Automated Dissection of Intact Single Cell from Tissue Using Robotic Micromanipulation System

Youchao Zhang, Xiangyu Guo, Qingyu Wang, Fanghao Wang, Chuanjie Liu
Mingchuan Zhou and Yibin Ying

***Abstract*—Obtaining single cell from tissues is important for intersection of information and bioscience research. In this article, a robotic framework based on micromanipulation system has been proposed, which can automatically and intelligently cut down single cells intact from tissue sections. The proposed method consists of several steps. An attention mechanism improved (AMI) tip localization neural network is proposed to detect and track the needle tip of micro-scale displacement end-effector within the limited field of view under microscopy. Then, the transformation matrix between the camera and coordinate system of the robot is calculated. And the cutting trajectory is generated and optimized. Finally, the end-effector is controlled to obtain intact single cells from tissues by model predictive control (MPC). The performance of the framework is verified in paraffin tissue sections dissection experiment which shows proposed framework is robust and precise enough to obtain an intact single cell. The error of autonomous single cell dissection is no more than 0.61 μm.**

***Index Terms*—Cell micromanipulation; Hand-eye calibration; Deep learning; Computer vision for robotic micromanipulation; Automation.**

I. Introduction

INFORMATION and biology intersection research domain such as bioinformatics, genome informatics analysis, and proteome informatics has been limited by the unavailability of intact single cells from tissues. Cells are only a few tens of microns in diameter, which makes it impossible to be cut by free hand due to hand tremors. The general purpose of tissue dissection is to remove any unnecessary portions of the tissue section and then extract DNA, RNA or protein from the desired area to enrich the molecule of interest [1].

The current common tissue cutting technique is laser capture microdissection (LCM) [2]. LCM is a popular form of tissue dissection in bioinformatic research. The laser is used to remove the region of interest (ROI) from tissue sections under microscopic guidance. The LCM technique is limited by the spot size of the laser (7.5-30μm) [3], which makes it difficult to achieve a small enough resolution to obtain single cells. The laser may denature the material and contaminate the genetic material when burning the tissue. LCM still involves specialized instruments and specific consumables, making it prohibitively expensive for most clinical applications.

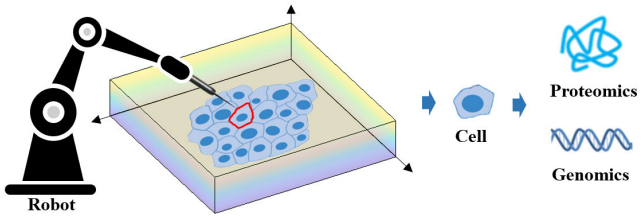

Fig. 1. Intact contamination-free single cells were obtained by cutting along the generated trajectory under the microscope with the end-effector controlled by robotic micro-manipulation techniques. The obtained single cells are further used in the intersection of informatics and biology such as genomics, proteomics, etc.

Robot is able to operate at room temperature and can achieve micron-level accuracy [4], [5], so it is expected that the disadvantages of LCM can be overcome by robotic micromanipulation. As shown in Fig. 1, the intact, uncontaminated single cells acquired by the robot can facilitate the intersection of informatics and biological research fields such as genomics and proteomics. Some robotic systems have been developed to perform tissue cutting for food processing [6], and clinical treatment [7]. These robotic systems were developed for tissue-level cutting at the macro scale, compared to which the separation of target cells from tissue at the micron scale is more challenging. Due to space limitation, there is a lack of robotic frameworks for cutting single cell out of tissues using micromanipulation.

Several micromanipulation robotic systems capable of manipulation of cells have been reported [8]–[10]. The identification, hand-eye calibration, and motion control of the end-effector are the keys to achieving precise operation. Zhang et al. developed a micro robotic system that uses a visual servo to estimate and compensate for the motion of the sperm tail [11]. The visual servo controller achieved a positioning accuracy of 1.7 μm. Gong et al. proposed an automated cell rotation method based on real-time detection and tracking of the polar body [12]. The micro robotic system uses a micropipette holding the object cell and an injection needle to contact the cell. The system achieved 98% success rate of out-of-plane rotation and 93% success rate of in-plane rotation. Dai et al. proposed a robotic manipulation system to control

Manuscript received: February, 9, 2023; Revised April, 20, 2023; Accepted June, 8, 2023.

This paper was recommended for publication by Editor Xinyu Liu upon evaluation of the Associate Editor and Reviewers' comments. This study was supported in part by Zhejiang Provincial Science and Technology Plan Project with project number 2023C02007. *(Corresponding author: Yibin Ying.)*

Youchao Zhang, Xiangyu Guo, Qingyu Wang, Fanghao Wang, Chuanjie Liu, Mingchuan Zhou and Yibin Ying are with the Key Team of Intelligent Bioindustry Innovation Team, Department of Biosystems Engineering, Zhejiang University, Hangzhou 310058, China. {youchaozhang, 12113031, 12013027, 12213026, chuanjieliu, mczhou, ybying} @zju.edu.cn

Digital Object Identifier (DOI): see top of this page.

the orientation of deformable cells [13]. The maximum deformation of the cell during the whole orientation control was 2.70 μm. Dong et al developed a robotic system for automated microinjection of nematodes [14]. The system identified the needle tip by Otsu adaptive thresholding and calibrated by linear regression, ultimately achieving a 78.8% operational success rate. Recently, a robotic system was proposed by Shan et al. for blastocyst biopsy [15]. The system achieved a localization error of 1.1 $\pm$ 0.1 μm by improving Unet for identification and localization. However, most of the existing cell manipulation robots manipulate cells in clear solution and cells are typically held in place by micropipette [4]. Only the end-effector and the target cell are in the field of view, object identification is of feasibility and consistency. The tissue sections under the microscope have complex textures, which can cause greater interference with the identification and positioning of the end-effector. In addition, the trajectory planning and hand-eye calibration of the end-effector also need to be further optimized to improve the control accuracy of the end-effector.

There are two main challenges. The identification of end-effectors tip in the field of view of a microscope could be further improved. Morphological methods are susceptible to color changes in tissue sections and are not accurate and robust enough [15]. Existing keypoint identification algorithms (regression method and heatmap method) rely only on local information of the keypoint, making it difficult to accurately track the keypoint [16], [17]. Hand-eye calibration is an important preparation for robot manipulation [18]. It is usually necessary to place a calibration plate in the field of view or a marker on the end-effector for calibration. However, it is difficult to perform calibration operations in the microscope field of view due to space constraints [19], [20]. Therefore, a marker free calibration method needs to be developed.

The main contributions of this letter are summarized as follows: i) A robotic framework based on micromanipulation system has been proposed, which can automatically and intelligently cut down single cells intact from tissue sections. ii) An attention mechanism improved (AMI) end-effector end point detection model has been designed, which combines the global information provided by the segmentation algorithm with the object attention loss. AMI achieves higher accuracy in keypoint detection of end-effector. iii) A marker free hand-eye calibration framework of cell micromanipulation has been presented and experiment shows that the path following error is no more than 0.61 μm.

In section II, we will describe the setup of the cell manipulation robot system. The main framework with specific methods will be presented in Section III. In Section IV, we describe how the experiment was designed and experimental results are discussed. Finally, in Section V, the paper is summarized.

II. System Overview

As shown in Fig. 2(a), the system consists of an inverted microscope, IXplore Standard (OLYMPUS, Japan) equipped with an XY motorized carrier stage and two commercially available microscope-manipulation robotic arms, TransferMan® 4m (EPPENDORF, Germany) with three degrees of freedom. The maximum magnification of the microscope is 400x through the eyepiece and objective lens. The robotic arms have a motion resolution of 0.05 µm. A camera, W160 (BGimaging, China) is attached to the inverted microscope to capture image information at a rate of 60 frames per second. As shown in Fig. 2(b), the tissue sections were fixed on the carrier table. The section is specially treated with a layer of paraffin wax between the tissue and the slide to ensure that the cells can be shed smoothly after being cut. The end-effector is a titanium steel microneedle with a tip of 0.1 μm which is significantly smaller than a typical cell size of 20-50 μm [21]. The end-effector has a certain degree of stiffness and toughness, which ensures that it has no deformation when performing cell cutting.

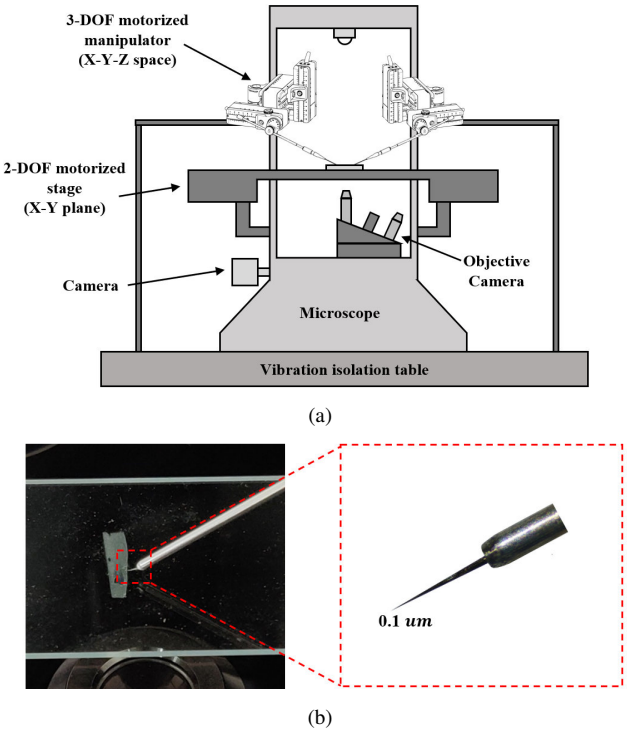

Fig. 2. System setup. (a) Hardware setup diagram of the whole system. (b) Detailed diagram of slicing and actuator placement with end-effector.

To ensure the performance of the system, we integrated the software and control part of the system into the Robot Operating System (ROS) Melodic, which also provided great convenience for our development. ROS runs on a desktop computer with Intel Core i9-10980Xe CPU and NVIDIA TAITAN Xp in Ubuntu 18.04.

III. Methods

The proposed framework is shown in Fig. 3. The calibration parameters of the camera are acquired in advance to eliminate distortions caused by the imaging device. After moving the end-effector to the plane of focus, we move the needle on a trajectory containing two orthogonal directions through robotically controlled microscale displacements. After each movement, the needle tip is identified by keypoint detection. After completing all the motions of the trajectory definition, we compute the transformation between the camera coordinate

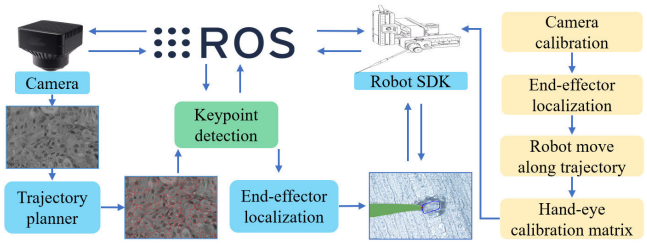

Fig. 3. The framework of the proposed framework

system and the robot coordinate system. We further plan the trajectory of the end-effector based on the segmentation results generated by cell segmentation. Finally, we design a robust controller to control the end-effector cutting along with the reference. The details of each step are shown below.

A. Detect and Track End-effector

Essentially what needs to be performed is the detection of end-effector endpoints. When traditional machine vision method is often affected by environmental changes. Therefore, we use deep learning to improve the robustness of keypoint detection.

In deep learning domain, for the keypoint detection task, there are two ways to represent the true value in the ground truth [22]. (1)Regression method: The coordinates of the target point are directly used as input for supervision so that the network is a regression network (the network ends with a fully connected layer) and the network output should be the coordinates of the target point (x, y) [16], i.e., 2 output neurons. If k points are detected at the same time, that is $2k$ output neurons. Fig. 4(b) shows that regression-based keypoint detection network (exemplified by Mask R-CNN) just outputs the coordinates of the target point through supervised learning of individual pixels at the target point. (2)Headmap method: Add two-dimensional Gaussian function to generate heatmap [17] at the location of ground truth coordinates. Use a heatmap to represent the target point to supervise. The output of the network is a feature map of the same size as the input, with 1 at the location of the detection point and 0 at the other locations. Fig. 4(c) shows that the heatmap-based keypoint detection network (exemplified by HRnet) encodes keypoint through two-dimensional Gaussian distribution. These algorithms both use local information of keypoints, rather than overall information of the object being detected and tend to produce large deviations in scenarios where accuracy is critical [23].

In order to solve the problem of keypoint drift and improve the accuracy of end-effector localization, an attention mechanism improved (AMI) end-effector end point detection model has been designed, which combines the global information provided by the segmentation algorithm with the object attention loss.

As depicted in Fig. 4(a), a mask will be split along the width direction and an object attention tensor with varying weights will match the mask and ground truth. The weight increases along width direction because we want to focus more concentration on the tip of the end-effector. Then an object attention loss will be calculated by weights multiplied by cross entropy loss along the width direction.

$$\hat{y}_{i,j} = Model(I_{i,j}) \tag{1}$$

$$H(y, \hat{y}) = -(y \cdot \log \hat{y} + (1 - y) \cdot \log (1 - \hat{y})) \tag{2}$$

where I is the image data from microscope, $H(y, \hat{y})$ is cross entropy loss. $\hat{y}$ is the probability that the model predicts the pixel to be an end-effector, and y is the label value that takes the value of 1 if the pixel belongs to the end-effector and 0 otherwise. The attention loss of the same mask $\mathcal{L}_a$ in different areas can be calculated as:

$$\mathcal{L}_a[i] = \boldsymbol{w_i} H(mask[i], label[i]) \tag{3}$$

where w_i is the weight of the predicted mask in the width direction. Which can be calculated as:

$$\boldsymbol{w_i} = \frac{2i}{l(l+1)} \tag{4}$$

where the l is the length of the mask in the width direction. The sum of the weights satisfies the normalization in the direction of the mask width:

$$\begin{aligned} \sum_{i=1}^{l} \boldsymbol{w_i} &= \frac{2}{l(l+1)}(1 + 2 + ... + l) \\ &= \frac{2}{l(l+1)} \frac{l(l+1)}{2} = 1 \end{aligned} \tag{5}$$

When the training is completed, the segmentation result of the end-effector has been acquired, which results in a binarized mask. Integer programming can be used to infer the location of the tip based on the region of the mask.

$$\begin{aligned} &(x, y) = \arg\max_{u,v} v \\ &s.t. \quad \begin{cases} mask[u][v] = 1 \\ u = 0, 1, 2, ..., h \\ v = 0, 1, 2, ..., l \end{cases} \end{aligned} \tag{6}$$

B. Hand-eye Calibration

Most hand-eye calibration methods require separate measurements and additional equipment (calibration grids, markers, etc.). However, in cell manipulation scenarios, it is impractical to place calibration objects or markers inside the microscope field of view. The marker free eye-to-hand hand-eye calibration framework is divided into two steps, first tracking the trajectory of the end-effector by the proposed detection algorithm. Then the transfer matrix between the trajectory in the image and the real trajectory in space is calculated by the calibration algorithm.

The hand-eye calibration is usually defined by solving a homogeneous matrix equation in the form $AX = ZB$, where X and Z are rigid transformations defined by the configuration used. [19], [24] In this system ,we have $A' = AX$. So the homogeneous matrix equations can be formulated as follows:

$$A' = \left[\begin{array}{c|c} R & T \end{array} \right] \times B \tag{7}$$

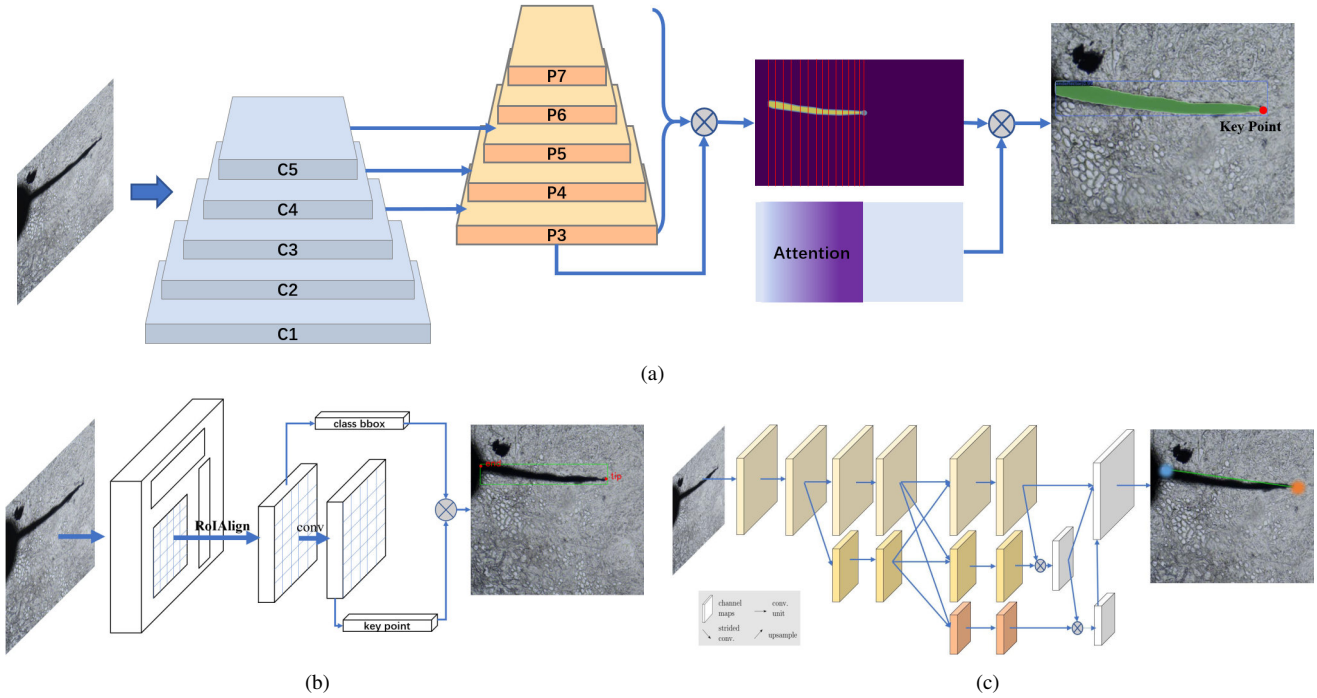

Fig. 4. The detected and tracked trajectory of the end-effector. (a) Attention-based keypoint detection. (b) Regression-based (Mask R-CNN) keypoint detection. (c) Heatmap-based (HRnet) keypoint detection.

where $\{A' = A'_i, ..., A'_j\}$ is the end-effector position in the robot coordinate system and $\{B = B_i, ..., B_j\}$ is the corresponding end-effector position in the camera coordinate system. $R \in SO_3$ and $T \in \mathbb{R}^3$ are the rotation matrix and translation vector that transforms the end-effector position from the camera coordinate system to the robot coordinate system.

In order to solve the transformation matrix, two solution methods have been used. The first method is based on singular value decomposition proposed by Arun et al [25]. The second method is Similarity Transformation(SIM3) algorithm based on the unit quaternion operation proposed by Horn et al [26]. The algorithm uses unit quaternions instead of matrices to express rotations and thus computes a more accurate closed-form solution.

In order to obtain a more accurate trajectory of the end-effector motion from the image, 2D linear Kalman filter (KF) [27] has been used to reduce the noise in the signal. The KF is able to estimate the state of the system from measurements containing noise, which in turn can predict the future state of the system, so it is useful for needle tip position detection and localization. Tip state is $P = [x, y, v_x, v_y]^T$. (x, y) and (v_x, v_y) is the position and velocity separately. The KF process includes a prediction and a correction step to obtain an optimal estimation of P. The mean and covariance estimations of P after propagation (prediction) from step $t-1$ to step t are denoted by $(\hat{P}_{t|t-1}, \hat{M}_{t|t-1})$, and the mean and covariance estimations of P after a keypoint detection measurement update (correction) at step t are denoted by $(\hat{P}_{t|t}, \hat{M}_{t|t})$. And (x, y) in $\hat{P}_{t|t}$ is an improved estimation for tip detection.

Prediction for P:

$$\begin{aligned} \hat{P}_{t|t-1} &= F\hat{P}_{t-1|t-1} \\ \hat{M}_{t|t-1} &= F\hat{M}_{t-1|t-1}F^T + Q_{t-1} \end{aligned} \tag{8}$$

Correction for P :

$$\begin{aligned} K_t &= \hat{M}_{t|t-1}H^T(H\hat{M}_{t|t-1}H^T + R_t) \\ \hat{P}_{t|t} &= \hat{P}_{t|t-1} + K_t(Z_t - H_t\hat{P}_{t|t-1}) \\ \hat{M}_{t|t} &= (I - K_tH)\hat{M}_{t|t-1} \end{aligned} \tag{9}$$

where Q and R are the covariance matrix of state function Gaussian noise and tip detection Gaussian noise. F and H are state transformation matrix and observation matrix.

$$F = \begin{bmatrix} 1 & 0 & d_t & 0 \\ 0 & 1 & 0 & d_t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{10}$$

C. End-effector trajectory planning

A critical step is to identify the cells to be manipulated and plan the trajectory of the end-effector movement. Since the field of view of a microscope is large relative to that of a tissue section, the images often contain multiple cells at the same time, making it challenging to plan motion paths for multiple cells at the same time. Cell segmentation has been studied by many researchers. The instance segmentation method used in this paper is Cellpose [21], which is a generalist algorithm for cellular segmentation. The average accuracy of Cellpose is 0.91 at intersection over union metric threshold of 0.5. The trajectory of the end-effector can be further acquired based on the segmentation results generated by cell segmentation.

Firstly, we need to calculate the connected regions of different single cells based on the segmentation results of the

model, and then calculate the outer contour of every single cell based on the connected regions. After the outer contour is obtained, the next part of the process is performed to eliminate some unsuitable contours. Finally, the cutting trajectory of every single cell is obtained.

After getting a rough trajectory, it should be further optimized. Firstly, for every single cell, the corresponding trajectory must be the one that encloses the single cell completely. Secondly, since a single cell cannot intersect with another single cell, the trajectories cannot overlap together. Finally, the trajectories also need to be smoothed, and the trajectories can be optimized using Bézier spline curves. The generated initial path usually has many inflection points, which is detrimental to the robot's motion. Suddenly changing the direction of motion wastes a lot of time and adds complexity to the robot control.

As shown in Fig. 5 the Bézier spline curve method is proposed to be used to smooth and optimize the initial trajectory. The initial trajectory is first sampled with uniform points to obtain $n+1$ control points $\{P_0, P_1, ..., P_n\}$ in which $P_i \in \mathbb{R}^3$, and then interpolated using the Bézier curve function. The curve is given by the following equation:

$$P(t) = \sum_{i=0}^{n} P_i B_{i,n}(t), t \in [0,1] \tag{11}$$

where $B_{i,n}$ is called Bernstein polynomial, which has the property of recursion and facilitates the recursive solution of the curve coefficients, is defined as follows:

$$B_{i,n} = C_n^i t^i (1-t)^{n-i}, i = 0, 1, ..., n \tag{12}$$

$$B_{i,n} = (1-t)B_{i,n-1}(t) + tB_{i-1,n-1}(t) \tag{13}$$

Algorithm 1 End-effector trajectory planning.

Input: M: the cell segmentation output;
Output: $T(X,Y)$: cutting trajectory;

for $mask$ in M: **do**
  $X, Y = findContours(mask)$;
  add (X,Y) into T;
end for
for mask in M: **do**
  if (X,Y) is unclosed **then**
    $Enclose(X,Y)$;
  end if
  if (X,Y) is intersecting with neighbors **then**
    $Split(X,Y)$;
  end if
  $(P_X, P_Y) = Sample(X,Y)$;
  $(X,Y) = BezierSpline(P_X, P_Y)$;
end for
return T;

It is difficult for the cell segmentation model to identify all the single cell contours altogether. In addition to generating the trajectory directly by the cell segmentation model, we can also interact with the image directly through the interface. Discrete contour points are first generated by mouse clicks along the contour of a single cell on the screen, and then a smooth path that can completely enclose the single cell is obtained according to the method of spline interpolation.

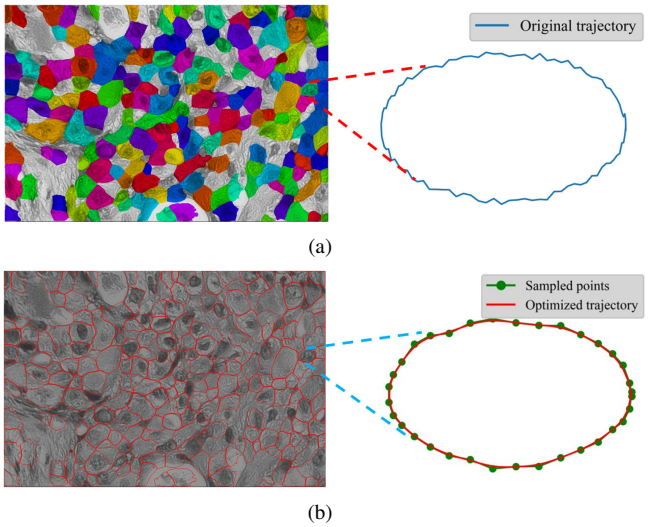

Fig. 5. Schematic diagram of trajectory optimization. (a) The result of cell instance segmentation obtained by the cell cutting model and the initial trajectory. (b) The trajectory of the end-effector is generated by the trajectory planning algorithm (red curve) and optimized by Bézier curves.

D. Controller design

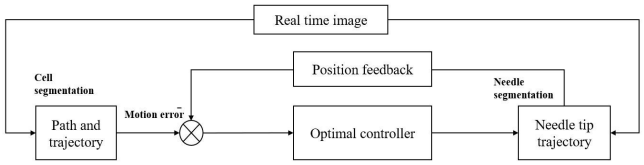

Fig. 6. Control diagram for automated end-effector cell cutting. After the trajectory planning is completed, the AMI returns the tip position information as real-time feedback, and the MPC controller controls the end-effector to cut along the trajectory.

When the cutting trajectory is acquired, a robust controller is needed to control the end-effector cutting along with the reference. For minimizing the trajectory deviation of the end-effector during cutting, as shown in Fig. 6, model predictive control (MPC) has been used. MPC calculates the sequence of control actions that minimizes the cost over the horizon. MPC applies to the robot only the first computed control action.

$$\min_{u_0,u_1,...,u_N} \sum_{k=0}^{N} \|X_k - r(k)\|^2 + \rho \Delta u_k^2$$
$$s.t. \quad \begin{cases} X_{k+1} = X_k + \Delta t u_k \\ X_{min} \le X_k \le X_{max} \\ u_{min} \le u_k \le u_{max} \end{cases} \tag{14}$$

where $X = [P_x, P_y]^T$, and $u = [v_x, v_y]^T$. X_{min} and X_{max} is the lower and upper bound of X. u_{min} and u_{max} is the lower and upper bound of u. N is the size of horizons. ρ is weight of Δu_k^2. $r(k)$ is the trajectory after optimization. The constrained optimization problem is solved by minimizing the cost function.

IV. Experiments

The regression approach, the heatmap approach, and the improved algorithm have been used to detect the keypoints in

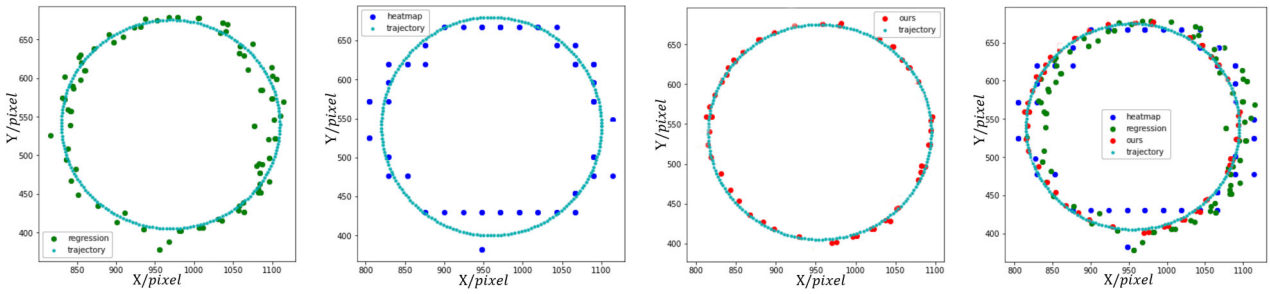

Fig. 7. The detected and tracked trajectory of end-effector, unit/pixel. (a) Result of the regression-based method. (b) Result of the heatmap-based method. (c) Result of the proposed method. (d) Comparison diagram of the results of the three methods.

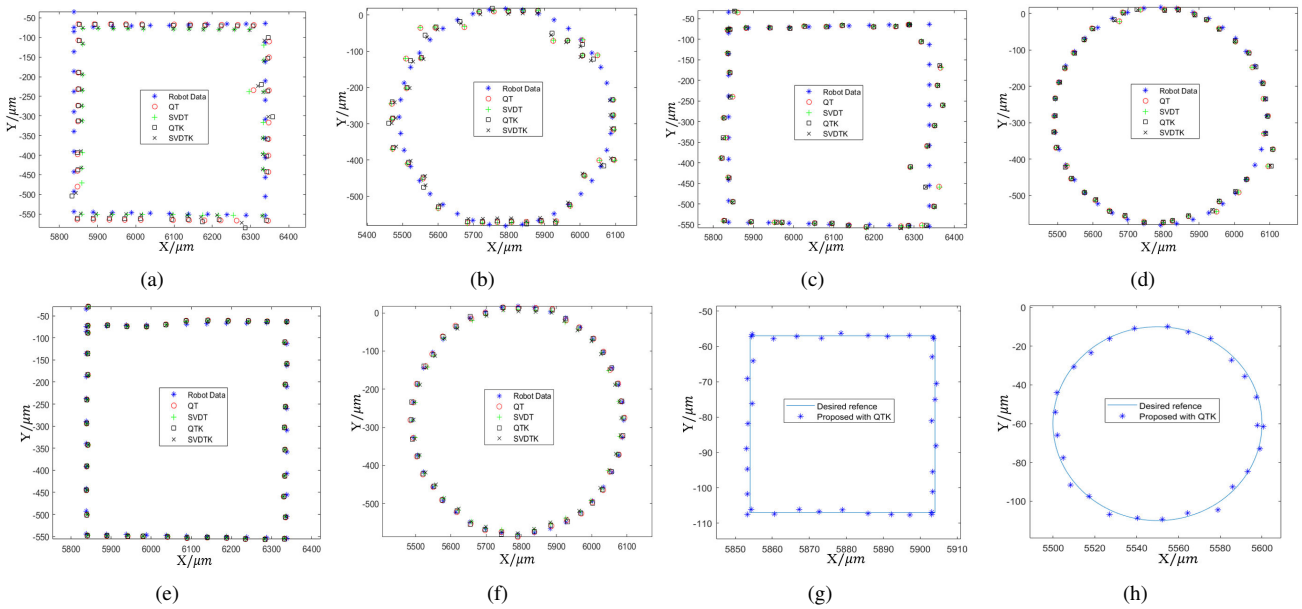

Fig. 8. Hand-eye calibration, unit/μm. Calibration of trajectory $\#1$ using regression-based method(a), heatmap-based method(c), and proposed method(e). Calibration of trajectory $\#2$ using regression-based method(b), heatmap-based method(d), and the proposed method(f). Rectangular trajectories(g) $\#3$ and circular trajectories(h) $\#4$ are with side lengths and diameters of 50 μm. The keypoints are detected by the proposed method and the transformation matrix is calculated by QTK.

a complex environment and try to have the robot perform a circular trajectory following as a way to compare the actual results of the three algorithms. The dataset includes 1293 images. The training and test dataset was split in the ratio of 8:2. The training parameters include: the number of Epoch is 500, the optimizer is Adam, the learning rate is 0.0003, and the weight decay is set to 0.0001. The test accuracy is 98.9% and recall is 96.7%. For comparison, Mask-RCNN [16] is used as a regression-based model and HRnet [28] as a heatmap-based model.

Cell tissue sections were placed on the carrier table and a robotic arm was adjusted to the field of view of the microscope simultaneously with the tissue sections. For a better comparison, two motion trajectories have been designed for the robot. The first trajectory $\#1$ is for the robot to draw a rectangle on the tissue section. The second trajectory $\#2$ is for the robot to draw a circle on the tissue section. A step size of 50 μm was set between every two points. While the robot is moving, the camera captures video of the robot's motion trajectory. To evaluate the accuracy of the proposed identification and calibration method, it has been compared with regression-based detection and heatmap-based detection.

Finally, cut down the cells intact using the autonomous micromanipulation robot and compare it with a manually controlled master-slave robot.

From Fig. 7, it can be found that the heatmap approach has a large offset error in the trajectory following, which is mainly because the heatmap approach is thought to generate keypoints in a range. The regression approach also has a large error, which is because when the test environment is complex, the regression approach is easy to over-fit and does not have strong robustness. The proposed algorithm has a small error in the circular trajectory following the process. From Fig. 7(d), the proposed method is able to better fit the circular robot motion trajectory.

As shown in Table I, the average pixel deviation of proposed method is 0.51, which is significantly lower than the 1.33 of the regression algorithm and 1.36 of the heatmap algorithm. The motion trajectory is detected using three keypoint detection algorithms, and then the transformation matrix is

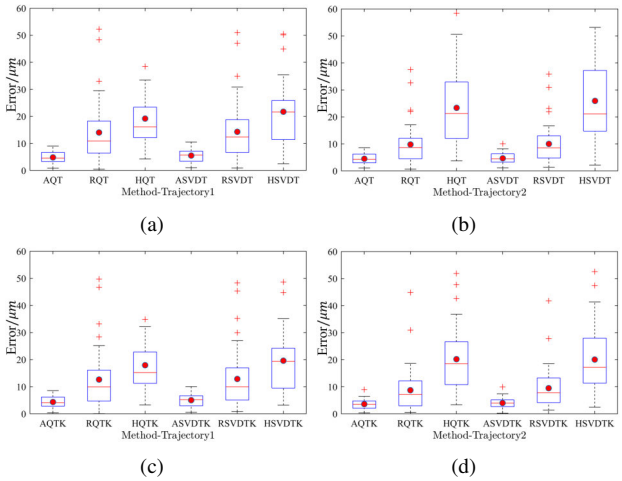

Fig. 9. The calibration error for two trajectories. AQT, RQT, HQT, ASVDT, RSVDT, and HSVDT means AttentionQT, RegressionQT, HeatmapQT, AttentionSVDT, RegressionSVDT, and HeatmapSVDT respectively. (a) Error box-and-whisker plot for the calibration of trajectory # 1 (without Kalman filter). (c) Error box-and-whisker plot for the calibration of trajectory # 1 (with Kalman filter). (b) Error box-and-whisker plot for the calibration of trajectory # 2 (without Kalman filter). (d) Error box-and-whisker plot for the calibration of trajectory # 2 (with Kalman filter).

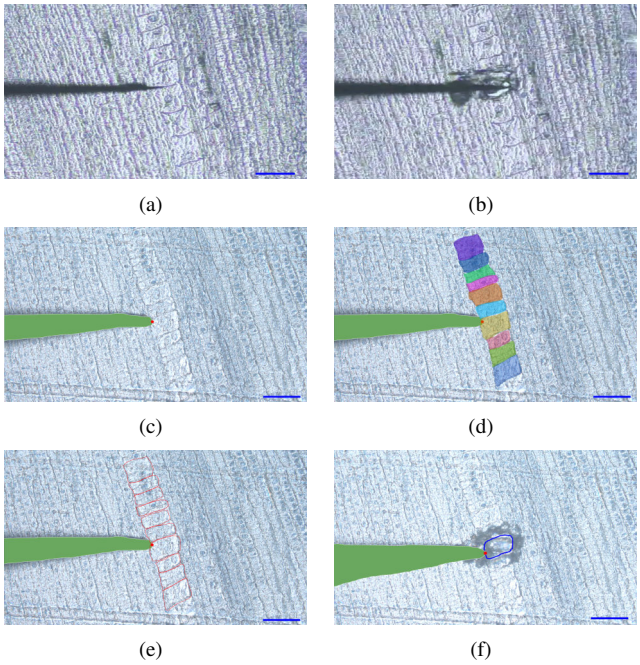

Fig. 10. Manually controlled master-slave manipulation(a, b) and Robot autonomous cell manipulation(c, d, e, f). (c) Identification of end-effector endpoints (redpoint). (d) Outline of the cell. (e) Trajectory used for cutting. (f) Cut out a single cell of the user's choice. Scale bar: 100μm.

calculated using the hand-eye correction algorithm mentioned in Section III(B). The methods described in Section III(B): singular value decomposition (SVDT), singular value decomposition with KF (SVDTK), unit quaternion method (QT), and unit quaternion method with KF (QTK). As shown in Fig. 8, comparing with regression and heatmap method, the trajectory by the proposed method matches the robot's original trajectory the best after calibration.

TABLE I
THE TRACKING PIXEL ERROR OF END-EFFECTOR

Methods	Regression	Heatmap	Proposed
error/pixels	1.33	1.36	0.51

The calibration error for a point is calculated as,

$$e_i = \sqrt{(x_i^R - x_i^{R'})^2 + (y_i^R - y_i^{R'})^2 + (z_i^R - z_i^{R'})^2} \quad (15)$$

where the (x_i^R, y_i^R, z_i^R) is the needle tip position in the robot coordinate system and $(x_i^{R'}, y_i^{R'}, z_i^{R'})$ is the needle tip position transformed from camera coordinate system. The box-and-whisker plots corresponding to the calibration errors for each method are given in Fig. 9. The maximum whisker length is specified as 1.0 times the interquartile distance. Data points outside the whiskers are considered as outliers. Whiskers show the minimum and maximum recorded errors, while the first (25%) and third (75%) quartiles are shown at the bottom and top edges of the box. The bars and red dots represent the median and mean of the error data, respectively.

TABLE II
THE AVERAGE CALIBRATION ERROR OF END-EFFECTOR

	Methods	Regression/μm	Heatmap/μm	Proposed/μm
#1	SVDT	2.81	3.12	0.65
#1	SVDTK	2.65	2.84	0.65
#1	QT	2.79	2.91	0.61
#1	QTK	2.63	2.83	0.61
#2	SVDT	1.82	3.68	0.84
#2	SVDTK	1.72	3.09	0.97
#2	QT	1.85	3.66	0.63
#2	QTK	1.81	3.04	0.60

As shown in Fig. 9, the distribution of box plots differs for the different methods. Firstly, the comparison of different keypoint detection methods is carried out to compare the keypoint detection methods based on regression and heatmap, and the proposed keypoint detection method corresponds to the smallest error with the original trajectory of the robot after the calibration of the trajectory. The minimum average error of regression-based keypoint detection is 2.63 μm for trajectory #1 and 1.72 μm for trajectory #2. Heatmap-based keypoint detection has a minimum average error of 2.83 μm for trajectory #1 and a minimum average error of 3.04 μm for trajectory #2. However, the proposed keypoint detection method has a minimum average error of 0.61 μm in trajectory #1 and 0.60 μm in trajectory #2. For the different calibration methods, the QTK method obtained the best results, slightly outperforming QT, SVDT, and SVDTK. Comparing before and after the use of the KF, it can be found that the trajectories after KF processing can achieve smaller calibration errors after calibration. More experimental results are presented in Table II.

Rectangular trajectories #3 and circular trajectories #4 with side lengths and diameters of 50 μm are shown in Fig. 8(g) and Fig. 8(h), respectively. The keypoints are detected by

the proposed method and the transformation matrix is calculated by QTK. The average calibration errors were 0.62 μm and 0.61 μm, respectively. The trajectory of the end-effector is closer to the actual size of the cell, and object detection is performed at smaller intervals, confirming the feasibility of this detection method in the proposed framework. To further test the performance of the cellular micromanipulation of the proposed robotic system, we attempted to use the robot to cut down a complete single cell on a longitudinal section. The transformation matrix is calculated by the QTK method. As shown in Fig. 10, manually controlled master-slave robot to obtain a single cell is unstable and inaccurate. Fig. 10(c) shows that it can accurately identify the tip of the needle (green) and the keypoint at the end of the needle (red). The tissue residue around the end of the needle tip did not interfere with the proposed keypoint detection method during the cell-cutting process. Fig. 10(d) shows the outline and segmentation of the cell. The optimized trajectory for robotic cutting was shown in Fig. 10(e). As shown in Fig. 10(f), a single cell of the user's choice was cut out. The cells were cut down intact and the contours of the cut were relatively intact, indicating the stability and accuracy of the robotic automated cell manipulation.

V. Conclusions

A robotic framework based on micromanipulation system has been proposed, which can automatically and intelligently cut down single cells intact from tissue sections. This method is the first to use robotics to obtain intact single cells from tissues. An attention mechanism improved (AMI) end-effector end point detection model has been designed, which achieves higher accuracy in keypoint detection of the end-effector. A marker-free hand-eye calibration framework of cell micromanipulation has been presented and the cutting trajectory is generated and optimized. The stability of automated robotic manipulation has been demonstrated by using the robot to cut a cell automatically compared to manually controlled master-slave robot. This will lead to greater possibilities for future cell micromanipulation.

References

[1] I. A. Eltoum, G. P. Siegal, and A. R. Frost, "Microdissection of histologic sections: past, present, and future," *Advances in anatomic pathology*, vol. 9, no. 5, pp. 316–322, 2002.

[2] M. R. Emmert-Buck, R. F. Bonner, P. D. Smith, R. F. Chuaqui, Z. Zhuang, S. R. Goldstein, R. A. Weiss, and L. A. Liotta, "Laser capture microdissection," *Science*, vol. 274, no. 5289, pp. 998–1001, 1996.

[3] S. Datta, L. Malhotra, R. Dickerson, S. Chaffee, C. K. Sen, and S. Roy, "Laser capture microdissection: Big data from small samples," *Histology and histopathology*, vol. 30, no. 11, p. 1255, 2015.

[4] Z. Zhang, X. Wang, J. Liu, C. Dai, and Y. Sun, "Robotic micromanipulation: Fundamentals and applications," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, no. 1, pp. 181–203, 2019.

[5] X. Guo, Y. Zhang, D. Jin, and M. Zhou, "A review of single-cell pose adjustment and puncture," *Advanced Intelligent Systems*, vol. 4, no. 11, p. 2200096, 2022.

[6] L. Han, H. Wang, Z. Liu, W. Chen, and X. Zhang, "Vision-based cutting control of deformable objects with surface tracking," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 4, pp. 2016–2026, 2020.

[7] J. H. Ma, S. Sefati, R. H. Taylor, and M. Armand, "An active steering hand-held robotic system for minimally invasive orthopaedic surgery using a continuum manipulator," *IEEE robotics and automation letters*, vol. 6, no. 2, pp. 1622–1629, 2021.

[8] F. Pan, Y. Jiao, S. Chen, L. Xing, and D. Sun, "Deep learning-enhanced dual-module large-throughput microinjection system for adherent cells," *IEEE Transactions on Automation Science and Engineering*, 2022.

[9] M. Sun, Y. Yao, X. Zhao, L. Li, H. Gong, J. Qiu, Y. Liu, and X. Zhao, "Precise aspiration and positioning control based on dynamic model inside and outside the micropipette," *IEEE Transactions on Automation Science and Engineering*, 2022.

[10] C. Dai, S. Zhuang, G. Shan, H. Liu, Y. Wang, C. Ru, and Y. Sun, "Automated piezo-assisted sperm immobilization," *IEEE Transactions on Automation Science and Engineering*, 2023.

[11] Z. Zhang, C. Dai, X. Wang, C. Ru, K. Abdalla, S. Jahangiri, C. Librach, K. Jarvi, and Y. Sun, "Automated laser ablation of motile sperm for immobilization," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 323–329, 2019.

[12] H. Gong, L. Li, J. Qiu, Y. Yao, Y. Liu, M. Cui, Q. Zhao, X. Zhao, and M. Sun, "Automatic cell rotation based on real-time detection and tracking," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7909–7916, 2021.

[13] C. Dai, Z. Zhang, Y. Lu, G. Shan, X. Wang, Q. Zhao, C. Ru, and Y. Sun, "Robotic manipulation of deformable cells for orientation control," *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 271–283, 2019.

[14] X. Dong, P. Song, and X. Liu, "Automated robotic microinjection of the nematode worm caenorhabditis elegans," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 2, pp. 850–859, 2020.

[15] G. Shan, C. Dai, H. Liu, X. Wang, W. Dou, C. Ru, Z. Zhang, and Y. Sun, "Robotic blastocyst biopsy," *IEEE/ASME Transactions on Mechatronics*, 2022.

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

[17] M. Kowalski, J. Naruniec, and T. Trzcinski, "Deep alignment network: A convolutional neural network for robust face alignment," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 88–97.

[18] S. Tanaka, Y. M. Baek, K. Harada, N. Sugita, A. Morita, S. Sora, H. Nakatomi, N. Saito, and M. Mitsuishi, "Robust forceps tracking using online calibration of hand-eye coordination for microsurgical robotic system," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 3529–3535.

[19] M. Zhou, M. Hamad, J. Weiss, A. Eslami, K. Huang, M. Maier, C. P. Lohmann, N. Navab, A. Knoll, and M. A. Nasseri, "Towards robotic eye surgery: Marker-free, online hand-eye calibration using optical coherence tomography images," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3944–3951, 2018.

[20] L. Yang, K. Youcef-Toumi, and U.-X. Tan, "Towards automatic robot-assisted microscopy: An uncalibrated approach for robotic vision-guided micromanipulation," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 5527–5532.

[21] C. Stringer, T. Wang, M. Michaelos, and M. Pachitariu, "Cellpose: a generalist algorithm for cellular segmentation," *Nature methods*, vol. 18, no. 1, pp. 100–106, 2021.

[22] M. Zhou, X. Guo, M. Grimm, E. Lochner, Z. Jiang, A. Eslami, J. Ye, N. Navab, A. Knoll, and M. A. Nasseri, "Needle detection and localisation for robot-assisted subretinal injection using deep learning," *CAAI Transactions on Intelligence Technology*, 2023.

[23] H. Jin, S. Liao, and L. Shao, "Pixel-in-pixel net: Towards efficient facial landmark detection in the wild," *International Journal of Computer Vision*, Sep 2021. [Online]. Available: http://dx.doi.org/10.1007/s11263-021-01521-4

[24] M. Zhou, Q. Yu, K. Huang, S. Mahov, A. Eslami, M. Maier, C. P. Lohmann, N. Navab, D. Zapp, A. Knoll *et al.*, "Towards robotic-assisted subretinal injection: A hybrid parallel–serial robot system design and preliminary evaluation," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 8, pp. 6617–6628, 2019.

[25] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on pattern analysis and machine intelligence*, no. 5, pp. 698–700, 1987.

[26] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Josa a*, vol. 4, no. 4, pp. 629–642, 1987.

[27] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

[28] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *CVPR*, 2019.