered with model points are lifted and initialized as new observations (Group 3). All three groups are merged and form the new model.

In Algorithm 3, we apply 'stability $\boldsymbol{S_i}$' to filter model points influenced by noisy depth. Unstable model point is defined as point with low weight (only seen in few times) which has not been observed for several recent frames. This point is likely a noisy point resulting from inaccurate depth estimation. Please refer to Algorithm 3 for how to filter points.

For a single point $\boldsymbol{v}_i$ in nth step, fusion with new depth is achieved by:

$$\tilde{\boldsymbol{v}}_{n+1}|_z = \frac{\tilde{\boldsymbol{v}}_n|_z * \omega_n + \mathbb{D}_{n+1}(P(\tilde{\boldsymbol{v}}_n))}{\omega_n + 1} \tag{10}$$

$$\boldsymbol{C}_{n+1} = \frac{\boldsymbol{C}_n * \omega_n + \mathbb{C}_{n+1}(P(\tilde{\boldsymbol{v}}_n))}{\omega_n + 1} \tag{11}$$

$$\tilde{\boldsymbol{n}}_{n+1} = \frac{\tilde{\boldsymbol{n}}_n \omega_n + \mathbb{N}_{n+1}(P(\tilde{\boldsymbol{v}}_n))}{\omega_n + 1} \tag{12}$$

$$\omega_{n+1} = min(\omega_n + 1, \omega_{max}) \tag{13}$$

Algorithm 2: Fusion of Point cloud with depth image

```
Input: Model $\mathbb{P}_{n-1}$ in last frame and current depth $\mathbb{D}_n$
        Distance and normal thresholds $\varepsilon_d$ and $\varepsilon_n$
Output: Fused model set $\mathbb{P}_n$
Step 1: Register and fuse model with depth (Group 1), the rest model are
  unregistered points (Group 2)
foreach $\boldsymbol{p}_k \in \mathbb{P}_{n-1}$ do
    Deform $\boldsymbol{p}_k$ to $\tilde{\boldsymbol{p}}_k$
    if $\mathbb{D}(P(\boldsymbol{v}_i)) \neq null\ and$
      $\|\tilde{\boldsymbol{p}}_k - \Gamma(P(\tilde{\boldsymbol{p}}_k))\| < \varepsilon_d\ and$
      $\boldsymbol{n_i} \cdot \boldsymbol{H}(P(\tilde{\boldsymbol{p}}_k)) > cos(\varepsilon_n)$ then
        Fuse $\tilde{\boldsymbol{p}}_k$ following Eq. 10, 11, 12 and 13.
        Push fused $\tilde{\boldsymbol{p}}_k$ Group 1
    else
        Push $\tilde{\boldsymbol{p}}_k$ to Group 2
    end
end
Step 2: Add newly observed points (Group 3)
foreach $u_k \in \mathbb{D}_n$ do
    if $u_k$ is not fused in Step 1 then
        Lift $u_k$ into 3D space (position ($\boldsymbol{v}_i$), normal($\boldsymbol{n}_i$), color $\boldsymbol{C_i}$
        Initialize color, $\omega_i = 1$, time stamp $t_i = i + 1$, stability $\boldsymbol{S_i}$= False. and
          pushed into Group 3
    end
end
Step 3: Fuse different types of points
Merge Group 1 Group 2 Group 3 to new model $\mathbb{P}_n$.
```

Algorithm 3: Removing noisy unstable model points

```
Input: Fused model set $\mathbb{P}_n$
        Time and weight thresholds $\tau_{time}$ and $\tau_{weight}$
Output: Filtered model set $\mathbb{P}^{'}_{n}$
        New node positions $\boldsymbol{g}$
foreach $p_k \in \mathbb{P}_{i+1}$ do
    if $t_k < (i - \tau_{time})\ and\ \omega_k < \tau_{weight}\ and\ \boldsymbol{S_i}= False$ then
        Delete $p_k$
    else
        Stamp $t_k = i + 1$
        if $t_k \geq (i - \tau_{time})\ and\ \omega_k \geq \tau_{weight}$ then
            $\boldsymbol{S_i}$= True
        end
    end
end
Regenerate new nodes $\boldsymbol{g}$ and initialize rotation $\boldsymbol{A}$ as identity matrix and
  translation $\boldsymbol{t}$ as zero vector.
```

where $\tilde{\boldsymbol{v}}_n|_z$ is the value of deformed point $\tilde{\boldsymbol{v}}_n$ on the z direction. $\tilde{\boldsymbol{n}}_n$ is deformed normal $\boldsymbol{n}_n$. $\mathbb{D}_n$, $\mathbb{C}_n$ and $\mathbb{N}_n$ are depth map, color map and normal maps in step n respectively. ω_{max} is the maximum weight for each point. Different from rigid transformation where uncertainty of all the points in 3D space are considered as equal, in the case of non-rigid fusion, if a point is further away to the nodes of warping field, we are less likely to believe the registered depth [14]. Therefore, we practically measure this certainty by using the minimum distance from point to nodes and regularize it with half of the unified node distance. Algorithm 2 and Eq. 10,11,12 and 13 show the details for point fusion.

Our improved weighted points based method offers a number of benefits: Point based data management is free of extent limitation. With our fusion based Algorithm 2 and noise point filter approach Algorithm 3, fused geometry can still keep its shape smoothly while avoiding noisy input. The reconstructed geometry preserves more vivid texture and details.

3 Results and discussion

We first validate MIS-SLAM on publicly available in-vivo stereo video datasets provided by the Hamlyn Centre for Robotic Surgery [21]. We also validate MIS-SLAM on ex-vivo phantoms and some simulations and compare with ground truth. In in-vivo validation, three videos with deformation and rigid scope movement are utilized. Other videos either have no deformation or no scope motion. Please note that no extra sensing data other than stereo videos from the scope is used in the our algorithm. The frame rate and image size of in-vivo porcine dataset (model 1 in Fig. 3) is 30 frame per second and 640×480 while the other dataset is 25 frame per second and 720×288. Distance from camera to surface of soft-tissue is between 40 to 70 mm. In our previous research [13], due to poor quality of obtained images and some extremely fast movement of camera, videos tested on porcine with fast or abrupt motion cannot generate good results. In this paper, however, MIS-SLAM can process large scale with data much better robustness. Deformations are caused by respiration and tissue-tool interactions.

3.1 Robustness enhancement

The robustness of MIS-SLAM is significantly improved when global pose from ORB-SLAM is uploaded to GPU and employed as initial scope pose. Fig. 4 shows the comparison between our previous work [13] and the proposed method.

One challenge facing reconstruction problem using stereoscope is the fast movement of scope [13]. Configuration without global pose initialization fails to track motion when camera moves fast. Like traditional SLAM approaches, severe consequences of fast motion are the blurry images and relevant disorder of depths. These phenomena happen especially when current constructed model deforms to match the depth with false edges suffering from image blurring. Fast motion is a very challenging issue because the only data source is the blurry images. ORB-SLAM, however, is a robust feature based system even works in deformable surgery scenario [4] [5] [6]. Though

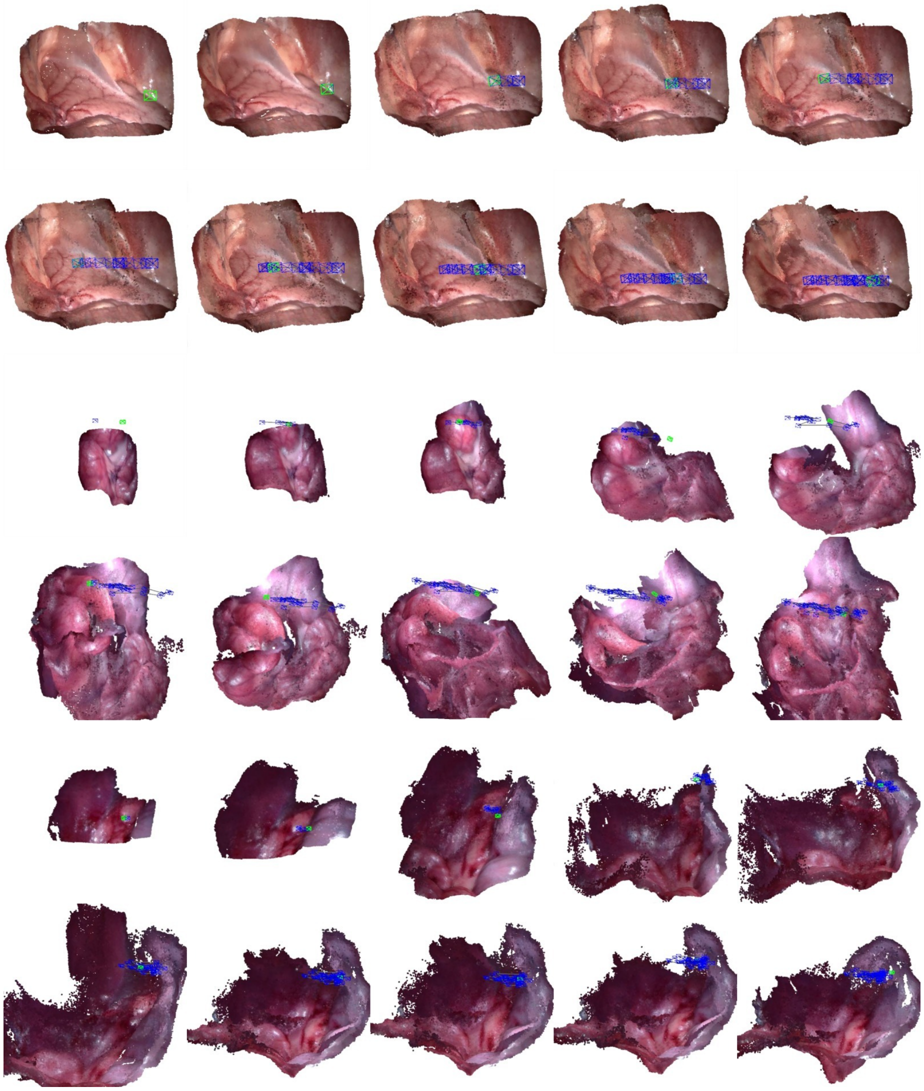

Figure 3: MIS-SLAM process different soft tissues using in-vivo datasets. Pictures present the whole constructed model at different frames. The three videos are (from top to bottom): Abdomen wall (1), abdomen example (2) and abdomen example (3).

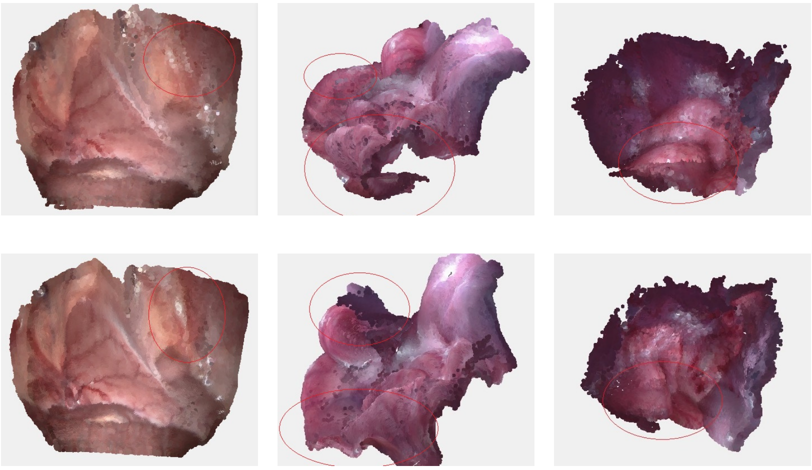

Figure 4: Comparisons between our previous work [13] (First row) and current work (Second row).

based on prior of stationary environment, it still relatively keeps global pose. Our supplementary video clearly shows how initialization of global pose prevents the system from failing to track camera pose.

3.2 Deforming the model and fusing new depth

For model 1, the point cloud density is set to 0.2mm and node density is set to 4mm. For the model 2 and 3, the point cloud density is set to 1mm and node density is set to 10mm. Point cloud downsampling process is carried out by setting a fixed box to average points fill inside each 3D box. The weights for optimization are chosen as $w_{rot} = 1000, w_{reg} = 10000, w_{data} = 1, w_{corr} = 10, w_{corr} = 1, w_r = 1000000, w_p = 1000$. A threshold is set to extract predicted visible points with point to plane distance ε_d as 15mm and angle threshold ε_n as 10°. We measure the error by subtracting projected model image and the observed depth image. The maximum weight is set to 20 and time stamp threshold is set to 10. Threshold ε_n and ε_d for point to depth registration is set as 10 degree and 10mm (20mm for model 2/3). Truncated distance is set as 40mm (60mm for model 2/3).

Threshold is employed to discard some frames when average errors are above due to low-quality depth generated from blurry images. Different from previous research, as we have good initialization of depth image, MIS-SLAM is robust to lost track. Fig. 3 shows the results of soft-tissue reconstruction of MIS-SLAM in different frames, using 3 in-vivo laparoscope datasets [21]. From the results it can be seen that the soft-tissues are reconstructed incrementally with texture.

The average distance of back-projection registration of the three simulation scenarios are 0.18mm (1), 0.13mm (2) and 0.12mm (3). Dataset with ground truth (Hamlyn

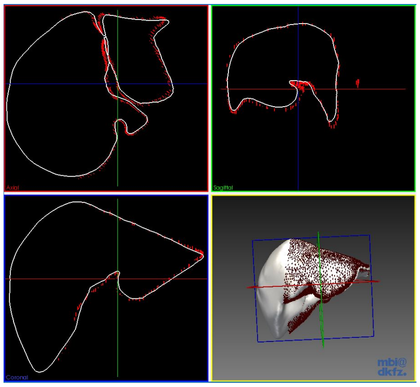

Figure 5: The Axial, Coronal, Sagittal and 3D views of the deformed model and ground truth at the last frame (liver). The red points denote the scan of the last frame.

dataset 10/11) achieves 0.08mm, 0.21mm (Average errors).

3.3 GPU implementation and computational cost

Our system is implemented on heterogeneous computing. The ORB-SLAM runs on CPU. The rest is executed on GPU. Initial global pose and ORB features are transferred to GPU for further optimization. This CPU to GPU data transferring doesn't require much bandwidth as the amount of data is fairly small. CPU initialize OpenGL for visualization framework but we utilize the interoperability from Nvidia's CUDA to directly visualize model in GPU end which saves huge amount of data transferring. Because in most cases GPU module is slower than CPU part, we utilize first-in-last-out feature in the 'stack' data structure to ensure GPU always process the latest data.

The open source ORB-SLAM is executed on desktop PC with Intel Core i7-4770K CPU @ 3.5 GHz and 8GB RAM. We follow [4] to tune the parameters and structures. The average tracking time is 15ms with 640x480 image resolution and 12 ms with 720x288 image resolution. As the frame rate of the three datasets are 25 or 30 fps, ORB-SLAM can achieve real-time tracking and sparse mapping.

By parallelizing the proposed methods for GPGPU, MIS-SLAM algorithm is currently implemented in CUDA with the hardware 'Nvidia GeForce GTX TITAN X'. Current processing rate for each sample dataset is around 0.07s per frame. ORB-SLAM does feature matching on CPU end, saved computation is spent on visualization. Computation increases as model grows and number of nodes rise.

3.4 Validation using simulation and ex-vivo experiments

We also validate the MIS-SLAM on simulation and ex-vivo experiment. In simulation validation process, three different soft-tissue models (heart, liver and right kidney) are downloaded from OpenHELP [22], which are segmented from a CT scan of a healthy, young male undergoing shock room diagnostics. The deformation of the soft-tissue is simulated by randomly exerting 2-3 mm movement on a point with respect to the status of the deformed model from the last frame [23]. We randomly pick up points in the model as the accuracy is measured by averaging all the distances from the source points to target points. Fig. 5 shows the final result of the simulation presented in axial, coronal, sagittal and 3D maps figures. By initializing with camera pose, the overall accuracies are improved from 0.46, 0.68, 0.82 to 0.41, 0.66, 0.62 (mm) regarding to heart, liver and right kidney.

We also tested MIS-SLAM on two ex-vivo phantom dataset from Hamlyn [21]. As the phantom deforms periodically, we do the whole process and compare it with the ground truth generated from CT scan. The average accuracies are 0.28mm and 0.35mm.

3.5 Limitations and discussions

One of the biggest problem in MIS-SLAM is texture blending. Results (Fig. 3 and attached video) indicate that when camera moves, the brightness of visible region shows significant illumination differences from other invisible region. Few tissues even indicate blurry textures. The texture blending procedure involves pixel selection and blending described in Fig. 1. If in perfect registration and identically fused, the reconstruction will only suffer from illuminations from different angles of light. This illumination problem cause systematic difference between two images. In MIS-SLAM, creating clean, pleasing looking texture map in our non-rigid scenario is more difficult than static scenario. There are many other challenges in MIS-SLAM: The number of nodes increases leading to slow optimization; the camera is very close to the tissue and the exposure differs much as it moves, resulting in visible seams in final model; image motion blurring is another problem due to the camera moves fast.

Another improvement will be how to design a better close loop module. ORB-SLAM uses sparse features to relocate camera based on assumption that no relative motion exist in environment. In surgical vision, however, the deforming scenario makes the assumption invalid.

4 Conclusion

We propose MIS-SLAM: a complete real-time large scale dense deformable SLAM system with stereoscope in Minimal Invasive Surgery based on heterogeneous computing. We significantly improved the robustness by solving unstableness caused by fast movement of scope and blurry images. Benefiting from robustness, MIS-SLAM is the first SLAM system achieving large scale scope localization and dense mapping in real-time. MIS-SLAM can potentially be useful for clinical AR or VR applications when camera is moving relatively fast. Future work will be focused on reducing the computational complexity when models grows and exploring an approach to balance textures from different illumination. We will also find a way to do close loop when previous shape is re-discovered.

References

[1] P. Mountney and G.-Z. Yang, "Dynamic view expansion for minimally invasive surgery using simultaneous localization and mapping," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 1184–1187, IEEE, 2009.

[2] O. G. Grasa, J. Civera, and J. Montiel, "Ekf monocular slam with relocalization for laparoscopic sequences," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 4816–4821, IEEE, 2011.

[3] B. Lin, A. Johnson, X. Qian, J. Sanchez, and Y. Sun, "Simultaneous tracking, 3d reconstruction and deforming point detection for stereoscope guided surgery," in *Augmented Reality Environments for Medical Imaging and Computer-Assisted Interventions*, pp. 35–44, Springer, 2013.

[4] N. Mahmoud, I. Cirauqui, A. Hostettler, C. Doignon, L. Soler, J. Marescaux, and J. Montiel, "Orbslam-based endoscope tracking and 3d reconstruction," in *International Workshop on Computer-Assisted and Robotic Endoscopy*, pp. 72–83, Springer, 2016.

[5] N. Mahmoud, A. Hostettler, T. Collins, L. Soler, C. Doignon, and J. Montiel, "Slam based quasi dense reconstruction for minimally invasive surgery scenes," *arXiv preprint arXiv:1705.09107*, 2017.

[6] M. Turan, Y. Almalioglu, H. Araujo, E. Konukoglu, and M. Sitti, "A non-rigid map fusion-based direct slam method for endoscopic capsule robots," *International journal of intelligent robotics and applications*, vol. 1, no. 4, pp. 399–409, 2017.

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

[8] X. Du, N. Clancy, *et al.*, "Robust surface tracking combining features, intensity and illumination compensation," *International Journal of Computer Assisted Radiology and Surgery*, vol. 10, no. 12, pp. 1915–1926, 2015.

[9] D. Stoyanov, "Stereoscopic scene flow for robotic assisted minimally invasive surgery," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 479–486, Springer, 2012.

[10] B. Lin, Y. Sun, X. Qian, *et al.*, "Video-based 3d reconstruction, laparoscope localization and deformation recovery for abdominal minimally invasive surgery: a survey," *The International Journal of Medical Robotics and Computer Assisted Surgery*, 2015.

[11] N. Haouchine, J. Dequidt, M.-O. Berger, and S. Cotin, "Monocular 3d reconstruction and augmentation of elastic surfaces with self-occlusion handling," *IEEE*

transactions on visualization and computer graphics, vol. 21, no. 12, pp. 1363–1376, 2015.

[12] A. Malti, A. Bartoli, and T. Collins, "Template-based conformal shape-from-motion from registered laparoscopic images.," in *MIUA*, vol. 1, p. 6, 2011.

[13] J. Song, J. Wang, L. Zhao, S. Huang, and G. Dissanayake, "Dynamic reconstruction of deformable soft-tissue with stereo scope in minimal invasive surgery," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 155–162, 2018.

[14] R. A. Newcombe, D. Fox, and S. M. Seitz, "Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 343–352, 2015.

[15] M. Innmann, M. Zollhöfer, M. Nießner, C. Theobalt, and M. Stamminger, "Volumedeform: Real-time volumetric non-rigid reconstruction," in *European Conference on Computer Vision*, pp. 362–379, Springer, 2016.

[16] M. Dou, S. Khamis, *et al.*, "Fusion4d: real-time performance capture of challenging scenes," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, p. 114, 2016.

[17] S. Billings, N. Deshmukh, *et al.*, "System for robot-assisted real-time laparoscopic ultrasound elastography," in *SPIE Medical Imaging*, pp. 83161–83161, International Society for Optics and Photonics, 2012.

[18] L. Zhang, M. Ye, P. Giataganas, M. Hughes, and G.-Z. Yang, "Autonomous scanning for endomicroscopic mosaicing and 3d fusion," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pp. 3587–3593, IEEE, 2017.

[19] R. W. Sumner, J. Schmid, and M. Pauly, "Embedded deformation for shape manipulation," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, p. 80, 2007.

[20] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb, "Real-time 3d reconstruction in dynamic scenes using point-based fusion," in *3D Vision-3DV 2013, 2013 International Conference on*, pp. 1–8, IEEE, 2013.

[21] S. Giannarou, M. Visentini-Scarzanella, and G.-Z. Yang, "Probabilistic tracking of affine-invariant anisotropic regions," *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 130–143, 2013.

[22] H. Kenngott, J. Wünscher, M. Wagner, *et al.*, "Openhelp (heidelberg laparoscopy phantom): development of an open-source surgical evaluation and training tool," *Surgical Endoscopy*, vol. 29, no. 11, pp. 3338–3347, 2015.

[23] J. Song, J. Wang, L. Zhao, S. Huang, and G. Dissanayake, "3d shape recovery of deformable soft-tissue with computed tomography and depth scan," in *Australian Conference on Robotics and Automation (ACRA)*, pp. 117–126, ARAA, 2016.