



# Monocular tissue reconstruction via remote center motion for robot-assisted minimally invasive surgery

Peng Li<sup>1</sup> · Ming Tang<sup>1</sup> · Ke Ding<sup>2,3</sup> · Xiaojun Wu<sup>1</sup> · Yunhui Liu<sup>4</sup>

Received: 18 March 2021 / Accepted: 14 July 2021 / Published online: 13 August 2021  
© The Author(s) 2021

## Abstract

In minimally invasive surgery, the primary surgeon requires an assistant to hold an endoscope to obtain visual information from the body cavity. However, the two-dimensional images acquired by endoscopy lack depth information. Future automatic robotic surgeries need three-dimensional information of the target area. This paper presents a method to reconstruct a 3D model of soft tissues from image sequences acquired from a robotic camera holder. In this algorithm, a sparse reconstruction module based on the SIFT and SURF features is designed, and a multilevel feature matching strategy is proposed to improve the algorithm efficiency. To recover the realistic effect of the soft-tissue model, a complete 3D reconstruction algorithm is implemented, including densification, meshing of the point cloud and texture mapping reconstruction. During the texture reconstruction stage, a mathematical model is proposed to achieve the repair of texture seams. To verify the feasibility of the proposed method, we use a collaborative manipulator (AUBO i5) with a mounted camera to mimic an assistant surgeon holding an endoscope. To satisfy a pivotal constraint imposed by the remote center of motion (RCM), a kinematic algorithm of the manipulator is implemented, and the primary surgeon is provided with a voice control interface to control the directions of the camera with. We conducted an experiment to show a 3D reconstruction of soft tissue by the proposed method and the manipulator, which indicates that the manipulator works as a robotic assistant which can hold a camera to provide abundant information in the surgery.

**Keywords** Robotic assistive surgery · Multiview stereo vision · Texture mapping · Remote center of motion

## Introduction

In traditional minimally invasive surgery (MIS), an assistant surgeon is required to hold an endoscope to provide the image view for the primary surgeon. However, the surgeon cannot control the operation scene in person all the time, and a misunderstanding in communication between the two surgeons may cause unexpected injuries. The use of robotic technology solves the aforementioned problems. Using a robotic manipulator and a 3D imaging device, a robotic system can directly

restore the three-dimensional structure of the surgical target area by precisely moving the 3D imaging device.

In robotic-assisted MIS surgery, reconstructing the 3D structure of the surgical field of view is of great significance in the robot-tissue safe interaction; the reconstruction can also be used for preoperative planning and online segmentation for surgical tools. Recently, images and speech have been used for human-robot interaction [1]. Research on 3D perception in the surgical field is often divided into active mode and passive mode: the active mode uses specialized endoscopes to obtain depth information, which is costly and has few clinical applications, and the passive mode uses stereo cameras or monocular endoscopes. Due to the rapid development of machine learning and GPU technology, a monocular camera is a cost-effective component and easy to use. Considering the wide application of monocular endoscopes, the 3D reconstruction of soft tissue by multiview vision is reasonable.

As early as the 1990s, Deguchi used the measurement matrix generated by an image sequence to acquire 3D infor-

✉ Xiaojun Wu  
wuxj@hit.edu.cn

<sup>1</sup> Harbin Institute of Technology Shenzhen, Shenzhen 518055, China

<sup>2</sup> Guangdong University of Technology, Guangzhou, China

<sup>3</sup> Foshan Nanhai Guangdong University of Technology CNC Equipment Cooperative Innovation Institute, Foshan, China

<sup>4</sup> The Chinese University of Hong Kong, Shatin NT, Hong Kong, China

mation by matrix decomposition; the resulting precision, however, was not high [2]. Burschka used the relationship between the feature points of each image frame to estimate the camera posture and restore the 3D points of each image frame one by one, but such feature extraction and matching are time-consuming and prone to errors [3]. The aforementioned methods only obtain sparse 3D point clouds. The scale of the point cloud model depends on the number of feature points in the image frame; thus, it is necessary to densify the point cloud. A dense algorithm based on the PMVS method, which can generate a relatively dense point cloud model [4], is proposed. Goesele proposed a method based on depth maps and generated a denser point cloud model that produces a depth map for each image frame [5]. Considering that the point cloud model is still a point set lacking structure, researchers have investigated surface reconstruction; the most commonly used surface reconstruction method is the Delaunay triangular mesh generation [6]. Although the Delaunay algorithm is simple and has a short runtime, the result is unstable and easily disturbed by noise. To reconstruct a more stable mesh model, Kazhdan combined the inhomogeneous information in the input point cloud, globally constructed the symbol distance field, and finally obtained the mesh model [7]. Fuhrmann [8] used the scale of sample points to construct an octree, locally construct the symbol distance field and generate a mesh model. The triangular mesh is dense and structural, but it lacks texture and does not have a satisfactory visual effect. Lempitsky [9] obtained the preliminary texture model by view selecting of all patches and globally alleviated the influence of seams on the model at the global. Perez [10] further eliminated the seams locally. Waechter [11] proposed a comprehensive texture mapping framework that performed well on some texture data sets.

Future semiautomatic surgery requires robots to perceive and recognize soft tissue in the body cavity and coordinate with the surgeon to complete the procedure. The 3D model can be used for online planning and registration purposes. Thus, this paper aims to build a robotic 3D soft tissue reconstruction platform, including a reconstruction algorithm and robot, to acquire online information and control camera motion more steadily and precisely for the primary surgeon. We propose a 3D reconstruction method using a monocular camera and employ a collaborative manipulator (AUBO i5) to hold the camera. Furthermore, a voice control interface is provided to the surgeon, which allows them to control the camera directly and avoid misunderstandings with the assistant. Moreover, the movement of the end effector and camera, a pivot constraint motion (remote center motion), is also considered and implemented to fulfill the MIS requirements.

The rest of this paper is organized as follows: “Reconstruction of soft-tissue” presents the complete reconstruction algorithm for soft tissue. “The RCM implement of surgical manipulator” introduces the implementation of the RCM-

based manipulation platform, and “Experiments and results” presents the results of the experiments. “Conclusion and future work” discusses our conclusions.

## Reconstruction of soft tissue

The image frame-based 3D reconstruction algorithm includes sparse point cloud reconstruction, point cloud densification, surface reconstruction and texture mapping, as indicated in Fig. 1. Sparse point cloud reconstruction is the first step to obtain feature points, the densification of these points and to obtain enough data to generate the surface. Finally, the meshing grids are filled with color pixels by texture mapping. The input of this pipeline is the image frames of soft tissue, and the output is the high-quality 3D model.

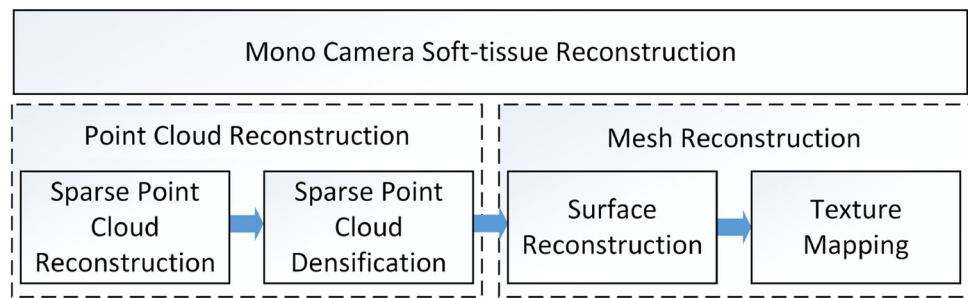
### Sparse point cloud reconstruction of soft tissue

Sparse reconstruction is also called feature-based reconstruction. This section includes three steps: feature point extraction, matching, and incremental reconstruction. The first step of the reconstruction is to extract enough features from the input image frames. Considering the lack of features in soft tissue, the SIFT and SURF feature extractions are selected to generate more feature points [12,13].

For matching the image frame pairs, low-resolution matching methods are used to disqualify obviously mismatched image pairs so that the matching efficiency increases [14]. In the feature matching stage, geometric constraints are introduced, and the mismatched feature pairs are eliminated by the constraints between the image pairs. The RANSAC estimation method is used to obtain a reliable basis matrix, and the matching feature points satisfying the geometric constraints are selected as the final matching results. Thus, a multilevel matching strategy is proposed:

- (a) Preprocessing stage: After extraction from the image frame, the feature points are arranged in descending order and take the first  $h$  feature points as feature subsets. Then, the minimum matching point threshold  $\alpha$  of the subset and the global minimum matching point threshold  $\beta$  are set.
- (b) Every two images form a pair of images, and every image pair is checked by traversal methods:
  - (1) Matching feature points in the feature subset of image pairs: If the number of matches is less than the threshold value  $\alpha$ , the remaining steps are skipped and returned.
  - (2) Matching feature points in the feature full set of image pairs: If the number of matches is less than the threshold value  $\beta$ , the remaining steps are skipped and returned.

**Fig. 1** Pipeline of the 3D soft-tissue reconstruction



- (3) The RANSAC estimation method is used to disqualify the feature points that do not meet the geometric constraints, and all the matching points are recorded and returned.

There are often thousands of features in an image, and the corresponding descriptors are 128 dimensional vectors, which makes the matching of these vectors a time-consuming task. To reduce the matching time and accelerate the matching of feature points between a pair of images, this paper employs the cascade hash matching method [15].

The cascaded hash matching algorithm has three stages, including hashing table look up, hashing remapping and Euclidean distance filtering, as indicated in Fig. 2. Hashing table lookup roughly screens out the candidate feature points using short codes to perform a coarse search. The purpose of hashing remapping is to remap the candidates to the Hamming space, obtain the binary string hash coding vector, calculate the Hamming distance, search the most similar feature points in the Hamming space, and select the top  $k$  items as the final matching candidates. Euclidean distance filtering searches Euclidean space to obtain the final matching feature points. Feature points are accepted as final matching points after passing Lowes [16] ratio test requirements.

After matching, incremental reconstruction is used to find the proper initial image pair and load images one after another to enrich the 3D point set, which is solved by the structure from motion (SFM) method [17,18]. First, an initial image pair is selected according to the number of matching points and the triangle measurement angle between these two images, triangulation is performed by using the common feature points of the tissue, and finally, the preliminary 3D point cloud is generated. Due to the existence of noise and the slight difference in feature point positions, the bundle adjustment (BA) optimization method is used to adjust and restore the internal and external parameter matrix of the camera and 3D point coordinates. Second, the reconstruction of the remaining views is performed. Every time a new view is inserted, the corresponding relationship between the restored 3D points and the feature points needs to be established. The motion parameters between views are solved by the PnP solver based on random sample consensus (RANSAC), and the 3D space points are restored through triangulation. Because the num-

ber of feature points is much greater than that of the camera parameters, the optimization of 3D point coordinates is very time consuming. Therefore, the parameters of the camera and 3D point coordinates are calculated simultaneously every time five images are inserted into the reconstruction queue, and only the camera parameters are updated for the remaining image. Thus, the initial feature points are restored.

### Dense reconstruction

The sparse point cloud model cannot present enough soft tissue surface information; thus, to achieve 3D dense point clouds, a method based on depth images is used [5].

First, neighborhood images are selected for every image to form stereo image pairs. Then, from the restored 3D points, the region growing method is used to calculate the depth information of each pixel in the current image. Second, according to the projection of the 3D feature points of the sparse point cloud to the current image, the neighborhood view is selected for each feature point, the normalized correlation coefficient (NCC) average value is calculated, and the priority queue is established. Then, a seed point is removed from the queue to perform nonlinear optimization on its depth value, and the average NCC is recalculated after optimization. If the NCC value is significantly improved or there is no depth value in the surrounding neighboring pixels, the neighboring pixels are added to the queue until the queue is empty. Then, the region growth method generates a corresponding depth map for each view, that is, a three-dimensional point in the camera coordinate system where each view is located.

After the sparse point cloud is reconstructed, the relative pose of each view is known. Thus, it is only necessary to use the transformation matrix to map the pose to the global coordinate system to generate a three-dimensional dense point cloud.

### Surface and texture reconstruction

To add the topology information of the dense point cloud model, we transform the input dense point cloud into a triangular mesh of soft tissue by the method described in [8]. It uses the scale of sampling points from the input point cloud

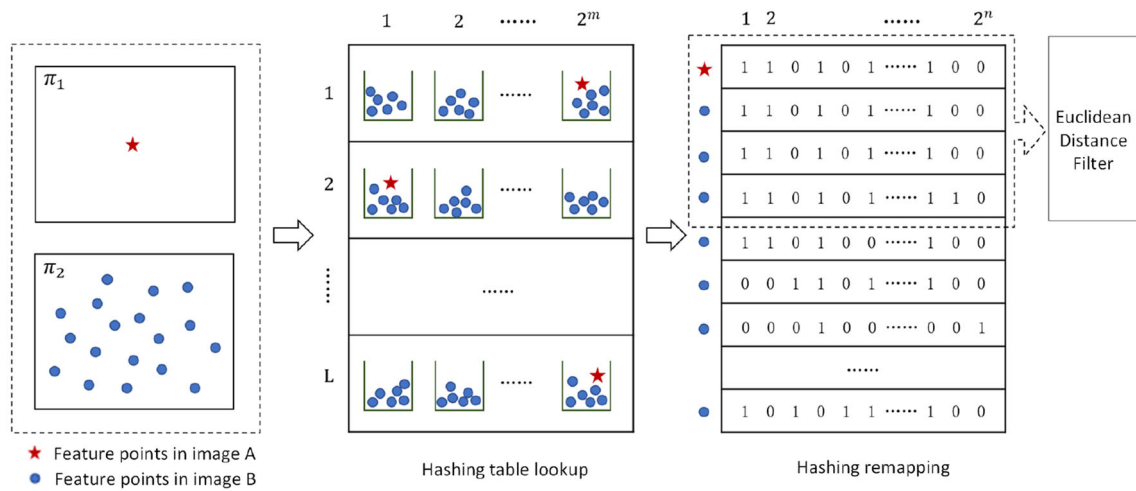


Fig. 2 Flowchart of cascade hashing matching

to create an octree and calculate the signed distance value of each node. The triangular mesh model is then obtained based on the marching cube algorithm [19].

Texture reconstruction attaches texture information to the input triangle mesh. The specific content of texture mapping includes two stages: image view selection of the patch and seam repair between patches. In the first stage, an image view is assigned to each patch on the triangle mesh to extract pixel information and obtain a preliminary texture model; in the second stage, pixel adjustments are made for the points on the patch to eliminate the seams between patches and generate a fine texture model.

We implement image view selection by establishing a Markov random field, which includes a data item and smoothing items [11]. The objective function  $E(I)$  includes a data item  $E_{dat}$  and a smoothing item  $E_{smo}$ . The data item satisfies the requirement of viewing angle selection by summing the gradient of the projection area; the smoothing item tries to ensure that adjacent patches are given the same label. The vector  $I$  to be solved contains all of the view labels with their corresponding patches.

$$\begin{aligned}
 E(I) &= \sum_{F_i \in \text{Faces}} E_{dat}(F_i, I_i) + \alpha \sum_{F_i \cap F_j} E_{smo}(I_i, I_j) \\
 E_{dat} &= - \int_{\Omega(F_i, I_i)} \|\nabla(I_i(p))\|_2 dp \\
 E_{smo}(I_i, I_j) &= \begin{cases} 0 & I_i = I_j \\ 1 & I_i \neq I_j \end{cases}, \tag{1}
 \end{aligned}$$

where  $\Omega(F_i, I_i)$  represents the projection area of the  $i$ -th patch projected on the image labeled with  $I_i$ ;  $\nabla(\cdot)$  is the pixel gradient of the image, where the Sobel operator is used as the pixel gradient;  $F_i \cap F_j$  is the set of adjacent patch pairs; and  $\alpha$  is the coefficient of the smoothing term that controls

smoothness. After constructing the Markov random field, optimization methods such as  $\alpha$ -expansion [20] are used to find the optimal label vector  $I$  so that the objective function  $E(I)$  reaches a minimum, thereby determining the viewing angle label to which each triangle patch belongs.

Problems such as the illumination difference between the viewing angles of adjacent patches will cause seams between adjacent patches. A common solution is to add a color adjustment value for all patch vertices and then obtain the value for all the pixels in patches by area interpolation.

Let  $g$  represent the adjustment value of the vertices on the patch and  $f$  represent the initial color value of the vertices on the patch, as shown in Fig. 3. According to [11], the energy function  $E(g)$  is defined, in which the data item  $E_d(g)$  constrains the same two vertices in the seams after color adjustment, and the smoothing term  $E_s(g)$  limits the difference in color changes of vertices other than the seams to preserve the details inside the surface.

$$\begin{aligned}
 E(g) &= E_d(g) + E_s(g) \\
 &= \sum_{v \in S} (f_v^l + g_v^l - f_v^r - g_v^r)^2 \\
 &\quad + \lambda \sum_{v' \notin S} \sum_{v'' \in N(v')} (g_{v''} - g_{v'})^2, \tag{2}
 \end{aligned}$$

where  $S$  represents the set of mesh vertices on all seams,  $N(v')$  is the set of mesh vertices adjacent to the mesh vertex  $v'$  and owns the same view label;  $f_v^l$  and  $f_v^r$  represent the pixel values of vertices  $v$  extracted from the input image labeled  $l$  and  $r$ , respectively;  $g_v^l$  and  $g_v^r$  represent the adjustment value of vertices  $v$  extracted from the input image labeled  $l$  and  $r$ , respectively.  $g_{v''}^l$  and  $g_{v'}^l$  represent the adjustment value of vertices  $v''$ , and  $v'$  and  $\lambda$  represent the weighting factor.



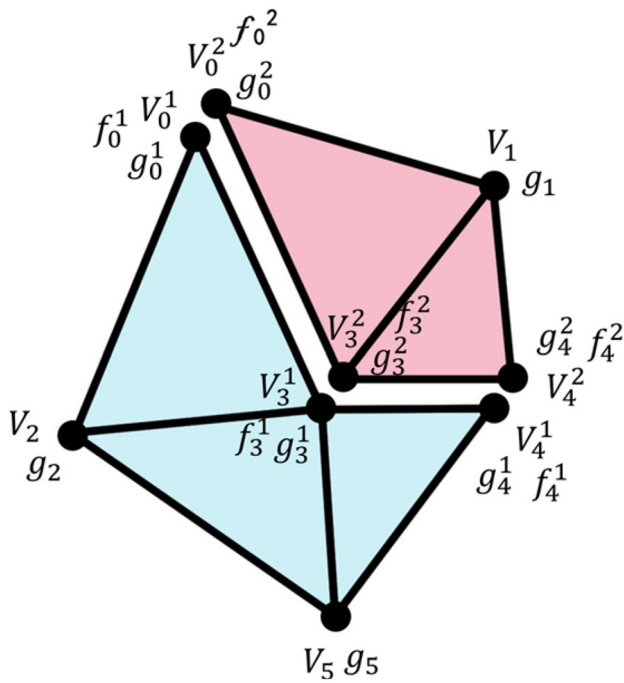


Fig. 3 Color adjustment of the vertices on the patch

However, the data term  $E_d(\mathbf{g})$  tends to reduce the pixel value difference on both sides of the vertex at the seams, while the smoothing term  $E_s(\mathbf{g})$  tends to reduce the difference in the amount of adjustment of the vertices not at the seams. These two terms cannot constrain the amplitude of the amount of color adjustment of the vertex. Assuming that the optimal solution  $\mathbf{g}'$  is obtained, then there is another solution  $\tilde{\mathbf{g}}'$  fitting the model. The two solutions have the following relationships.  $n$  represents the size of the scale, and  $\mathbf{e}$  represents the unit vector along the direction of  $\mathbf{g}'$ . When Eq. (2) is used for solving texture adjustment, it cannot be guaranteed to obtain an appropriate adjustment amount. When the amount of color adjustment becomes too large, the visual effect of the entire texture model may be affected.

$$\tilde{\mathbf{g}}' = \mathbf{g}' + n\mathbf{e}. \tag{3}$$

Thus, this paper proposes an improved model to find the adjustment values by using the following energy function. The model preserves the data and smoothing terms and adds a penalty term to limit the magnitude of vector  $\mathbf{g}$ , which suppresses the preference for larger adjustment values and obtains more reliable results. For the penalty term, the L2 norm is used to suppress the variation of the variable.

$$E(\mathbf{g}) = E_d(\mathbf{g}) + E_s(\mathbf{g}) + E_r(\mathbf{g}) = \sum_{v \in S} (f_v^l + g_v^l - f_v^r - g_v^r)^2$$

$$+\lambda_1 \sum_{v' \notin S} \sum_{v'' \in N(v')} (g_{v''} - g_{v'})^2 + \lambda_2 \|\mathbf{g}\|^2, \tag{4}$$

where  $E_r(\mathbf{g})$  represents the penalty term,  $\lambda_1$  is the coefficient for the smoothing term and controls smoothness, and  $\lambda_2$  is the coefficient of the penalty term and adjusts the variation of the solution for all the seam vertices.

Then, we can obtain the simplified Eq. (5) according to [11].

$$E(\mathbf{g}) = \mathbf{g}^T \left( \mathbf{A}^T \mathbf{A} + \frac{\lambda_1}{2} \mathbf{L} + \lambda_2 \mathbf{E} \right) \mathbf{g} + 2\mathbf{b}^T \mathbf{A} \mathbf{g} + \mathbf{b}^T \mathbf{b}, \tag{5}$$

where  $\mathbf{A}$  represents a sparse matrix composed of the subscripts of the seam vertices in the vector  $\mathbf{g}$ ,  $\mathbf{L}$  represents a sparse matrix related to the subscript of the slot vertex in vector  $\mathbf{g}$ ,  $\mathbf{E}$  represents the identity matrix and  $\mathbf{b}$  is the vector composed of the color difference of the seam vertices under different view labels. Hence, this optimization problem and the vector  $\mathbf{g}$  can be solved with the conjugate gradient method [21].

### The RCM implement of surgical manipulator

In robot-assisted MIS, a surgical robot holds endoscopes or instruments to insert into the patient’s abdomen through an incision on their stomach. Thus, the movement of the endoscope must follow a pivot constraint known as the remote center of motion (RCM). To achieve RCM motion, the classical double parallelogram and circular guide rail mechanism serve as the driving mechanism for surgical robots [22].

In this paper, RCM motion planning based on a collaborative 6-DOF robot (AUBO i5, AUBO robotics Inc.) and a camera mounted on its end instrument is implemented for soft tissue image acquisition. The image sequence is used as input data for the above 3D reconstruction algorithm. The camera fixed at the end of the instrument imitates and serves as an endoscope to perform the various movements we required. For intuitive usage by surgeons and to create multiple positions for acquiring soft tissue images, six modes of camera movement are defined: that is, “move up”, “move down”, “move left”, “move right”, “feed” and “back”. As shown in Fig. 4, these movements are based on the plane of soft tissue, which is perpendicular to the central axis of the instruments and camera.

The RCM motion planning is based on the Robot Operate System (ROS) and the MoveIt! plugin, which helps us move the camera when designing its movement in ROS. Therefore,

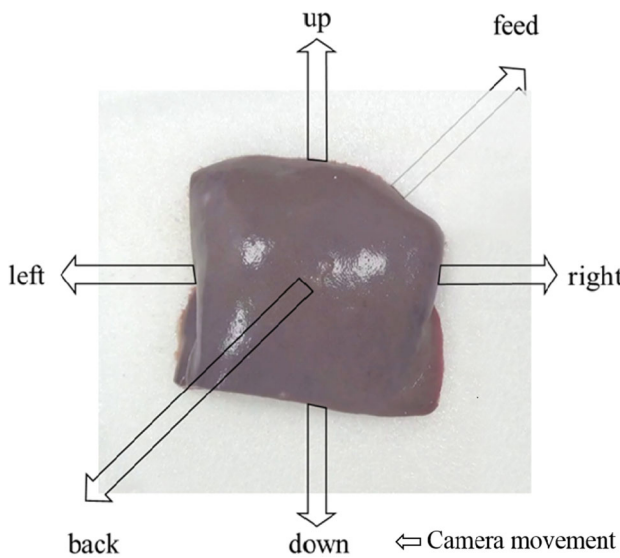


Fig. 4 Directions for movement based on the soft-tissue plane

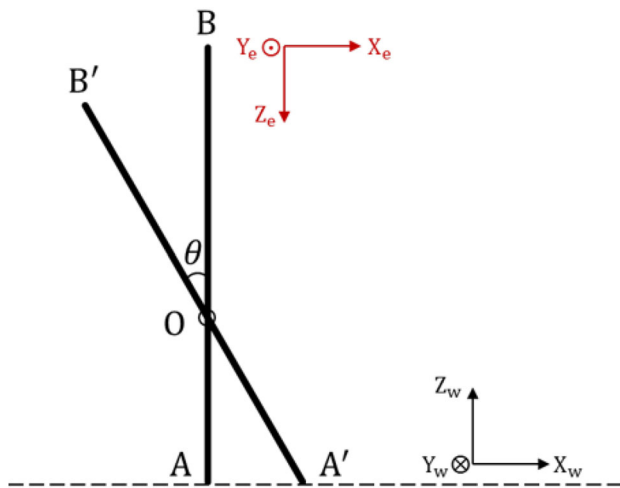


Fig. 5 Right movement of the instrument and camera

we need to define the position and pose of the camera on the robot model. The MoveIt! plugin receives the command and converts it to the joint space so that the camera can move along the predefined position and posture.

Among these motions, the “left”, “right”, “up” and “down” motions have similar tip motions, while “feed” and “back” have linear motions along the instrument’s central axis. We analyze the move right translation motion, as the other motions follow in a similar way.

Taking “moving right” as an example, we ensure that the camera moves right in a straight manner; the motion diagram of the instrument tip for the movement is indicated in Fig. 5. We set the initial position of point  $B$  to  $(X_B, Y_B, Z_B)$  and rotate the instrument right from  $A$  to  $A'$  around the pivot constraint  $O$ . Since the length  $l_t$  of the instrument and the position  $(X_O, Y_O, Z_O)$  of point  $O$  are known, the position

$(X'_B, Y'_B, Z'_B)$  of point  $B'$  can be derived using Eq. (6). The pose of instrument  $R$  is derived from Eq. (7), in which  $\alpha$ ,  $\beta$  and  $\gamma$  are the Euler angles of matrix  $R$ .

$$\begin{bmatrix} X'_B \\ Y'_B \\ Z'_B \end{bmatrix} = \begin{bmatrix} X_B - l_t \sin \theta + (l_t - l_{OB}) \tan \theta \\ Y_B \\ Z_B - l_t(1 - \cos \theta) \end{bmatrix}, \quad (6)$$

$$(\alpha, \beta, \gamma) = (\pi, -\theta, 0). \quad (7)$$

## Experiments and results

### Soft tissue reconstruction results

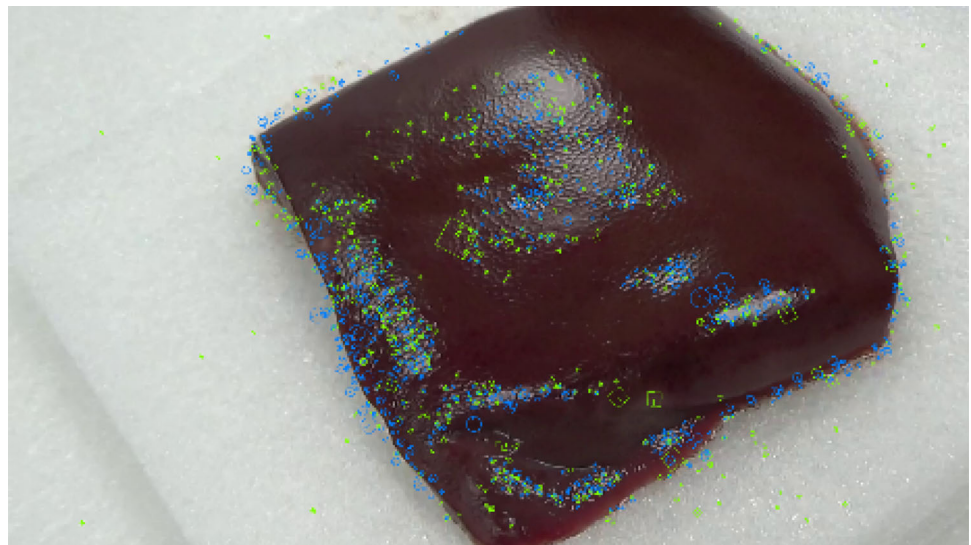
In the experiment, we use a pork liver, which is easily bought in the supermarket and is usually refrigerated. We collected an image of the pork liver image with a camera, as shown in Fig. 6. The positions of the SIFT features and the SURF features are represented by blue and green circles, respectively. The size of the circle and the shape represent the scale of the feature points. There are a total of 608 SIFT feature points and 857 SURF feature points; part of the feature point areas are at the boundary of the pig liver image, that is, at the junction with the background image, and there are some potential reflection areas missing feature points, but, overall, there are still relatively rich features.

To conduct feature matching between two images, a similarity search is performed first and then geometric constraints are used to eliminate mismatched feature points. In the similarity search, cascaded hash matching is used to detect the corresponding feature points. In the geometric constraint stage, basic matrix estimation based on RANSAC is used to screen out the correspondence that meets the epipolar constraint feature points. Figure 7a is the initial feature matching pair obtained through the similarity search of the feature descriptor. The feature matching pair is further restricted by the basic matrix estimation, and the matching pair that does not meet the geometric constraints is eliminated to obtain a more accurate matching pair, as indicated in Fig. 7b.

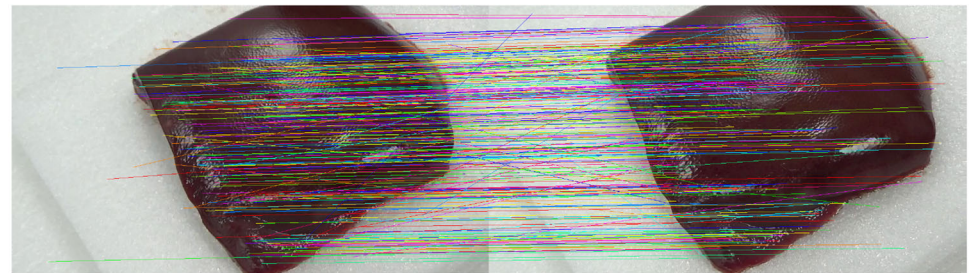
To verify the cascaded hash feature detection searching algorithm, a set of comparative tests is conducted by selecting the two images in Fig. 7, and extracting the number of  $n_1 = 1485$  and  $n_2 = 1625$  groups in the image feature extraction process. Setting the theoretical number of successful matches to a total of  $n = \min(n_1, n_2) = 1485$  groups, the matching point rate is the ratio of the number of successful matches to the theoretical value. The experimental results are shown in Table 1. The results show that the cascaded hash feature search method improves the feature matching speed while ensuring the matching accuracy.

Furthermore, comparative tests are carried out to verify the feature matching strategy in this paper. Table 2 shows

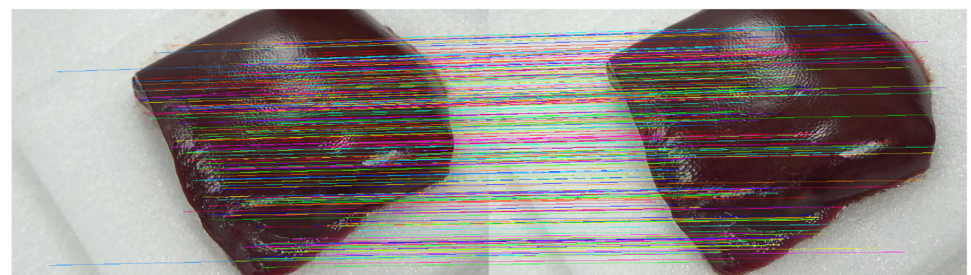
**Fig. 6** SIFT and SURF feature points of a pork liver



**Fig. 7** Image feature matching based on epipolar geometric constraint screening



**(a)** Initial image feature matching



**(b)** Image feature matching that satisfies geometric constraints

**Table 1** Feature point matching comparison of the brute force search and cascaded hash methods

Matching strategy	Feature matching running time (ms)	Number of matching features	Matching point rate%
Brute force search	64	158	10.64%
Cascaded hash search	33	167	11.24%

the runtime of the image feature matching phase, the number of valid image matching results and the feature tracks under the following strategies. The matching strategy “Baseline” indicates that low-resolution prescreening and cascade hash feature detection are not used for feature matching; the matching strategy “Combined” indicates that both low-resolution prescreening and the cascade hash method are used

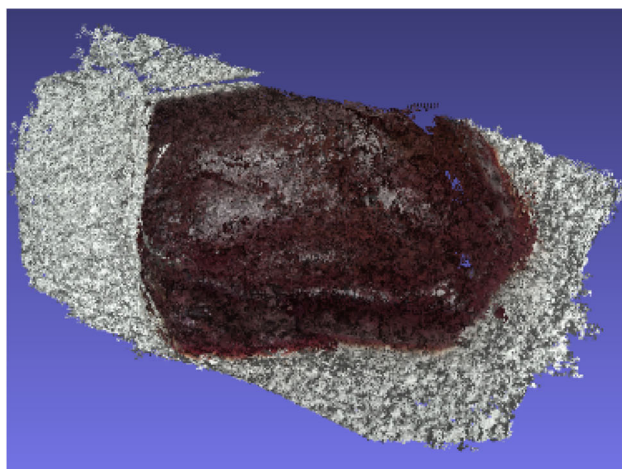
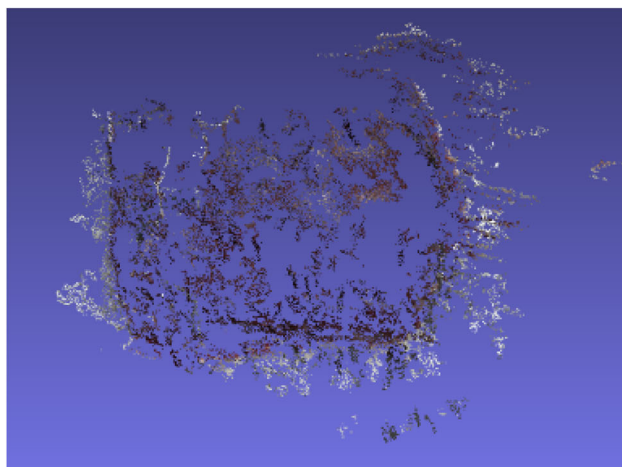
during the matching stage. Compared with “Baseline”, the cascade hash strategy reduces the search time by 19.2% while the matching rate drops 2.8%, and the “Combined” strategy reduces the search time by up to 21.47% while the matching rate drops by 2.46%.

This paper compares two dense reconstruction methods. Figure 8a shows the result of point cloud densification



**Table 2** Four strategies for feature matching

Matching strategy	Valid image matching pairs	Valid feature tracks	Consumption time (ms)
Baseline	694	21,580	25,027
Low resolution	689	21,542	23,151
Cascade hash	805	20,962	20,217
Combined	794	21,047	19,653

**(a)** Based on depth map**(b)** Based on CMVS-PMVS**Fig. 8** Densification of the point cloud model

based on the depth map, and Fig. 8b shows the dense point cloud model output by the CMVS-PMVS algorithm. Table 3 reflects the performance comparison of the two algorithms. The results show that although the densification method based on depth map reconstruction recovers enough space points to obtain a more realistic three-dimensional model, it has a longer runtime.

Experimental point cloud meshing and texture reconstruction are carried out. Figure 9a is the meshing result with the

redundant points replaced with a mesh composed of triangular surfaces to simplify the 3D model under the premise of ensuring the accuracy of the surface. Figure 9b is the result of texture mapping, which contains texture information of the pork liver. However, in the local region of the texture model, there are texture seams caused by light differences.

The zoomed-in image of the area can be seen in Fig. 9e; the sharp seams shown influence the overall effect of the texture model. Figure 9c is the result of the seam repair method according to previous work [11]. As shown, the output result is overexposed, which does not satisfy visual requirements. Figure 9d shows that the seam is sutured according to the proposed model of this paper. Figure 9f shows the zoomed in area, which is the same area as in Fig. 9b, after repair, and verifies the effectiveness of the proposed model. This seam repair is similar to the triangular mesh level smoothing operation, which relieves the seam in the texture model and improves the visual effect of the texture model.

The running time and structural information of each phase of the 3D reconstruction algorithm is recorded, as indicated in Table 4. The computer for running the algorithm has an Intel i5-8500 3.0 GHz CPU and a 16 GB RAM. In the dense point cloud reconstruction stage, many redundant 3D point data are obtained. However, the mesh generation method greatly reduces the memory consumption of the 3D model and ensures the accuracy of the model. The presented 3D reconstruction algorithm relies only on CPU computing resources, and takes approximately 17 min from the input image sequence to the final texture model.

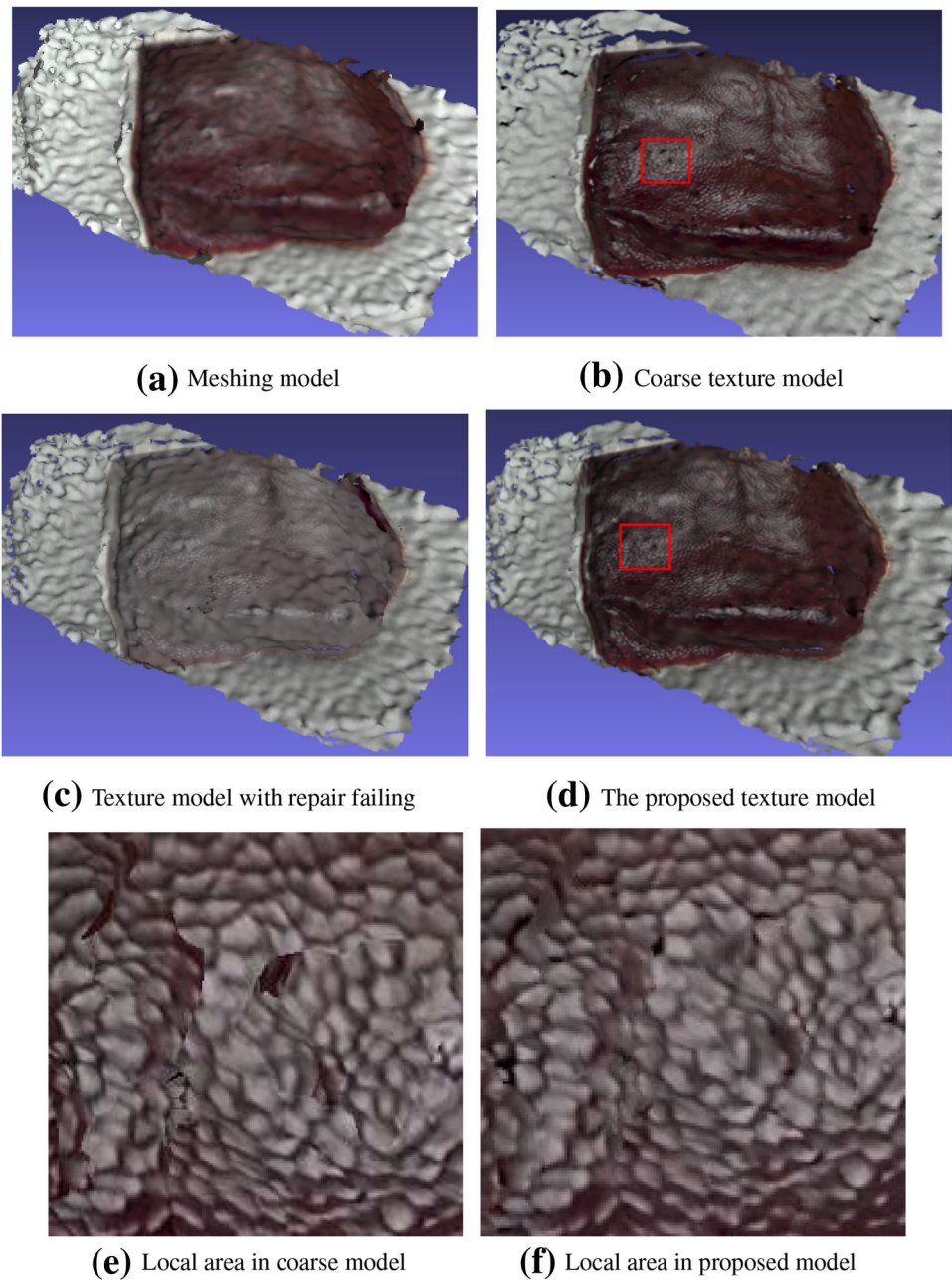
The 3D model obtained by image-based 3D reconstruction differs from the real object by a scale factor; that is, it lacks a similar transformation, so it is often impossible to directly compare the two to verify the accuracy of 3D point cloud reconstruction. The reprojection error is commonly used as an evaluation standard. We calculate the reprojection error of all views separately and obtain the root mean square (RMS) reprojection error vector of the point cloud model on each view, as shown in Eq. (8).

$$e_j = \sqrt{\frac{1}{N_j} \sum_{i=1}^{N_j} (x_i - P_j X_i)^2}, \quad (8)$$



**Table 3** Comparison of the two densification methods

Densification method	Calculation device	Running time (s)	Number of restored points
CMVS–PMVS	GPU	51	24,083
Dense map	CPU	149	1733,195

**Fig. 9** Mesh generation and texture mapping for point cloud

where  $N_j$  represents the number of 3D points visible on view  $j$ ;  $\mathbf{x}_i$  is the coordinate of the feature point corresponding to the  $i$ -th 3D point on view  $j$ ;  $\mathbf{P}_j$  represents the projection matrix of view  $j$ ; and  $\mathbf{X}_i$  is the coordinates of the  $i$ -th 3D point.

The results are shown in Fig. 10. It can be seen from the figure that the RMS error of each view is mostly within one pixel, and the average projection error is 0.73 pixels, which indicates the accuracy of the sparse point cloud reconstruction results in this paper.

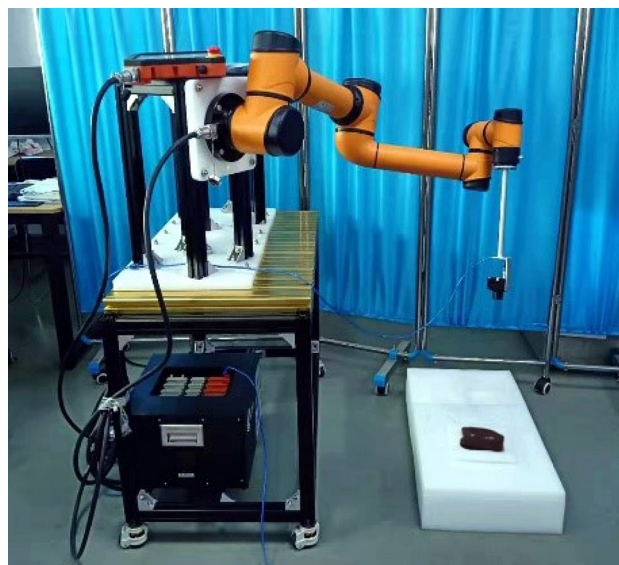
**Table 4** Results of each phase of the 3D reconstruction

Model name	Vertex	Facet	Runtime(s)
Sparse point cloud	5821	51	57
Dense point cloud	1733,195	149	149
Mesh model	38,702	75,542	68
Texture model	38,702	75,542	786

**Soft tissue image acquisition via RCM motion**

An experimental platform based on a manipulator (AUBO i5) is implemented to verify the feasibility of robot-assisted endoscope holding and 3D reconstruction, as shown in Fig. 11. The robot is fixed on a base, and a rod shape instrument and a camera are fixed on the end of the manipulator. An industrial computer (Intel i5-8500, RAM 16G) is used to receive the image from the camera and send motion commands to the Moveit! plugin in the ROS. Additionally, we implemented a voice control interface based on AlexNet [23] to provide the operator (surgeon) with voice control of the direction of the endoscope. The advantage of the voice control interface is that the operator (surgeon) can control the camera motion while both their hands are busy. We also employ the RVIZ visualization tool in ROS to observe and verify the motion planning of the manipulator.

Figure 12 shows four motion modes of the manipulator in the ROS simulation environment, including “left”,

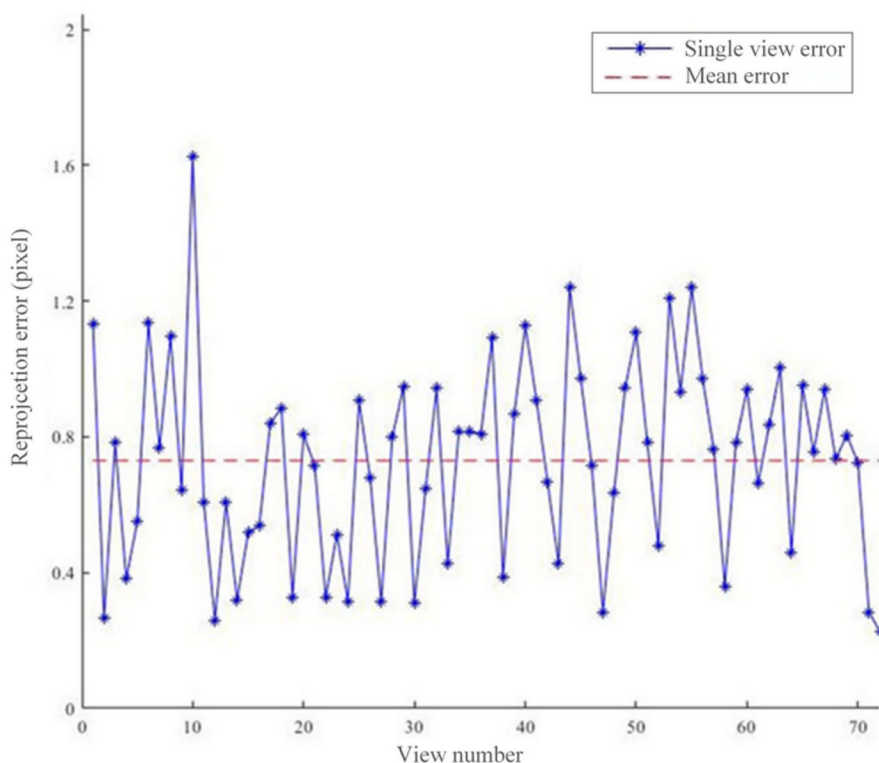


**Fig. 11** The 3D vision reconstruction experimental platform

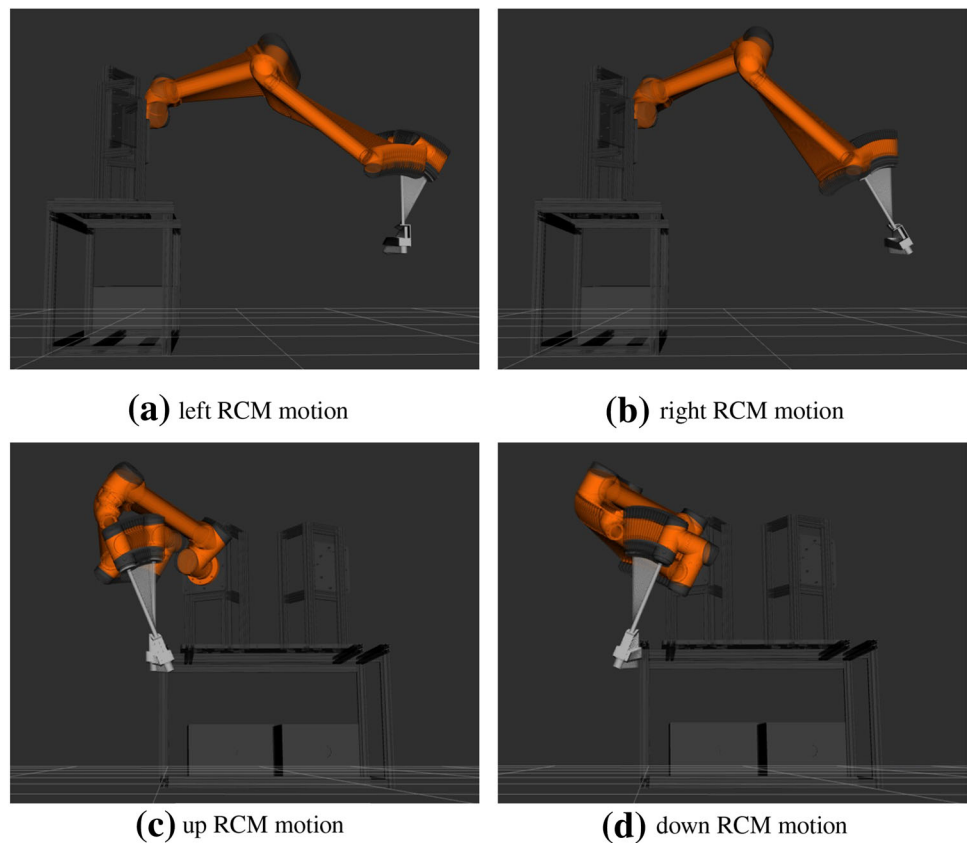
“right”, “up” and “down”. The simulation results show that the motion planning of the manipulator can adequately implement the RCM motion.

We carried out the image acquisition experiment with the camera under RCM motion. Figure 13 shows the state of the manipulator under voice control by the operator and the images acquired by the camera under these RCM motions.

**Fig. 10** Reprojection error of 3D point cloud



**Fig. 12** The motion implementation in RIVZ



Through the above experiments, we collect soft tissue images captured from multiple perspectives during the motion of the camera. Then, the image sequence is sent to the proposed 3D reconstruction algorithm, the results of which are shown in Fig. 14.

There are some holes in the reconstructed model because of the partial reflective area and because the RCM movement of the camera limits the image acquisition positions, but the overall reconstruction effect, which reflects the shape and color information of the pork liver, is accurate. The results show that the proposed 3D reconstruction algorithm can successfully recover a high-quality 3D soft tissue model under the RCM motion of the robot using voice commands from a user.

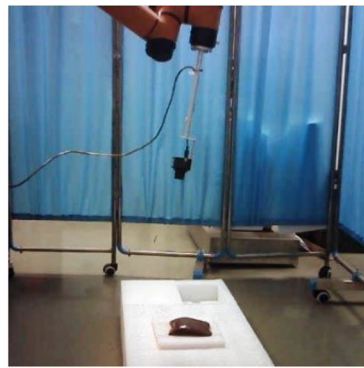
## Conclusion and future work

In this paper, we proposed a 3D reconstruction algorithm for soft tissue based on a camera and implemented an assistive manipulator to perform image sequence acquisition via voice control commands. In the reconstruction algorithm, we first performed initial feature point extraction and proposed a multilevel matching acceleration strategy to improve the

matching efficiency. We used a depth map-based densification method, and meshed the dense point cloud to reveal the surface of soft tissue. Furthermore, we used texture mapping to obtain a more realistic soft-tissue model. To eliminate the seams, we proposed an improved model to solve the repair of texture seams. Comparative experiments were carried out to verify the proposed model. We implemented a 6-DOF manipulator with RCM constraint motion and fixed the camera on its end to imitate an endoscope inserted in the patient abdominal cavity to obtain the images. This manipulator can drive the camera with six mode motions under a voice control interface, such as left, right, up, down and so on. Experiments verified that the user can move the robot via a defined voice command and obtain a visualized 3D soft tissue model. Limitations of current research include the reconstruction error with ground truth and the time consumption of texture mapping. Since we want the real 3D soft tissue model rather than the model lacking the scale factor in multiview 3D reconstruction, we use the RMS reprojection error to measure the accuracy. Future work can include using a GPU to accelerate the texture mapping algorithm and implementing calibration methods to obtain the ground truth value of the soft tissue.



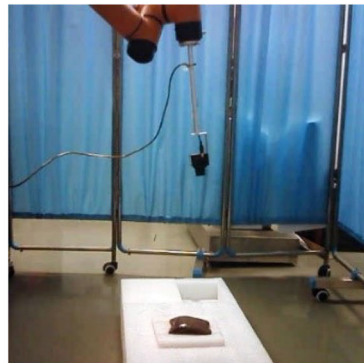
**Fig. 13** Soft tissue image acquired from the robot via voice control



**(a)** left RCM motion of the robot



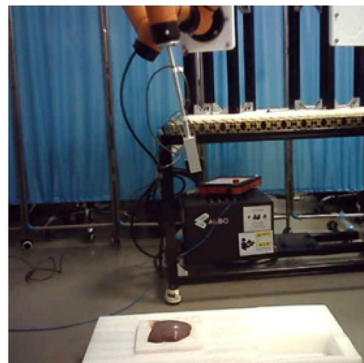
**(b)** soft-tissue image captured



**(c)** right RCM motion of the robot



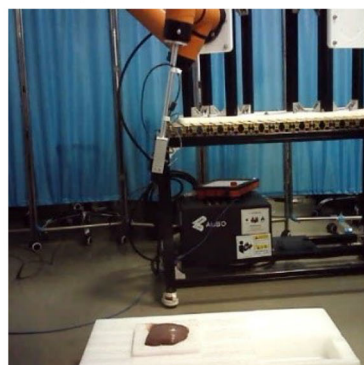
**(d)** soft-tissue image captured



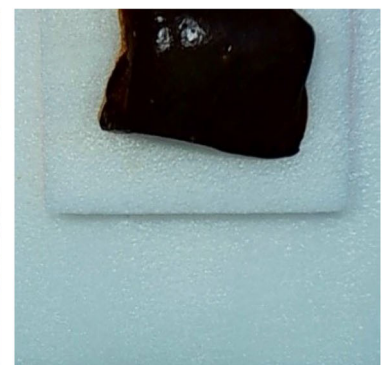
**(e)** up RCM of the robot



**(f)** soft-tissue image captured

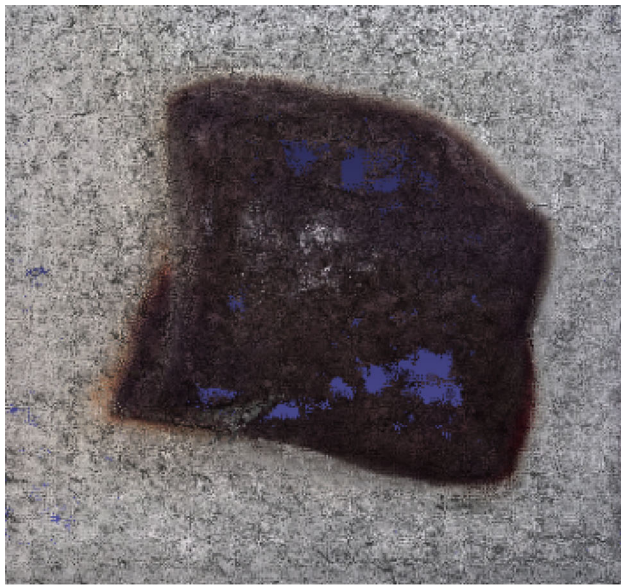


**(g)** down RCM of the robot

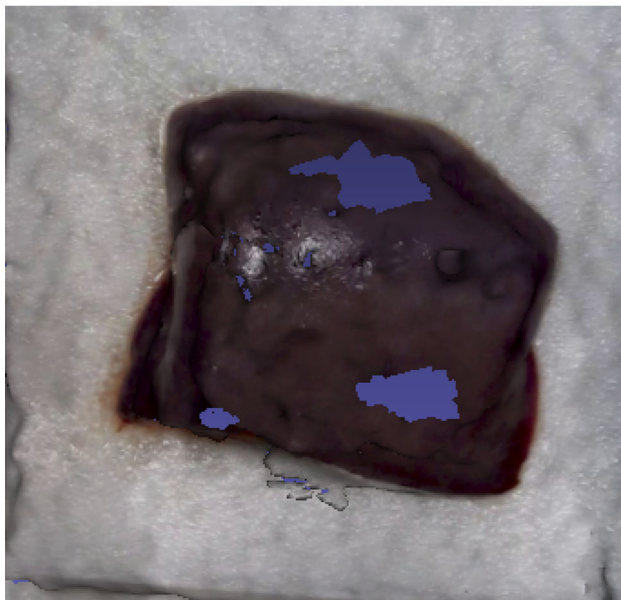


**(h)** soft-tissue image captured





(a) Dense point cloud



(b) Texture model

Fig. 14 Reconstruction result in the model of robot

**Acknowledgements** This work is supported by the National Natural Science Foundation of China (62073099, U2013208) and Shenzhen Fundamental Research Grant (JCYJ20180507183456108, GXWD20201230155427003-20200822100405001).

## Declarations

**Conflict of interest** The authors declare that they have no conflicts of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adap-

tation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Qi J, Xu K, Ding X (2021) Approach to hand posture recognition based on hand shape features for human-robot interaction. *Complex Intell Syst* 7, <https://doi.org/10.1007/s40747-021-00333-w>
2. Deguchi K, Sasano T, Arai H, YOSHIKAWA H (1996) 3-d shape reconstruction from endoscope image sequences by the factorization method. *IEICE Trans Inf Syst* 79(9):1329–1336
3. Burschka D, Li M, Ishii M, Taylor RH, Hager GD (2005) Scale-invariant registration of monocular endoscopic images to CT-scans for sinus surgery. *Med Image Anal* 9(5):413–426
4. Furukawa Y, Ponce J (2009) Accurate, dense, and robust multiview stereopsis. *IEEE Trans Pattern Anal Mach Intell* 32(8):1362–1376
5. Goesele M, Snavely N, Curless B, Hoppe H, Seitz SM (2017) Multi-view stereo for community photo collections. In: 2017 IEEE 11th International Conference on Computer Vision. IEEE, pp 1–8, <https://doi.org/10.1109/ICCV.2017.4408933>
6. Boissonnat J-D, Dyer R, Ghosh A (2018) Delaunay triangulation of manifolds. *Found Comput Math* 18(2):399–431
7. Kazhdan M, Hoppe H (2013) Screened poisson surface reconstruction. *ACM Trans Graph (ToG)* 32(3):29, pp1–13
8. Fuhrmann S, Goesele M (2014) Floating scale surface reconstruction. *ACM Trans Graph (ToG)* 33(4):46, pp1–11
9. Lempitsky V, Ivanov D (2007) Seamless mosaicing of image-based texture maps. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp 1–6, <https://doi.org/10.1109/CVPR.2007.383078>
10. Pérez P, Gangnet M, Blake A (2003) Poisson image editing. *ACM Trans Graph (TOG)* 22(3):313–318
11. Waechter M, Moehrl N, Goesele M (2014) Let there be color! large-scale texturing of 3d reconstructions. In: European Conference on Computer Vision–ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8693. Springer, Cham. [https://doi.org/10.1007/978-3-319-10602-1\\_54](https://doi.org/10.1007/978-3-319-10602-1_54)
12. Fuhrmann S, Langguth F, Goesele M (2014) MVE-An image-based reconstruction environment. *Comput Graph* 53:44–53
13. Liu W, Li F, Jing C, Wan Y, Su B (2020) Recognition and location of typical automotive parts based on the RGB-D camera. *Complex Intell Syst*, <https://doi.org/10.1007/s40747-020-00182-z>
14. Wu C (2013) Towards linear-time incremental structure from motion. In: 2013 International Conference on 3D Vision-3DV2013., pp. 127–134, <https://doi.org/10.1109/3DV.2013.25>
15. Cheng J, Leng C, Wu J, Cui H, Lu H (2014) Fast and accurate image matching with cascade hashing for 3d reconstruction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8, <https://doi.org/10.1109/CVPR.2014.8>
16. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60: pp 91–110
17. Schonberger JL, Frahm JM (2016) Structure-from-motion revisited. In: IEEE Conference on Computer Vision & Pattern Recognition, pp 4104–4113, <https://doi.org/10.1109/CVPR.2016.445>

18. Zhu S, Zhang R, Lei Z, Shen T, Long Q (2018) Very large-scale global SFM by distributed motion averaging. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4568–4577, <https://doi.org/10.1109/CVPR.2018.00480>
19. Kazhdan MM, Klein A, Dalal K, Hoppe H (2007) Unconstrained isosurface extraction on arbitrary octrees. In: Eurographics Symposium on Geometry Processing (SGP '07). Eurographics Assoc, Goslar, DEU, 125–133
20. Vineet V, Narayanan PJ (2010) Solving Multilabel MRFs Using Incremental  $\alpha$ -Expansion on the GPUs. In: Zha H, Taniguchi R, Maybank S. (eds) Computer Vision –ACCV 2009. ACCV 2009. Lecture Notes in Computer Science, vol 5996. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-12297-2\\_61](https://doi.org/10.1007/978-3-642-12297-2_61)
21. Polizzi E, Kestyn J (2012) Feast eigenvalue solver v3.0 user guide. arXiv preprint [arXiv:1203.4031](https://arxiv.org/abs/1203.4031)
22. Yip HM, Li P, Navarro-Alarcon D, Wang Z, Liu Y-h (2014) A new circular-guided remote center of motion mechanism for assistive surgical robots. In: 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014). IEEE, pp 217–222, <https://doi.org/10.1109/ROBIO.2014.7090333>
23. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Bottou L, and Weinberger KQ (eds) Advances in Neural Information Processing Systems, vol. 25. Curran Associates, Inc., <https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.