2020 8th IEEE International Conference on Biomedical
Robotics and Biomechatronics (BioRob)
New York, USA. Nov 29 - Dec 1, 2020

Toward Synergic Learning for Autonomous Manipulation of Deformable Tissues via Surgical Robots: An Approximate Q-Learning Approach

Sahba Aghajani Pedram[†,1], Peter Walker Ferguson[†,1], Changyeob Shin[1],
Ankur Mehta[2], Erik P. Dutson[3], Farshid Alambeigi[4], and Jacob Rosen[1]

Abstract— **In this paper, we present a synergic learning algorithm to address the task of indirect manipulation of an unknown deformable tissue. Tissue manipulation is a common yet challenging task in various surgical interventions, which makes it a good candidate for robotic automation. We propose using a linear approximate Q-learning method in which human knowledge contributes to selecting useful yet simple features of tissue manipulation while the algorithm learns to take optimal actions and accomplish the task. The algorithm is implemented and evaluated on a simulation using the OpenCV and CHAI3D libraries. Successful simulation results for four different configurations which are based on realistic tissue manipulation scenarios are presented. Results indicate that with a careful selection of relatively simple and intuitive features, the developed Q-learning algorithm can successfully learn an optimal policy without any prior knowledge of tissue dynamics or camera intrinsic/extrinsic calibration parameters.**

I. Introduction

Robot-Assisted Surgery (RAS) is becoming the norm of many operating room procedures, as it enables enhanced precision, dexterity, and feedback. Compared to conventional minimally invasive surgery (MIS) and/or open surgery, however, RAS restricts paramount information such as haptic feedback and direct vision from the surgeons. These deficiencies make surgical sub-tasks such as suturing or tissue manipulation very challenging for, and demand high cognitive loads from, surgeons [1]. To reduce the surgeon's burden, recent work has begun on automation of specific MIS tasks including suturing [2], [3], [4], tissue manipulation [5], [6], tissue dissection [7], and drilling [8], [9]. Robotic automation of these tasks would place surgeons in a supervisory position and could reduce the physical and cognitional strain of the operation. Hence, robotic automation has great potential to improve surgery, and therefore patients' outcome.

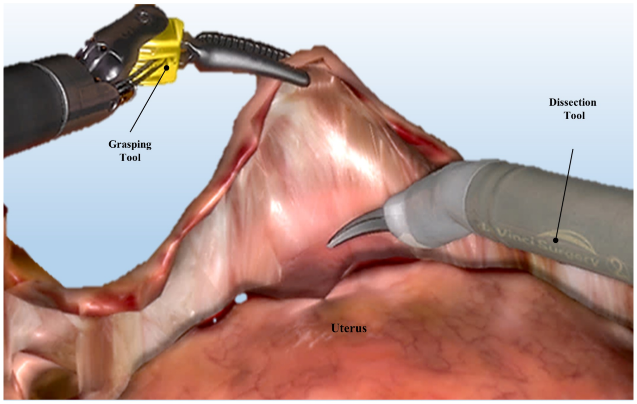


Fig. 1. Example of an MIS tissue manipulation during a tissue debridement procedure.

A particular task of interest in various MIS interventions (e.g. tumor dissection and tissue debridement) is tissue manipulation, as shown in Fig. 1. Due to the uncertain deformation behavior of tissue, however, this manipulation task is challenging even for expert clinicians. The main sources of this complication are homogeneous/heterogeneous physical or geometrical properties of deformable tissue such as stiffness and viscoelasticity which are unknown and hard-to-model [10]. Recent advances in machine/robot learning fields have enabled them to learn very complicated mappings and/or tasks [11], [12]. Hence, in order to lessen the surgeon's burden during RAS, learning-based autonomous tissue manipulation strategies are reasonable paths to explore.

From a clinical perspective, tissue manipulation tasks in MIS can be defined in various ways. This includes placing some points on the tissue to desired locations, or applying force to achieve and maintain certain tension of specific tissues. As a first step, we select the former method and define three sets of points: a grasp location, a point on the tissue to be moved, and a desired location of the point on the tissue. In general, it cannot be assumed that any of these points of interest initially coincide. Of specific interest is that the grasp location may not be the same as the point on the tissue to be moved. This scenario is the case of 'indirect' tissue manipulation, where the point of clinical interest to be moved is inconvenient or impossible to grasp. Such cases frequently arise in surgery, such as in a suturing task where the insertion point of the needle must be placed at a desired location but it cannot be grabbed directly as it would prevent needle insertion.

[†] Sahba Aghajani Pedram and Peter Walker Ferguson contributed equally to this work.

*Research supported by National Science Foundation award IIS-1227184: Multilateral Manipulation by Human-Robot.

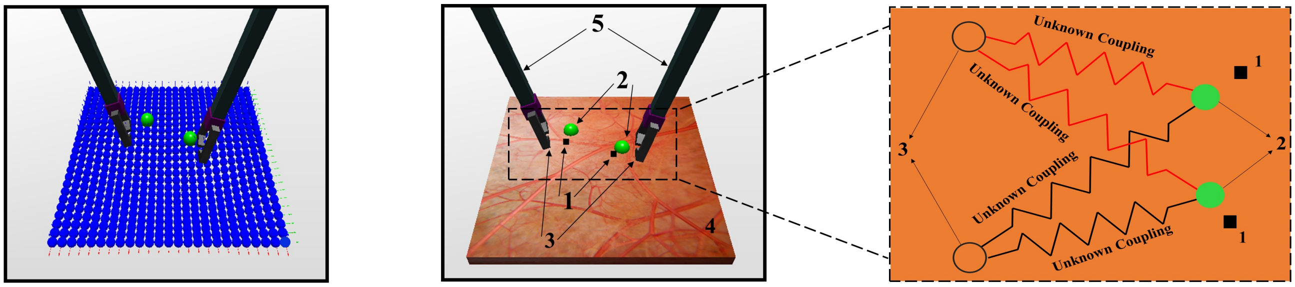
[1] Sahba Aghajani Pedram, Peter Walker Ferguson, Changyeob Shin, and Jacob Rosen are with the Mechanical and Aerospace Engineering Department, University of California at Los Angeles (UCLA), Los Angeles, CA, USA {sahbaap,pwferguson,shinhujune,jacobrosen} @ucla.edu

[2] Ankur Mehta is with the Electrical and Computer Engineering Department, University of California, Los Angeles (UCLA), CA, USA mehtank@ucla.edu

[3] Erik Dutson is with the Department of Surgery, David Geffen School of Medicine, University of California at Los Angeles (UCLA), Los Angeles, CA 90095, USA EDutson@mednet.ucla.edu

Farshid Alambeigi is with the Department of Mechanical Engineering, University of Texas at Austin, Austin, TX, USA, 78712. farshid.alambeigi@austin.utexas.edu

978-1-7281-5907-2/20/$31.00 ©2020 IEEE



(a) (b) (c)

Fig. 2. Simulation Environment. (a) All nodes used in the gel simulation (blue) with TTP highlighted (green). (b) Components of interest: 1) IDPs, 2) TTPs, 3) TGPs, 4) simulated tissue, and 5) graspers. (c) Representation of intuitive coupling between TTPs and actions of TGPs (1, 2, and 3 are as defined).

Several researchers have attempted different methods to address the task of deformable object/tissue manipulation. These methods can be classified into two broad categories based on whether they deploy human knowledge about physics of the problem or purely rely on the machine learning capabilities. The first category, which we refer to as *human-knowledge driven* algorithms, incorporate explicit human knowledge about the model of the tissue and/or manipulation task. For example, indirect positioning of a deformable object based on a linearized second order model was proposed in [13]. Reference [14] comprehensively reviews manipulation of deformable objects where algorithms deployed human knowledge in their methodologies explicitly. The second category of methods, which we refer to as *machine-knowledge driven* or *data driven* algorithms, encompass algorithms that machines deploy data to build knowledge of the tissue and/or manipulation task. For example, an optimization framework with a an online Jacobian update was used to learn the dynamics of the tissue manipulation task [5].

Due to unknown or large inter- and intra-subject variability between biological tissues in complex surgical scenarios, data driven approaches may be viewed as a more appropriate solution. However, these methods (i.e. [15], [16]) do not incorporate prior human knowledge into their solution, and thus may require large amounts of data to train which is both expensive and time-consuming [17]. For real-world robotics applications, methods of choice are data-driven algorithms which use more suitable and task-specific representations [18], [19], [20]. Hence, in the case of robotic tissue manipulation, a synergic learning (SL) algorithm which combines basic human knowledge about the physics of the problem with a learning algorithm is of interest.

To explore the performance of a SL algorithm in autonomous manipulation of deformable tissues, we propose an approximate Q-learning algorithm. For this algorithm, human knowledge contributes to selecting useful yet simple features of tissue manipulation while the algorithm learns to take optimal actions and accomplish the task. Q-learning [21], [22] is among the most popular reinforcement learning algorithms in which the agent learns to take optimal actions without explicitly building knowledge of the system dynamics and/or control. Q-learning has already been successfully applied to a range of challenging robotics problems [23].

A. Contribution

The tissue manipulation problem has already been approached by many researchers in [9], [13], [24] which have used either human-knowledge or machine-knowledge driven algorithms. In this paper, to the best of the authors' knowledge, for the first time, we propose an SL algorithm based on feature-based linear approximate Q-learning approach, which combines human knowledge with the learning algorithm to address this problem. In fact, we demonstrate that upon careful selection of simple yet effective features and reward, the developed linear approximate Q-learning algorithm is capable of accomplishing the desired manipulation task without any prior knowledge of tissue deformation behavior or camera calibration.

II. Methods

A. Problem Definition

This paper targets the task of indirect tissue manipulation with surgical robots that pinch *Tissue Grasp Points (TGPs)* to place *Tissue Target Points (TTPs)* at *Image Desired Points (IDPs)*, shown in Fig. 2b. The tissue phantom is manipulated at two TGPs representing the grasp locations of a surgical robotic system such as the Raven [25]. In operating rooms, TGPs, TTPs, and IDPs could be selected by the surgeon.

Q-learning is a popular Reinforcement Learning (RL) algorithm that attempts to find (near-) optimal policy by learning the Q values of the underlying Markov Decision Process and taking an action that maximizes Q at a given state. The most thorough form of Q-learning is to create a table of Q values for each possible state-action pair which suffers from the curse of dimensionality problem [26]. To remedy this problem, in our SL framework the Q values are approximated with a linear function of features that are input based on human knowledge of the problem.

B. Tissue Manipulation with Approximate Q-learning

As stated, the goal of the tissue manipulation task is to (indirectly) manipulate the TTPs to IDPs. Since the dynamics of the system (tissue) is not known *a priori*, this problem can be viewed as an RL problem. Moreover, while camera calibration (intrinsic and extrinsic) information may be available for a surgical robotic system, it might not be accurate or may vary when the experimental parameters are changed. Hence in this paper we assume the robot has no camera calibration information to account for real robotics surgery scenarios.

1) Approximate Q-Learning: To incorporate human knowledge into solving the RL problem as well as to avoid the curse of dimensionality, we deploy a linear function approximation approach [26], [27] to approximate the Q function values given state and action ($Q^*(s,a) \approx Q(s,a;w)$):

$$Q(s,a;w) = \sum_{i=1}^{N} w_i f_i(s,a) \tag{1}$$

where N is the number of features, w_i is the i^{th} weight, w is the vector of weights, and f_i is the i^{th} basis function.

This method allows incorporating human knowledge by handcrafting basis functions (known as*features*) to create the function approximator. Although features must be selected with domain specific knowledge, in real-world robotics applications learning algorithms with task-specific representations/knowledge are suggested [20]. In fact, in this paper we show that by carefully selecting simple yet effective features, a linear function approximator is rich enough to capture and accomplish the very complicated task of robotic manipulation of deformable tissue.

In linear approximate Q-learning, the weights, w in (1), are updated to reduce the expected value of error norm (usually 2-norm) between the unknown $Q^*(s,a)$ value and the value of the Q approximator [28], [29]:

$$L(w) = \mathbb{E}\Big[\Big(Q^*(s,a) - Q(s,a;w)\Big)^2\Big] \tag{2}$$

where

$$Q^*(s,a) = \mathbb{E}\Big[R(s,a,s') + \gamma \max_{a'} Q(s',a')\Big] \tag{3}$$

and s, a, R, γ, s', a' are the current state, action, reward, discount factor, next state, and any possible action respectively. Differentiating the loss function in (2) with respect to w, we obtain the gradient (∇) as:

$$\nabla = \mathbb{E}\Big[\Big(R(s,a,s') + \gamma \max_{a'} Q(s',a') - Q(s,a)\Big) f(s,a)\Big] \tag{4}$$

where w is then updated based on gradient decent. In (4), $f(s,a)$ is the vector of features.

2) States and Actions: The actual state of the problem contains the location of all of the nodes of the simulated tissue, which is intractable for computational purposes. Since the goal for tissue manipulation is to accurately move the TTPs to IDPs, we approximate the states of the system by only the x and y pixel coordinates of each of TTPs in image space (i.e. $s \in \mathbb{R}^4$).

Although actual surgical robot manipulators are capable of moving in three dimensions, in order to simplify the problem and enable the use of a single camera image, we constrained the motion of the TGPs to a plane parallel to and above the tissue surface. To reduce the action space dimension, an action set consisting of staying still, moving up, down, left, or right was selected for each TGP (i.e. $a \in \mathbb{R}^2$). This selection resulted in 5 possible actions for each TGP and, therefore, a total of 25 possible actions for both TGP points. Of note, all actions are defined in robot base frame as camera calibration information is assumed to be unavailable.

3) Reward: We deploy reciprocal Euclidean norm of the pixel error between IDPs and TTPs:

$$R(s,a,s') = R(s) = \frac{1}{\sqrt{e^T e + \epsilon_s}} \tag{5}$$

where $e \in \mathbb{R}^4$ is the pixel error vector between IDPs and TTPs. Note that the pixel information of TTPs is obtained with an image processing thread and ϵ_s is a small number (10^{-8}) to avoid reward divergence to infinity.

4) Features: In order to satisfactorily build a good linear approximation function for $Q^*(s,a)$, it is necessary to select a set of features that can properly distinguish what is a good action in a given state for the task of tissue manipulation. The only simple human reasoning needed to select such features is that the actions from TGPs (instruments) changes the states of TTPs, which implies unknown coupling between them (see Fig. 2c). We embed these couplings into our Q-learning framework by selecting features in the form of x or y distances between each TTP and corresponding desired points, multiplied by all permutations of actions from TGPs (represented by numeric values). For the x actions of the k^{th} ($k \in \{1,2\}$) TGP, a_{kx} is set to -1 for an action in the negative x direction, 0 for no movement in the x direction, and +1 for an action in the positive x direction. Similarly, for the y actions of the k^{th} ($k \in \{1,2\}$) TGP, a_{ky} is set to -1 for an action in the negative y direction, 0 for no movement in the y direction, and +1 for action in the positive y direction. Therefore, a total number of 16 (4 couplings × 2 relative x or y pixel positions between TTPs and IDPs × 2 TTP-IDP pairs) are selected which can be expressed in the form:

$$f_i(s,a) = (d_{kl} - t_{kl})\ a_{pq},\ i = 1,2,...,16 \tag{6}$$

where t and d represent TTP and IDP respectively, $k,p \in \{1,2\}$ indicating the k^{th} or p^{th} point, $q,l \in \{x,y\}$ indicating the coordinate or direction. These features work to consider how the action of each grasp point affect the x and y distances between each target point and corresponding desired point. However, none of these features will output the "no move" action. Intuitively, the TGPs should stop moving if the task is complete and the TTPs are very close to the IDPs. To enable this functionality, a 17^{th} feature was added that is essentially a Boolean statement to output a 1 if the reward is above a certain threshold and the "no move" action is selected. It can be formulated as:

$$f_{17}(s,a) = \begin{cases} 1, & \text{if } R(s,a=0,s') > 0.08 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

The threshold selected was 0.08, which roughly corresponds to when the target points mostly overlap the desired points in the image. Increasing or decreasing the threshold therefore correspondingly changes the amount of training needed.

5) Episodic Training: It is unlikely for random actions along a single trajectory to explore a wide variety of state-action pairs. Hence, a training episode structure was used where the state configuration is reinitialized at the beginning of each episode with a total number of N_{action} actions. This is repeated for a total number of $N_{episode}$ episodes.

6) Exploration Strategy: To properly train a Q-learning algorithm, it is necessary to find a balance between exploring unknown state-action pairs and exploiting the current optimal policy. This is known as *exploration-exploitation trade-off* in the literature [23]. Some of the standard methods that were attempted for this task include ϵ-first, ϵ-greedy, and ϵ-decreasing. The most success was an ϵ-decreasing strategy according to (8)

$$\epsilon = \max(0.1, 1 - \frac{N_{action} \times n_{episode} + n_{action}}{N_{episode} \times N_{action}}) \tag{8}$$

where n_{action} is the number of actions that have been taken in the current episode, $n_{episode}$ is the current episode number from 0 to $N_{episode} - 1$.

7) Learning Rate and Discount Factor: The learning rate, α is the step size in the gradient descent of updating weights in (4). There is a trade-off between smaller or larger values of learning rate. The larger α will enable faster convergence but has a potential for divergence while smaller α has higher chance for convergence. A common strategy for learning rate is to set it to the following form for some constant K:

$$\frac{K}{K + N_{action} \times n_{episode} + n_{action}} \tag{9}$$

In this paper, we deployed a cyclical learning rate (as proposed in [30]) in the form mentioned above. Such a learning rate oscillates between a lower and upper bound to overcome saddle points and increase convergence speed. This was chosen as it causes convergence of weights as the number of actions approaches infinity, but also periodically resets the learning rate to a higher value to overcome saddle points and correct for noise due to visual occlusion. Discount factor was set to $\gamma = 0.9$.

8) Approximate Q-learning Algorithm: The general procedure of the approximate Q-learning algorithm for training used in this paper is detailed in algorithm 1. After training, ϵ is set to zero, w_i stops updating, and deterministic optimal trained policy is used for testing.

III. Simulation Study and Evaluation

To evaluate the efficacy of the proposed SL algorithm based on linear approximate Q-learning for robotic tissue manipulation, we first developed a simulation environment. We then designed tissue manipulation simulation experiments based on real surgical scenarios and evaluated the performance of the algorithm from both learning and tissue manipulation point of views.

A. Simulation Environment

The simulation environment was created using CHAI3D [31]. OpenCV library [32] was deployed for image processing purposes. Fig. 2 depicts the simulation environment.

The high frequency (∼1 kHz) *Dynamics Thread* solves the underlying partial differential equations (PDEs) in real time to update the position of each node (shown by blue spheres in Fig. 2a). The low frequency (∼60 Hz) *Graphics Thread* receives the data from dynamics thread and renders the image by updating the image pixels. This rendering is internally taken care of by means of OpenGL library [33]. For the purpose of this work, a new thread, *Image Processing Thread*, was added that runs at (∼60 Hz) and performs image filtering to obtain an estimate of the states for the problem.

1) Simulation Model: The GEL module of the open-source CHAI3D library [31] is used to simulate the motion/force of a real deformable tissue. The model consists of a rectangular prism structure of $25 \times 25 \times 1$ *Nodes* (see Fig. 2b). The nodes of each lateral surface of the model are held fixed. Each node is connected to its adjacent nodes by means of *Links* which specify how the motion of the nodes affect each other. The model takes into account the physical properties of each node (i.e. mass and radius) along with the physical properties of each link (i.e. linear and rotational stiffness and damping). This general structure is then used to calculate the interaction forces and dynamics of the tissue. For the purpose of this paper, additional software was developed so any node can be grasped and moved.

2) Vision Algorithm: Since in real surgical scenarios the IDPs are expressed in endoscopic images by surgeons, proper image processing should be used to extract TTPs pixel information from the images in real time. The image processing thread receives the frame buffer of the CHAI3D virtual camera at 60 Hz which could be replaced by an endoscopic frame buffer for software testing on an actual robotic surgery system. While more advanced algorithms (i.e. based on Convolutional Neural Nets [34]) could be used for

Algorithm 1 Approximate Q-learning

```
set $n_{episode}$ to 0
set $N_{episode}$ and $N_{action}$ values
initialize weights to random numbers uniformly
distributed between -0.5 and 0.5
while ($n_{episode} < N_{episode}$) do
    initialize configuration of simulation environment
    set $n_{action}$ to 0
    increase $n_{episode}$ by 1
    while ($n_{action} < N_{action}$) do
        set $\epsilon$ and $\alpha$
        select $\epsilon$-greedy action a
        calculate all $f_i(s,a)$ and $Q(s,a)$
        take action a
        calculate $R(s,a,s')$ and $\max_{a'} Q(s',a')$
        update all $w_i$ according to (4)
        increase $n_{action}$ by 1
```

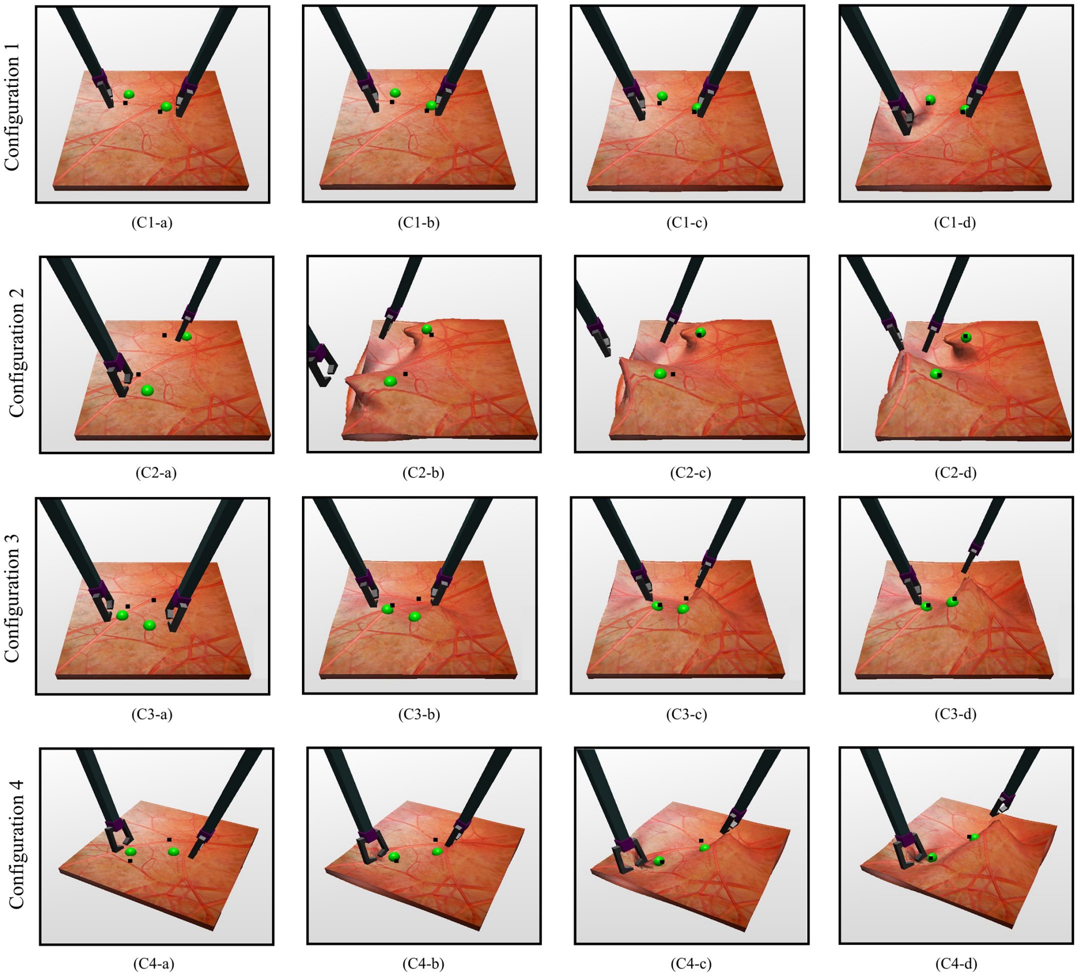


Fig. 3. Example successful policies. Four configurations (C1-C4) are displayed for representative progressing actions during testing episodes from (a) *initialization* to (d) *task completion*. Each configuration represents a different set of initial TTPs, IDPs, and TGPs. Additionally, in (C4), the camera configuration is changed. In order to maintain simulation stability, it was necessary to manipulate TGPs via simulated springs with low stiffness connected to the graspers. As such, small gaps between graspers and the tissue are observable for large deformations.

tracking TTPs, in this work we use filters including Gaussian Blur and color-based HSV for image processing. The x and y center of the smallest circle containing all the identified TTP pixels are used to represent the position of the TTPs.

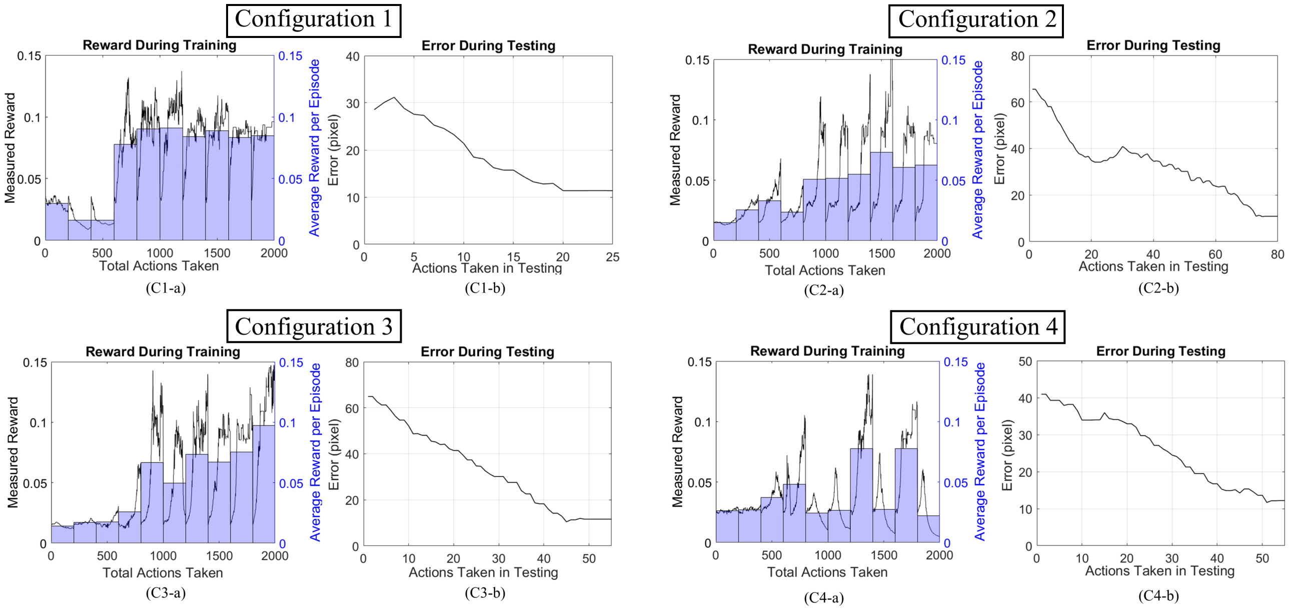
3) Q-Learning Algorithm: Once the Vision Algorithm obtains an estimate of states (position of the TTPs), it inputs the estimated states to the Q-learning algorithm (**Algorithm 1**) upon a query. State estimation is then used to calculate $f_i(s,a), Q(s,a), R(s,a,s')$ and $\underset{a'}{max}\ Q(s',a')$ and it takes ϵ-greedy action a.

B. Simulation Experiment Design

To design the experiments, we considered realistic situations that might happen during a robotics tissue manipulation procedure. For instance, locations of TGPs and IDPs are defined based on the clinical situations, surgical sites, and the preference of the surgeon. Therefore, the utilized manipulation algorithm must be robust to these variations. A relevant example for autonomous tissue manipulation is an anastomosis, where two tissues must be accurately positioned adjacent to each other for proper suturing. Moreover, although the camera calibration data may be available during real surgical robotic procedures, it may not be accurate or may vary when the experimental parameters are changed. Hence, it is highly desirable that the algorithm performance be independent of camera calibration and/or configuration. Considering these realistic cases, the performance of the algorithm was assessed on four different configurations, as shown is Fig. 3. In the first three sets of experiments, the initial configurations of all TGPs, IDPs, and TTPs were changed (CASE I). For the fourth experiment, in addition to changing these points, the camera location was also altered (CASE II).

C. Performance Evaluation Metrics

To evaluate our algorithm, we assessed its performance from learning point of view as well as tissue manipulation perspective. In this regard, the learning performance was measured based on the reward during training episodes and error during testing. Tissue manipulation task was considered accomplished based on the Root Mean Square (RMS) error between TTPs and IDPs pixel coordinates.



(C1-a) (C1-b) (C2-a) (C2-b)

(C3-a) (C3-b) (C4-a) (C4-b)

Fig. 4. Reward during training and error during testing. (a) Rewards across all episodes of training for each configuration. The line plots indicate the reward per action while the bar plots display the average reward in each episode. (b) Euclidean error norm for single testing episodes for each configuration.

IV. Results and Discussion

In each training set, $N_{episode} = 10$ episodes and $N_{action} = 200$ actions were used. Following the complete 2000 action training set, the trained policy was assessed with testing episodes using a greedy policy ($\epsilon = 0$) with $\alpha = 0$.

Fig. 3 displays four time instances from (a) *initialization* to (d) *task completion* of robotic tissue manipulation for the described four different simulation study. In these studies, the goal is to indirectly manipulate Green TTPs via TGPs to reach black IDPs. The first three rows are associated with the CASE I and the last row is associated with CASE II.

Fig. 4 shows the defined performance evaluation metrics including the reward during the training episodes of each simulation study as well as the error which is reciprocal of the reward during testing, as defined in (5). As can be observed in Fig. 4, the algorithm correctly learns a policy that increases reward (on average) as training progresses. Of note, the displayed rewards are associated with all actions during the training sets leading to the policies shown in Fig. 3. For each configuration, reward increases both within episodes and across episodes. Note that the high frequency variation in reward at high values is due to the action size causing individual actions to significantly effect reward.

For testing episodes, Fig. 4 plots the Euclidean error norm between TTPs and IDPs against the number of actions taken. While configuration 1 does not require many actions, the learned policy directs the TGPs to quickly move the TTPs to the IDPs. Configuration 2 and 3 represent more challenging tasks that requires significant strain on the simulated tissue. Despite the increased complexity of the dynamics with large deformation, successful policies are learned. Configuration 4 further increases the complexity of the task by changing the configuration of the camera. As can be seen in Fig. 3, instances C4-a through C4-d and Fig.4, instances C4-a through C4-b, the proposed algorithm was capable of learning weights that provide a successful policy. The converging errors in Fig. 4 instances C1-b through C4-b highlight the value of the defined feature in (7). This feature effectively serves as a completion criterion for the task by halting both graspers when the reward exceeds 0.08, which corresponds to a Euclidean error of 12.5 pixels, or roughly 8 pixel error for each TTP. This criterion can be modified as needed for a given surgical setup to reflect proper completion of the task. The four configurations shown represent very different initial configurations, yet the weight associated with the feature is properly trained in each case, demonstrating the learning capabilities of the proposed SL algorithm.

There were a number of challenges for both developing and running the proposed SL algorithm:

Visual Occlusion: Due to the simulation environment including avatars of both surgical robot graspers, as well as the use of an opaque tissue phantom model, it is not uncommon for vision of one or both TTPs to become occluded. Proper selection of TGPs relative to TTPs in the camera frame can help to prevent the graspers or dragged tissue from obscuring the TTPs during exploitation with a good policy. However, during exploration, the random actions will cause the grasper to pass over nearly all nearby points including the TTPs. When this occurs, it was found necessary to halt the update of all weights until proper vision is restored via actions selected by the current ϵ-greedy policy. Additionally, if a grasper passes over a TTP, it is possible for the vision algorithm to incorrectly determine the center of the TTP, creating noise in the reward and features. The cyclical decreasing learning rate used in this work was found to assist in overcoming this noise.

Feature Selection: Feature selection is among most challenging part of feature-based Q-Learning as there is not a standard method for selecting features. Features must be selected based upon knowledge of the system and potentially subjective judgments for what equation should be optimized.

This work aimed to show that incorporation of simple features based on intuitive human knowledge were capable of solving the highly complicated task of simulated tissue manipulation with no camera calibration information with approximate Q-learning. The features chosen were selected to be the simplest and minimum number necessary to approach the task without knowledge of registration between camera, tissue, and robot. The developed algorithm converges to a successful solution, indicating that the simulation environment, image processing, and Q-learning all are designed reasonably. As such, the authors consider this work to be a successful implementation of a computational framework on a previously undocumented Q-learning problem.

Future work will be geared towards applying the presented algorithm in a real-world environment. Specifically, the Raven surgical robotic system will be used in conjunction with tissue phantoms. To accomplish this, the current algorithm will need to be expanded to consider anatomical constraints as well as the robot and gripper kinematics and joint limitations. Additionally, a more generalized 3D motion of the robot gripper will be considered. These will, for example, affect the a and a' in (3). Lastly, the assumption of an ideal point grasping scenario will need to be validated.

References

[1] A. Kapoor, M. Li, and R. H. Taylor, "Spatial motion constraints for robot assisted suturing using virtual fixtures," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 89–96, Springer, 2005.

[2] S. A. Pedram, P. Ferguson, J. Ma, E. Dutson, and J. Rosen, "Autonomous suturing via surgical robot: An algorithm for optimal selection of needle diameter, shape, and path," in *Robotics and Automation (ICRA), IEEE Int'l Conference on*, pp. 2391–2398, IEEE, 2017.

[3] S. Sen, A. Garg, D. V. Gealy, S. McKinley, Y. Jen, and K. Goldberg, "Automating multi-throw multilateral surgical suturing with a mechanical needle guide and sequential convex optimization," in *IEEE Int'l Conf. on Robotics and Automation (ICRA)*, pp. 4178–4185, 2016.

[4] A. Shademan, R. S. Decker, J. D. Opfermann, S. Leonard, A. Krieger, and P. C. Kim, "Supervised autonomous robotic soft tissue surgery," *Science translational medicine*, vol. 8, no. 337, 2016.

[5] F. Alambeigi, Z. Wang, Y.-h. Liu, R. H. Taylor, and M. Armand, "Toward semi-autonomous cryoablation of kidney tumors via model-independent deformable tissue manipulation technique," *Annals of Biomedical Engineering*, vol. 46, no. 10, pp. 1650–1662, 2018.

[6] F. Alambeigi, Z. Wang, R. Hegeman, Y.-H. Liu, and M. Armand, "Autonomous data-driven manipulation of unknown anisotropic deformable tissues using unmodelled continuum manipulators," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 254–261, 2019.

[7] A. Murali, S. Sen, B. Kehoe, A. Garg, S. McFarland, S. Patil, W. D. Boyd, S. Lim, P. Abbeel, and K. Goldberg, "Learning by observation for surgical subtasks: Multilateral cutting of 3d viscoelastic and 2d orthotropic tissue phantoms," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1202–1209, IEEE, 2015.

[8] C. Coulson, R. Taylor, A. Reid, M. Griffiths, D. Proops, and P. Brett, "An autonomous surgical robot for drilling a cochleostomy: preliminary porcine trial," *Clinical Otolaryngology*, vol. 33, no. 4, pp. 343–347, 2008.

[9] F. Alambeigi, M. Bakhtiarinejad, S. Sefati, R. Hegeman, I. Iordachita, H. Khanuja, and M. Armand, "On the use of a continuum manipulator and a bendable medical screw for minimally-invasive interventions in orthopedic surgery," *IEEE Trans. on Medical Robotics and Bionics*, 2019.

[10] F. Alambeigi, Z. Wang, R. Hegeman, Y.-H. Liu, and M. Armand, "A robust data-driven approach for online learning and manipulation of unmodeled 3-d heterogeneous compliant objects," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4140–4147, 2018.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[12] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[13] S. Hirai and T. Wada, "Indirect simultaneous positioning of deformable objects with multi-pinching fingers based on an uncertain model," *Robotica*, vol. 18, no. 1, pp. 3–11, 2000.

[14] P. Jiménez, "Survey on model-based manipulation planning of deformable objects," *Robotics and computer-integrated manufacturing*, vol. 28, no. 2, pp. 154–163, 2012.

[15] C. Shin, P. W. Ferguson, S. A. Pedram, J. Ma, E. P. Dutson, and J. Rosen, "Autonomous tissue manipulation via surgical robot using learning based model predictive control," in *Int'l Conference on Robotics and Automation (ICRA)*, pp. 3875–3881, IEEE, 2019.

[16] F. Alambeigi, S. A. Pedram, J. L. Speyer, J. Rosen, I. Iordachita, R. H. Taylor, and M. Armand, "Scade: Simultaneous sensor calibration and deformation estimation of fbg-equipped unmodeled continuum manipulators," *IEEE Transactions on Robotics*, 2019.

[17] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[18] J. Peters, K. Mulling, and Y. Altun, "Relative entropy policy search," in *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[19] M. P. Deisenroth, G. Neumann, J. Peters, *et al.*, "A survey on policy search for robotics," *Foundations and Trends® in Robotics*, vol. 2, no. 1–2, pp. 1–142, 2013.

[20] S. Gu, T. Lillicrap, I. Sutskever, and S. Levine, "Continuous deep q-learning with model-based acceleration," in *International Conference on Machine Learning*, pp. 2829–2838, 2016.

[21] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[22] C. J. C. H. Watkins, *Learning from delayed rewards*. PhD thesis, King's College, Cambridge, 1989.

[23] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[24] D. Navarro-Alarcon, H. M. Yip, Z. Wang, Y.-H. Liu, F. Zhong, T. Zhang, and P. Li, "Automatic 3-d manipulation of soft objects by robotic arms with an adaptive deformation model," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 429–441, 2016.

[25] B. Hannaford, J. Rosen, D. W. Friedman, H. King, P. Roan, L. Cheng, D. Glozman, J. Ma, S. N. Kosari, and L. White, "Raven-ii: an open platform for surgical robotics research," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 4, pp. 954–959, 2013.

[26] R. S. Sutton and A. G. Barto, *Introduction to reinforcement learning*, vol. 135. MIT press Cambridge, 1998.

[27] T. J. Rivlin, *An introduction to the approximation of functions*. Courier Corporation, 2003.

[28] D. Pandey and P. Pandey, "Approximate q-learning: An introduction," in *2010 int'l conf. on machine learning and computing*, pp. 317–320.

[29] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[30] L. N. Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 464–472, March 2017.

[31] F. Conti, F. Barbagli, R. Balaniuk, M. Halg, C. Lu, D. Morris, L. Sentis, J. Warren, O. Khatib, and K. Salisbury, "The chai libraries," in *Proc. of Eurohaptics 2003*, (Dublin, Ireland), pp. 496–500, 2003.

[32] G. Bradski and A. Kaehler, "Opencv," *Dr. Dobb's journal of software tools*, vol. 3, 2000.

[33] D. Hearn, M. P. Baker, and M. P. Baker, *Computer graphics with OpenGL*, vol. 3. Pearson Prentice Hall Upper Saddle River, NJ:, 2004.

[34] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.