OPEN ACCESS
Journal of Medical Robotics Research, Vol. 6, Nos. 1 & 2 (2021) 2140003 (16 pages)

DOI: 10.1142/S2424905X21400031

Journal of Medical Robotics Research
https://www.worldscientific.com/worldscinet/jmrr

Journal of Medical Robotics Research

Multicamera 3D Viewpoint Adjustment for Robotic Surgery via Deep Reinforcement Learning

Yun-Hsuan Su[*,§], Kevin Huang[†,¶], Blake Hannaford[‡,‖]

[*]*Department of Computer Science, Mount Holyoke College*
50 College Street, South Hadley, MA 01075, USA

[†]*Department of Engineering, Trinity College*
300 Summit Street, Hartford, CT 06106, USA

[‡]*Department of Electrical and Computer Engineering, University of Washington*
185 Stevens Way, Paul Allen Center — Room AE100R, Campus Box 352500, Seattle, WA 98195-2500, USA

While robot-assisted minimally invasive surgery (RMIS) procedures afford a variety of benefits over open surgery and manual laparoscopic operations (including increased tool dexterity, reduced patient pain, incision size, trauma and recovery time, and lower infection rates [1], lack of spatial awareness remains an issue. Typical laparoscopic imaging can lack sufficient depth cues and haptic feedback, if provided, rarely reflects realistic tissue–tool interactions. This work is part of a larger ongoing research effort to reconstruct 3D surfaces using multiple viewpoints in RMIS to increase visual perception. The manual placement and adjustment of multicamera systems in RMIS are nonideal and prone to error [2], and other autonomous approaches focus on tool tracking and do not consider reconstruction of the surgical scene [3–5]. The group's previous work investigated a novel, context-aware autonomous camera positioning method [6], which incorporated both tool location and scene coverage for multiple camera viewpoint adjustments. In this paper, the authors expand upon this prior work by implementing a streamlined deep reinforcement learning approach between optimal viewpoints calculated using the prior method [6] which encourages discovery of otherwise unobserved and additional camera viewpoints. Combining the framework and robustness of the previous work with the efficiency and additional viewpoints of the augmentations presented here results in improved performance and scene coverage promising towards real-time implementation.

Keywords: Robot-assisted minimally invasive surgery; 3D reconstruction; deep reinforcement learning; machine vision.

JMRR

1. Introduction

In their previous work, the authors proposed an autonomous camera viewpoint adjustment method for robot-assisted minimally invasive surgery (RMIS) that both considered tool tracking and maximizing scene coverage [6]. The work presented there also provided insight into the performance effects of the number of cameras used. The work here extends that work by incorporating a deep reinforcement learning (RL) agent to inform camera motion in between iterations of calculated poses to reduce computational burden. This addition seeks to garner improved viewpoint generation to explore the unknown surgical environment, and experiments demonstrate performance of the novel framework over simple interpolation with a real-time streaming point cloud of a surgical scene and tool.

Received 15 December 2020; Revised 28 March 2021; Accepted 16 April 2021; Published 10 July 2021. Published in JMRR Special Issue: ISMR 2020. Guest Editor: Iulian Ioan Ioardachita.
Email Addresses: [§]*msu@mtholyoke.edu*, [¶]*kevin.huang@trincoll.edu*, [‖]*blake@uw.edu*

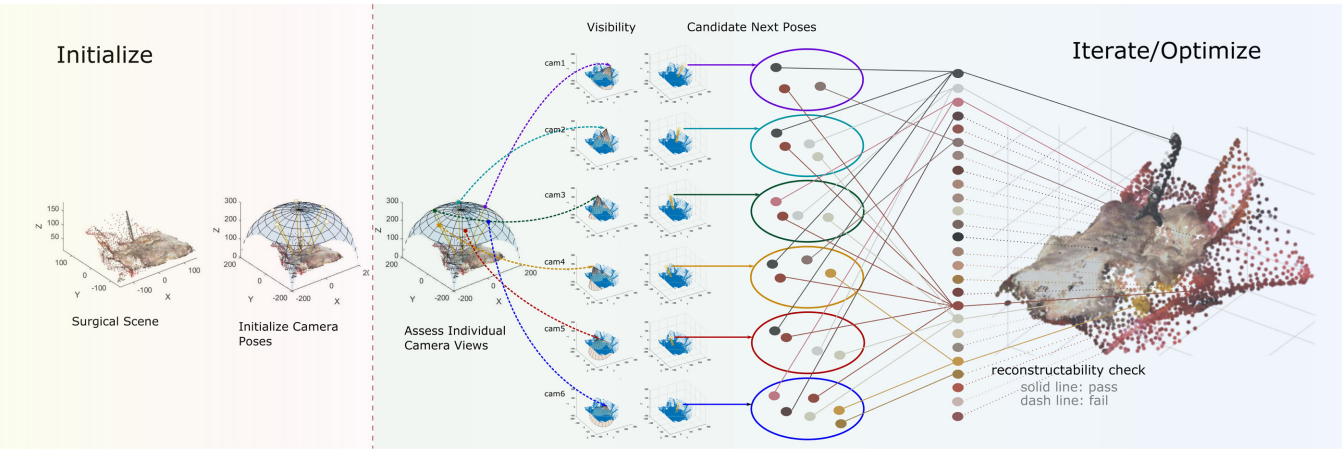


Fig. 1. Automating camera viewpoint adjustment to find next camera poses following abdominal wall constraints to achieve the maximum number of 3D reconstructable points. First cameras are positioned so that they conform to abdominal wall physical constraints and spaced evenly in an initial position. In each iteration, candidate next poses are evaluated for each camera. The selection accounts for coverage, surgical tool-tip position and a custom reconstructability score.

1.1. *Contributions*

In this work, the authors present augmentations and improvements to the previous work by

- incorporating deep RL to govern camera motion and exploration in between viewpoints calculating using previous approaches for constrained maximum coverage;
- experimentally demonstrating and evaluating the method with dynamic surgical scene 3D reconstruction performance.

The procedure prescribes camera poses and reconstructability coverage as cameras are instructed to move to another goal pose determined via previous work [6].

1.2. *Background*

To address surgeon situational awareness and perception, 3D surgical cavity reconstruction can provide spatial awareness with increased field of view [7] as well as a conduit towards vision-based force estimation [8–10]. While approaches using single monocular views for 3D surgical reconstruction have been explored [11, 12], the use of multiple viewpoints has been demonstrated to improve the surgeon's perception of the surgical task and scene [13]. Multiple camera configurations do not require additional incisions or ports, as demonstrated by Silvestri *et al.* [14], who inserted an array of camera devices embedded in a single device into the surgical cavity via a single trocar. Following insertion, cameras can then be individually controlled wirelessly using external magnets, resulting in an extremely articulated set of vision sensors for RMIS [15–18]. Because of the nature of this positioning mechanism, camera poses exist along the patient's inner abdominal wall.

Automating the motion and arrangement of multiple cameras to generate quality surgeon viewpoints is a point of interest, since manual human control requires coordination and communication with the operating surgeon. Beyond directly assisting the surgeon's scene understanding, a robust reconstruction of the scene can enable intelligent intraoperative augmentations for RMIS. The authors' previous work [6] presented an autonomous multicamera viewpoint adjustment scheme to such an approach, which considered reconstructability and aimed to maximize scene coverage viewed by at least two different cameras. The problem setup can be summarized in Fig. 1. The method is used in this work to determine approximate end poses of a surgical process. Due to the computational complexity of the method, a more streamlined approach, the deep RL application presented here, is used to interpolate between calculated camera configurations.

1.3. *Related work*

1.3.1. *Next best view and swarm mapping*

Automated optimal viewpoint generation is paramount for a variety of robotic operations, including environmental monitoring, object recognition, object manipulation, robotic navigation, and mesh simplification for polygonal models [19] to name just a few. A similar problem involves placing heterogeneous sensory agents in constrained physical tasks presents to aid in achieving optimal coverage to automatically monitor a perimeter

[20]. Similarly, the next best view (NBV) problem aims to solve or prescribe ideal sensing commands (typically view sequences) to achieve a task, with either surface-based, global-based, or volumetric approaches [21]. These methods use context, task or feature cues to plan the NBVs, and simultaneously reduce an ambiguity function and consider planning [22]. Additional augmentations to NBV approaches include incorporating task specific weights into objective functions, such as manipulator motion cost [23], reachability, view overlap [24, 25], and even camera parameters [26, 27].

Unmanned aerial vehicle (UAV) swarms have been deployed for distributed monitoring tasks, including object tracking (single target [28, 29] and multi-target methods [30]), coordination, environmental monitoring, data collection, and path planning [31]. Individual UAV agents within a swarm oftentimes implement local algorithms, e.g. obstacle avoidance or wall-following, while simultaneously following centralized, high-level departure point or dispersion commands [32]. Combining centralized and decentralized objectives can result in robust and efficient mapping, making UAV swarms suitable for a variety of response or monitoring tasks, including water [33] and pollution [34] management, and wild fire detection [35].

Coverage and view-planning methods described above have been demonstrated in environments with large-scale physical constraints and openly navigable spaces. Furthermore, most obstacles or targets are either tracked at lower specificity or assumed rigid. While the methods afford insight into potential approaches, surgical environments are inherently different; surfaces are deforming, dynamic, and highly reflective, and the operating physical space is highly constrained [36].

1.3.2. *Surgical cavity viewpoint adjustment*

Direct control of both surgical tools and vision systems for surgery can lead to increased burden, time and risk. The use of multiple vision sensors and intelligent autonomy may potentially reduce these costs.

While multiview systems do exist for intra-abdominal telerobotic and laparoscopic surgery, intelligence with concurrent high degree of maneuverability remains a challenge. Tamadazte *et al.* [37] developed a multiview approach by embedding two high-definition cameras into a single endoscope to generate a stereoscopic view of the surgical scene. The configurations of two viewpoints once deployed are fixed in position relative to one another. Kim *et al.* [38] investigated a similar approach for laparoscopic surgery with a four-camera array. Similar devices have also been developed that exhibit slightly higher degrees of reconfigurability, such as the trocar-fixed devices developed by Kanhere *et al.* [39] and Afifi *et al.* [40]. These devices, while they can enhance scene visibility compared to a single endoscope and can provide some depth information, lack the reconfiguration flexibility to more thoroughly view the surgical scene.

A single camera may afford greater maneuverability and automation. Intelligent motion of single endoscopes or vision sensors can increase safety and occupy less space. Ma *et al.* introduced an autonomous flexible endoscope that was visually servoed for tool tracking [41]. Space occupation and motion were incorporated as cost factors. Prendergast *et al.* developed an autonomous region estimator-based endoscope navigation strategy that was evaluated on simulated anatomical structures [42]. Martin *et al.* utilized machine vision to incorporate intelligence and autonomy into a magnetic endoscope [43]. Da Col *et al.* conducted a user study that suggested that for the Laparoscopic Skills Training and Testing assessment, an autonomous camera positioning strategy that tracks tool location provided performance enhancements over manual control of the camera [44]. In these cases, a single vision sensor was used, and in most cases the viewpoint was determined by tracking tool location, and 3D scene reconstruction was not the primary goal — direct utility to the surgeon's view was the focus.

Multiple cameras that may be individually articulated without coupling to other configurations is a promising alternative to gather scene information to perhaps inform intelligent assistance. Magnetic anchoring guidance systems are promising enhancement that extends the surgical field of view [45]. The method has also been demonstrated within porcine models [15, 16, 18]. The development of a multicamera device [14] enables more viewpoints, and intelligently maneuvering the cameras to assist in surgical operations with better surgical scene reconstruction is of interest in this work. The direct benefit of the scene reconstruction may not necessarily be used as visual feedback to the surgeon, but instead to afford other augmentations, e.g. force estimation.

In prior work [6], the authors explored a context-aware method for automatic multicamera positioning to both track the surgical tool position and optimize reconstructability, i.e. coverage with at least two cameras viewing. A dynamic, updating 3D surgical cavity was used [46], and the camera viewpoint problem was formulated as an optimization problem [47]. The results provide insight into computational efficiency and coverage with varying camera number. Because of computational complexity, real-time application is highly constrained.

In this work, the same methods are used to predict high-coverage camera goal poses at a coarser temporal scale, while a streamlined deep RL approach drives the camera state to the desired pose while negotiating similar constraints. At the coarse timescale, recognizing surgical gestures [48–50] and predicting organ movements [51, 52] can provide one-step-away surgical state

estimates to inform optimal camera poses. In the following sections, the authors summarize relevant portions of previous work. For more detailed descriptions, the authors refer the reader to the previous publication [6].

2. Methods

2.1. *Maximum coverage formulation*

2.1.1. *Camera poses and motion*

Suppose there are $N_{\mathbb{C}} \in \mathbb{N}$ cameras and $N_{\mathbb{C}} \geq 2$. At each iteration, a camera has $N_{\mathbb{K}} \in \mathbb{N}$ motion options (including remaining stationary). Then denote $\mathbf{c}_i, \mathbf{n}_{ik} \in \mathbb{R}^6$ as current and candidate next camera configurations (see Fig. 2) for camera i, respectively, where $i = 1 \ldots N_{\mathbb{C}}$, and $k = 1 \ldots N_{\mathbb{K}}$.

If the one-hot representation of the selected next camera pose is $\mathcal{C}_i \in [0,1]^{N_{\mathbb{K}}}$, then

$$\|\mathcal{C}_i\|_1 = \sum_{k=1}^{N_{\mathbb{K}}} \mathcal{C}_i(k) \leq 1, \quad \forall k = 1, \ldots, N_{\mathbb{K}}, \mathcal{C}_i \in \mathcal{C}, \tag{1}$$

where $\mathcal{C}_i(k)$ indicates the kth element in $\mathcal{C}_i$ and $\mathcal{C}$ is $\{\mathcal{C}_i | i = 1, \ldots, N_{\mathbb{C}}\}$.

2.1.2. *Coverage cost function*

The problem is formulated as an optimization of surgical scene coverage. The surgical scene is assumed to be represented as an unordered 3D point cloud. Denote this model as a set of $N_{\mathbb{P}}$ points and call it $\mathbb{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_{N_{\mathbb{P}}}\}$ where $\mathbf{p}_j \in \mathbb{R}^3$. In this formulation, a point $\mathbf{p}_j$ is deemed visible so long as it is viewable by at least two different cameras; this is to satisfy the requirements of 3D stereo reconstruction [53]. To track the visibility of each point, a visibility flag, v_j, is assigned to each point $\mathbf{p}_j$. Specifically, $\forall j = 1, \ldots, N_{\mathbb{P}}$, $v_j = 1$ if $\mathbf{p}_j$ is visible by two or more cameras, and $v_j = 0$ otherwise.

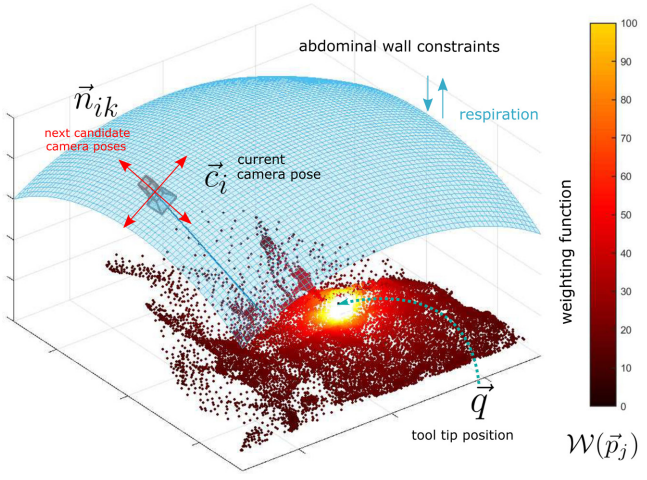


Fig. 2. An overview of the formulated task for each camera. Cameras are constrained physically to the inner abdominal wall, and can move from current, $\mathbf{c}_i$, to next candidate camera poses, $\mathbf{n}_{ik}$. The colorbar shows the weighting value $\mathcal{W}(\mathbf{p}_j)$.

In addition to visibility, the proposed objective function accounts for the relative importance of viewing each point. To that end, a custom weight positively correlated with the viewing importance is assigned to each arbitrarily selected point $\mathbf{p}_j \in \mathbb{P}$ via a weighting function $\mathcal{W}(\mathbf{p}_j)$. The definition of the viewing importance can be crafted according to distinct surgical operation needs. In this experiment, it is assumed that the surgeon's region of interest is localized around the surgical tool tip; structures proximal to the surgical tool are weighted as more important. Therefore, the viewing importance is inversely proportional to the capped distance between point $\mathbf{p}_j$ and the tool tip $\mathbf{q}$. For this, let the surgical tool tip position be denoted $\mathbf{q} \in \mathbb{R}^3$, ascertained, for example, via robot kinematics. Then the weighting function is defined as

$$\mathcal{W}(\mathbf{p}_j) = \frac{1}{\max(\|\mathbf{p}_j - \mathbf{q}\|_2, \epsilon)}, \tag{2}$$

where ϵ is a heuristically tuned minimum distance threshold. With the weighting function defined, the maximum coverage problem can be expressed as

$$\text{Maximize} \sum_{j=1}^{N_{\mathbb{P}}} \mathcal{W}(\mathbf{p}_j) v_j. \tag{3}$$

2.1.3. *Motion penalty*

A motion penalty for camera $\mathbf{c}_i$ and next pose $\mathbf{n}_k$, $\mathcal{M}(i,k) \in \mathbb{R}$, may be assigned to encourage smooth and stable camera adjustments.

$$\begin{aligned} m_1 &= \mathcal{P}(\mathbf{c}_i) - \mathcal{P}(\mathbf{n}_{ik}), \\ m_2 &= \mathcal{Q}(\mathbf{c}_i) \cdot p\mathcal{Q}(\mathbf{n}_{ik}), \\ \mathcal{M}(i,k) &= \|m_1\|_2 + \alpha_1 \|m_2 - 1\|_1 \\ &\quad + \alpha_2 \|m_1 - \hat{m}_1^*\|_2 + \alpha_3 \|m_2 - \hat{m}_2^*\|_1, \end{aligned} \tag{4}$$

where $\mathcal{P}$, $\mathcal{Q}$ extract position and quaternion orientation information, respectively. $\hat{m}_1^*$ and $\hat{m}_2^*$ are m_1, m_2 values from the previous time step.

A heuristically tuned tolerance threshold $T_{\mathcal{M}}$ imposes

$$\mathcal{L}(\mathcal{C}) = \sum_{i=1}^{N_{\mathbb{C}}} \sum_{k=1}^{N_{\mathbb{K}}} \mathcal{M}(i,k) \mathcal{C}_i(k) \leq T_{\mathcal{M}} \tag{5}$$

to regulate jitter. This threshold determines the extent of overall camera motion and should be tuned based on the surgical operation type and the individual surgeon's preference.

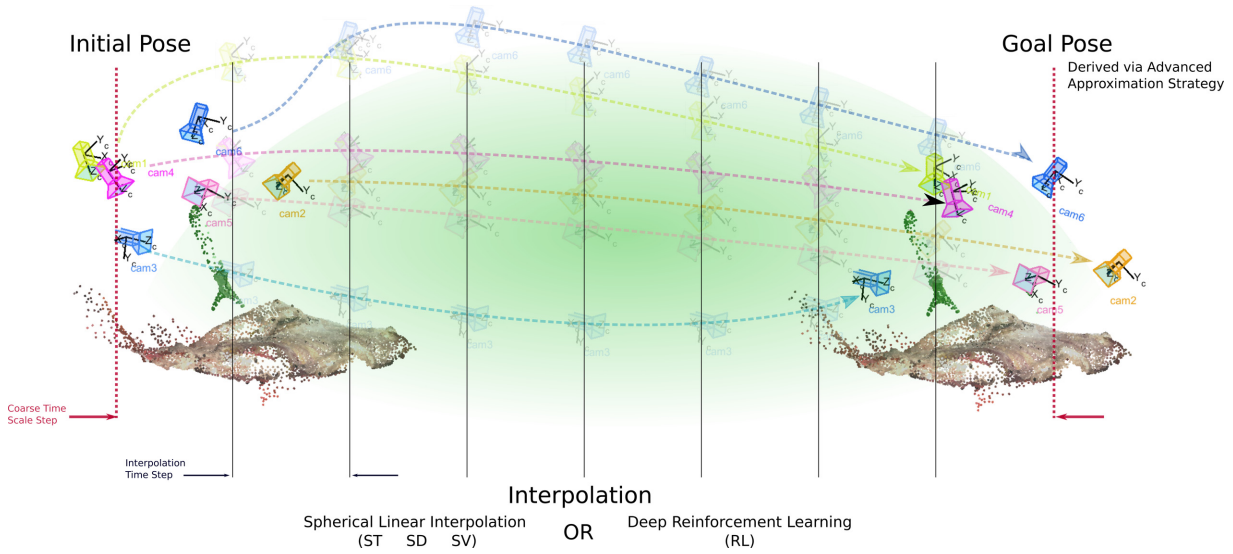


Fig. 3. The goal pose generation and interpolation framework investigated in this paper. Initial camera poses, as depicted on the left, inform a goal pose using optimal camera viewpoint methods developed in previous work [6]. The time scale of this algorithm lends itself to interpolating camera poses in between. Four different interpolation methods — ST, SD, SV, and RL — are defined and explored in Secs. 2.3.1 and 2.3.2.

2.1.4. *Visibility and reconstructability*

Each candidate pose's visibility and reconstructability are also evaluated. Consider visibility function $\mathcal{V}(i,k,\mathbf{p}_j)$:

$$\mathcal{V}(i,k,\mathbf{p}_j) = \begin{cases} 1 & \text{if } \mathbf{p}_j \text{ is viewable from } \mathbf{n}_{ik}, \\ 0 & \text{if } \mathbf{p}_j \text{ is not viewable from } \mathbf{n}_{ik}. \end{cases} \tag{6}$$

Specular reflections [54] and proximity to image edge [55] also negatively affect reconstructability. This is accounted for via reconstructability function

$$\mathcal{R}(i,k,\mathbf{p}_j) = \begin{cases} 0.0 & \text{if } \beta < 0.1, \\ \min(\cos(\beta),\cos(\gamma)) & \text{else if } \beta \text{ or } \gamma > 1, \\ 1.0 & \text{else}, \end{cases} \tag{7}$$

where γ is the projection angle from pose $\mathbf{n}_{ik}$ and β the angle of reflection at point $\mathbf{p}_j$. Note that a local tissue patch with $\beta \simeq 0$ corresponds to a higher chance of specular reflections. See the authors' prior work [6] for mathematical definitions and details.

Then the following inequality constraint:

$$0.5 \sum_{i=1}^{N_{\mathbb{C}}} \sum_{k=1}^{N_{\mathbb{K}}} \mathcal{R}(i,k,\mathbf{p}_j)\mathcal{V}(i,k,\mathbf{p}_j)\mathcal{C}_i(k) \geq v_j, \quad \forall \mathbf{p}_j \in \mathbb{P} \tag{8}$$

ensures that v_j is set to 1 only if $\mathbf{p}_j$ is within view of two or more cameras and accounts for reconstructability.

2.2. *Goal pose optimization*

The goal pose for the multicamera system is derived by solving a budgeted maximum coverage problem:

$$\text{argmax}_{\mathcal{C}_i, \forall i \in N_{\mathbb{C}}} \sum_{j=1}^{N_{\mathbb{P}}} \mathcal{W}(\mathbf{p}_j) v_j \tag{9}$$

subject to constraints (1), (5), and (8)

$$\text{where } \mathcal{C}_i \in [0,1]^{N_{\mathbb{K}}}, \quad \forall i \in N_{\mathbb{C}}, \tag{10}$$

$$v_j \in [0,1], \quad \forall j \in N_{\mathbb{P}} \tag{11}$$

whose solution is approximated via methods described in previous work [6]. The method consists of three primary algorithms: $\mathcal{B}$, $\mathcal{G}$, and $\mathcal{A}$.

$\mathcal{B}$ compares coverage weight score $\mathcal{S}(h)$ of two input sets of next camera poses h_1 and h_2, then returns the set with greater score. For camera pose h,

$$\mathcal{S}(h) \sum_{j=1}^{N_{\mathbb{P}}} \mathcal{W}(\mathbf{p}_j) v_j, \tag{12}$$

where $\mathcal{G}$ algorithm returns a set of selected next camera poses derived from a greedy search of increased weight loss ratio. These results form the seed for a hybrid exhaustive search, $\mathcal{A}$. The results of $\mathcal{A}$ are compared with $\mathcal{B}$, of which the better one is selected. The reader is directed to previous work [6] for more details.

2.3. *Pose interpolation*

Given the start and goal multicamera configurations, the task is to navigate each camera between poses such that coverage and reconstructability described in previous sections are maintained, as depicted in Fig. 3. Let the set of current camera poses be $K = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{N_c}\}$ and

the calculated goal poses be denoted ${}_{\dagger}K = \{{}_{\dagger}\mathbf{c}_1, {}_{\dagger}\mathbf{c}_2, \ldots, {}_{\dagger}\mathbf{c}_{N_\mathbb{C}}\}$. The goal is to prescribe a sequence of camera motion to navigate from K to ${}_{\dagger}K$.

In this work, camera image frames are captured at 33 frames per second, and camera motion commands are executed at the same rate of 33 Hz. With this setup, a deep RL approach for navigating cameras between start and goal poses is proposed. This approach is experimentally compared to direct approaches of spherical linear interpolation (slerp) [56].

2.3.1. *Spherical linear interpolation*

As a baseline, three different utilizations of slerp [56] between calculated camera configurations are explored. Three different multicamera slerp procedures are explored, including

(i) ordinal turn taking (ST);
(ii) furthest distance (SD);
(iii) greedy visual coverage (SV).

These three procedures differ in the selection of camera slerp motion to execute and in what order.

To begin, interpolated slerp commands are calculated for each of the $N_\mathbb{C}$ cameras, label the cameras $\kappa_1, \kappa_2, \ldots, \kappa_{N_\mathbb{C}}$. Specifically, suppose that a total of $\mathbb{X}$ ordered slerp commands are calculated for each camera. Then camera κ_i has the ordered commands of

$$(x_{i1}, x_{i2}, \ldots, x_{i\mathbb{X}}).$$

Then in ST, the slerp motion commands are executed in order as $x_{11}, x_{21}, \ldots, x_{N_\mathbb{C}1}, x_{12}, x_{22}, \ldots, x_{N_\mathbb{C}\mathbb{X}}$, such that each camera takes a turn executing a motion command.

To execute procedure SD, first consider that each camera has $N_\mathbb{K}$ potential motion directions, and thus $\frac{N_\mathbb{K}}{2}$ degrees of freedom. For each camera κ_i and each degree of freedom d, calculate a distance score for each camera via

$$\mathcal{D}(\kappa_i, d) = ||\delta(\mathbf{c}_i) - \delta(-\dagger\mathbf{c}_i)||_1, \tag{13}$$

where δ extracts the value in camera configuration $\mathbf{c}_i$ directly affecting degree of freedom d. Then in SD, the next camera slerp motion is drawn from camera κ_s where

$$s = \text{argmax}_i[\mathcal{D}(\kappa_i, d)] \tag{14}$$

across all degrees of freedom. For SV, slerp command order is determined by evaluating the resultant score by evaluating (3) for each camera for the next motion command. Among the available motion commands, the one that maximizes the expression in (3) is selected. Figure 4 demonstrates a sequence of selected actions $a[\tau]$ from time $\tau = 1$ to 562 using the four investigated methods — ST, SD, SV, and RL. Each $a[\tau]$ is chosen from a list of potential actions $\{a_{11}, a_{12}, \ldots, a_{N_\mathbb{C}N_\mathbb{K}}\}$, where a_{ij} represents the ith camera moving in the jth potential motion direction. The camera motion action procedure RL will be elaborated in Sec. 2.3.2.

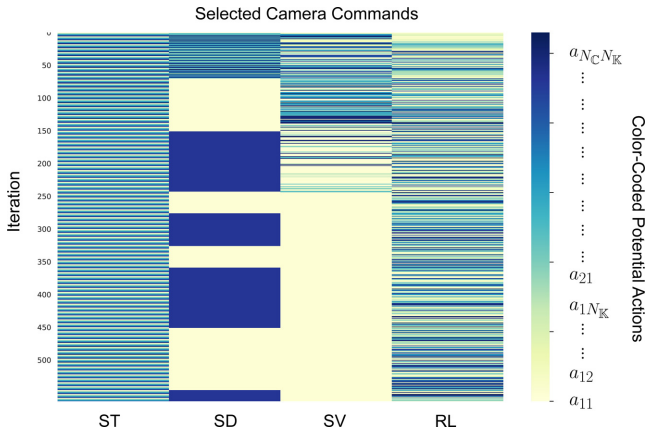


Fig. 4. Selected actions for each interpolation method (ST, SD, SV, and RL) over increasing iterations. The vertical axis shows the iteration number, and the chosen actions are color coded via color map to the right, labeling the potential actions $\{a_{11}, a_{12}, \ldots, a_{N_\mathbb{C}N_\mathbb{K}}\}$. See Sec. 2.3.1 for details about the list of potential actions and the action selection procedures.

2.3.2. *Deep reinforcement learning*

Viewing the three slerp baselines through the lens of RL, ST, and SD are variations of goal-reaching procedures that only value the final reward of reaching the target, whereas SV takes a slightly greedier approach that picks the best "scenic" route among the few shortest routes to target. However, these approaches are inflexible in considering potentially better views within reach of the interpolation path set forth. The authors are interested in investigating a deep RL-based [57] interpolation algorithm RL, which will both prioritize maximizing longer term visibility and simultaneously navigate towards the goal camera poses. Maximum reconstructability coverage and computational efficiency are metrics of interest in evaluating these approaches.

Theoretical background

To solve the multi-agent sequential decision problem of optimal camera viewpoint adjustment using RL [58], a few terms and variables must first be introduced. Suppose $Q_\pi(s[\tau], a[\tau], i)$ is an action-value function after i training epochs that predicts the benefit of making a move $a[\tau]$ at a particular state $s[\tau]$ given an action selection policy π. The quantification of benefit can be defined as the expected sum of the immediate reward from the state transition $R(s[\tau], a[\tau], s[\tau+1])$ and future rewards when taking that action and following the optimal policy π^* thereafter. Ideally, as $i \to \infty$, $Q_\pi(s[\tau], a[\tau], i)$ converges to a single value, $Q_\pi(s[\tau], a[\tau])$. Larger $Q_\pi(s[\tau], a[\tau])$ indicates greater incentives for the learning agent.

The policy π, on the other hand, defines most preferable state-action pairs that were visited and updated during training. In each epoch i of the training process, the agent updates $Q_\pi(s[\tau], a[\tau], i)$ for every state-action combination through the Q learning [59] formula:

$$Q_\pi(s[\tau], a[\tau], i+1) = (1-\zeta_1)Q_\pi(s[\tau], a[\tau], i) + \zeta_1 {}_\dagger R, \tag{15}$$

$${}_\dagger R = R(s[\tau], a[\tau]) + \zeta_2 \max_{a[\tau+1]} Q_\pi(s[\tau], a[\tau], i), \tag{16}$$

where $\zeta_1 \in [0,1]$ is the reward learning rate, and $\zeta_2 \in [0,1]$ is a discount factor used to balance how much the RL agent emphasizes immediate versus future rewards.

This framework oftentimes leads to an intractably large combination of states and actions. Such is the case in this multicamera pose adjustment task. To address this, the authors utilized a deep RL approach to estimate $Q_\pi(s[\tau], a[\tau])$ leveraging recent developments in deep learning (DL) methods. Two main challenges arise with the incorporation of supervised DL with RL [60].

(i) the requirement for independent and identically distributed (iid) state transition data in supervised DL is difficult to acquire in online RL;
(ii) the nonstationary target $Q_\pi(s[t], a[t])$ in RL causes convergence issues in supervised DL.

To resolve the former issue, the agent stores previous experiences in a local memory element called the replay buffer. This information is then randomly sampled to train the deep network. As the data from the replay buffer is sampled randomly, the sampled data approaches iid. This offline data generation procedure has been coined as experience replay [61–63]. The latter issue is addressed utilizing a temporarily fixed target network [60]. For this, two deep networks with parameter sets ${}_\dagger\theta$ and θ are created. Network ${}_\dagger\theta$ temporarily retrieves fixed $Q_\pi(s[\tau], a[\tau], {}_\dagger\theta)$ targets, thus effectively eliminating the moving target issue. Concurrently, network θ includes all additional updates in the training. This update rule of Deep Q-Network (DQN) learning is depicted in the following:

$$\Psi = [Q_\pi(s[\tau+1], a[\tau+1], \theta_i) - {}_\dagger R]^2, \tag{17}$$

$$\theta_{i+1} = \theta_i - \zeta_3 \nabla_\theta \mathbb{E}_{s[\tau+1]} \Psi, \tag{18}$$

where ζ_3 is the network learning rate and ${}_\dagger\theta$ is synced with θ every predetermined number of epochs. Furthermore,

$$\Lambda = \zeta_2 \max_{a[\tau+1]} Q_\pi(s[\tau+1], a[\tau+1], {}_\dagger\theta), \tag{19}$$

$$\dagger R = R(s[\tau], a[\tau]) + \Lambda. \tag{20}$$

Finally, in order to remove positive bias towards $Q_\pi(s[\tau+1], a[\tau+1], \theta)$ using DQN alone, double DQN [64] was utilized to greedily select the action and the target network ${}_\dagger\theta$ to estimate $Q_\pi(s[\tau+1], a[\tau+1], \theta)$.

Problem formulation

In general, the state $s[\tau]$ should be an aggregated feature representation of the environment, which includes all information needed for the agent to decide the next action $a[\tau]$. While sufficient information is necessary for the model to learn a good policy π, excessive information not only confuses the model but also delays the convergence by making the state-action search space exponentially larger. Therefore, the only features considered in the state space should fall into either of the two categories:

(i) features directly controlled by actions a;
(ii) features not directly controlled by actions a AND will affect the agent.

Since the problem objective is to optimally adjust camera poses, features falling into the former category, (i), are the camera poses K. Meanwhile, category (ii) may include the surgical tool configuration $\mathcal{P}(\mathbf{q}), \mathcal{Q}(\mathbf{q})$ and velocity $\mathcal{P}(\mathbf{q}_v)$, which affect changes in the surgical scene and thus the observed point clouds, as well as camera visibility; the patient's breathing cycle time index t_b affects the camera Z position; the camera goal poses ${}_\dagger K$. With that said, at time interval τ, the state $s[\tau]$ can be denoted as a $(10N_{\mathbb{C}}+9)\times 1$ vector

$$s[\tau] = [K, {}_\dagger K, \mathcal{P}(\mathbf{q}), \mathcal{P}(\mathbf{q}_v), \mathcal{Q}(\mathbf{q}), t_b]^T, \tag{21}$$

where the rotation about the Z-axis is disregarded. Hence, $K, {}_\dagger K \in \mathbb{R}^{5N_{\mathbb{C}}}$, $\mathcal{P}(\mathbf{q}), \mathcal{P}(\mathbf{q}_v) \in \mathbb{R}^3$, and $\mathcal{Q}(\mathbf{q}) \in \mathbb{R}^2$.

During the training phase of the DDQL algorithm, the reward function $\Gamma(s[\tau], a[\tau], s[\tau+1])$ consists of three terms:

$$\begin{aligned} &\Gamma_\alpha(a[\tau], s[\tau+1]) && \text{action reward,} \\ &\Gamma_r(s[\tau+1]) && \text{reconstruction reward,} \\ &\Gamma_g(s[\tau], s[\tau+1]) && \text{goal achievement reward.} \end{aligned}$$

Suppose d_M is the maximum distance in the surgical scene (in mm), then the action reward Γ_α reflects the state transition status and camera motion penalty. Specifically, this reward is derived as

$$\Gamma_\alpha(a[\tau], s[\tau+1]) = \begin{cases} -2.0 & \text{if illegal move,} \\ 10.0 & \text{else if goal reached,} \\ \dfrac{-\mathcal{L}(\mathcal{C})}{d_M} & \text{else. See (5).} \end{cases} \tag{22}$$

An action $a[\tau]$ is considered illegal when the camera either moves out of bounds or collides with another camera. In addition, the goal reached reward is only valid when all cameras are in their respective goal poses.

The reconstruction reward Γ_r rewards high reconstructability of the surgical scene and is calculated for

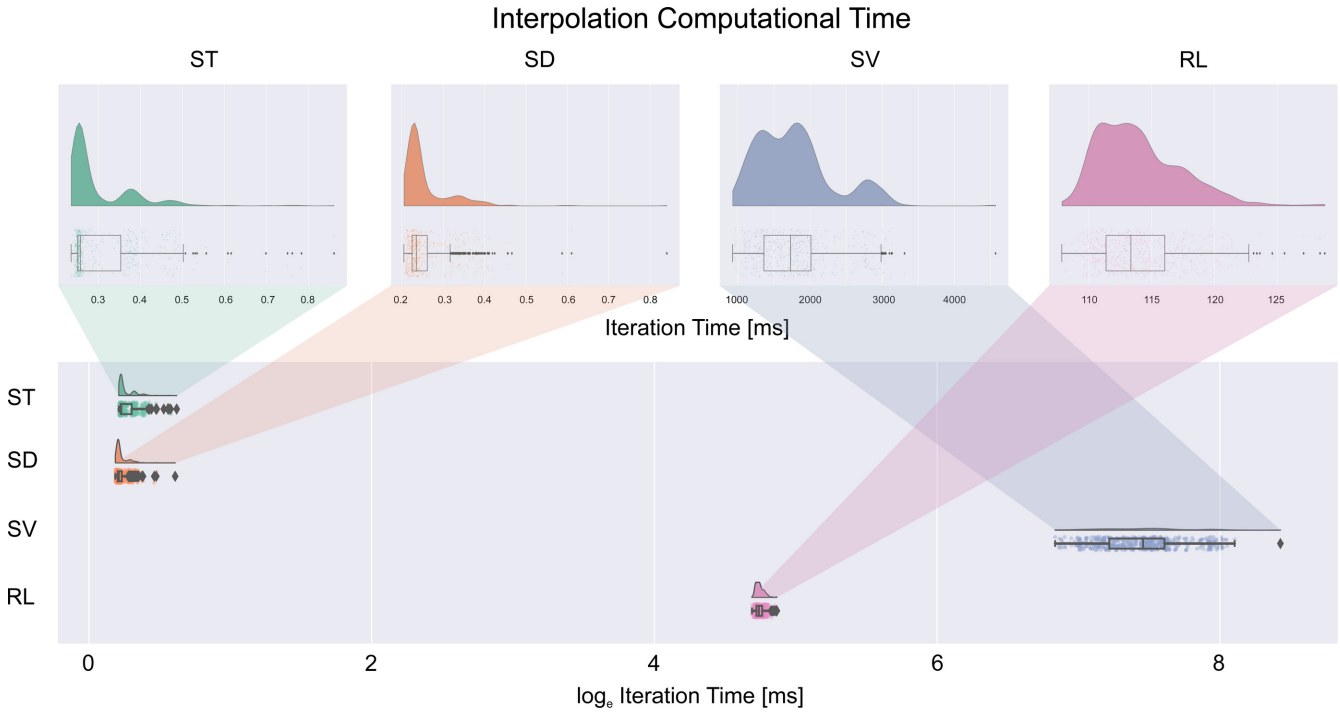


Fig. 5. Rain-cloud and box-whisker plots depicted the distribution of runtime per iteration for each of the four tested interpolation methods. In the top row, the distributions for each individual method are plotted separately, with iteration time on the horizontal axes. The distributions are then shown on the same log time scale in the bottom row. The vertical axis contains the four pose interpolation procedures — ST, SD, SV, RL — and the horizontal axis is a logscale of runtime in ms.

each point $\mathbf{p}_j$ by

$$\mathcal{O}_{VR}(\mathbf{p}_j) = \left(\sum_{i=1}^{N_{\mathbb{C}}} \sum_{k=1}^{N_{\mathbb{K}}} \mathcal{R}(i,k,\mathbf{p}_j)\mathcal{V}(i,k,\mathbf{p}_j)\mathcal{C}_i(k) \right), \quad (23)$$

$$\Gamma_r(s[\tau+1]) = \frac{\mathcal{O}_{VR}(\mathbf{p}_j) \cdot \mathcal{W}(\mathbf{p}_j)}{\sum_{k=1}^{N_{\mathbb{P}}} \mathcal{W}(\mathbf{p}_j)}, \quad (24)$$

where $\mathcal{O}_{VR}(\mathbf{p}_j) \in R^{N_{\mathbb{P}}}$ is first shown in (8) and $\cdot$ is the Euclidean vector dot product. Now consider the distance to goal is reflected by the Γ_g reward component

$$\Gamma_g(s[\tau], s[\tau+1]) = \frac{1-\alpha[\tau]}{N_{\mathbb{C}}} \sum_{k=1}^{N_{\mathbb{C}}} {}_{\tau}\mathcal{O}_G(k) - {}_{\tau+1}\mathcal{O}_{\mathcal{G}}(k). \quad (25)$$

Note that $\alpha \in (0,1)$ is an attenuation scalar, and the pose difference metric ${}_{\tau}\mathcal{O}_g(k)$ is calculated as

$${}_{\tau}\mathcal{O}_g(k) = \|{}_{\tau}g_1(k)\|_2 + \|{}_{\tau}g_2(k) - 1\|_1, \quad (26)$$

where similar to (4),

$${}_{\tau}g_1(k) = \frac{\mathcal{P}(\mathbf{c}_k[\tau]) - \mathcal{P}({}_{\dagger}\mathbf{c}_k)}{d_M}, \quad (27)$$

$${}_{\tau}g_2(k) = \mathcal{Q}(\mathbf{c}_k[\tau]) \cdot \mathcal{Q}({}_{\dagger}\mathbf{c}_k). \quad (28)$$

Finally, the overall reward is the sum of the reward components, i.e.

$$\Gamma = \Gamma_{\alpha} + \Gamma_r + \Gamma_g. \quad (29)$$

Learning network structure

The Double DQN consists of three hidden fully connected hidden layers with dimension (512, 128, 64), respectively. Since the states are preprocessed in a way that there are no significant correlations among features in a local neighborhood, no convolutional layers are added. Recurrent layers are also unnecessary since the breathing cycle time index t_b is already included in the state space. This minimal network structure prevents model overfit and is amenable to the desired timescale.

3. Experiments

For the conducted experiments, a sequence of $N_{\mathbb{E}} = 562$ animated point cloud frames of a deforming abdominal surgical scene was collected. The scene roughly spanned $[-150, 150]$ mm in both X- and Y-directions and $[0, 140]$ mm in the Z-direction. Each frame contained $N_{\mathbb{P}} = 3200$ sampled points. A total of $N_{\mathbb{C}} = 6$ cameras independently moved on a nonstationary hemisphere (modeling the inner abdominal wall), which spanned $[-200, 200]$ mm in both X- and Y-directions. The Z-value of the hemisphere oscillates mimicking a patient's breathing cycle. The camera poses are initialized along the abdominal wall roughly 40 mm away from the top of the hemisphere with equal rotational spacing around the abdomen. Figure 8 depicts the initial camera poses and the experimental surgical scene.

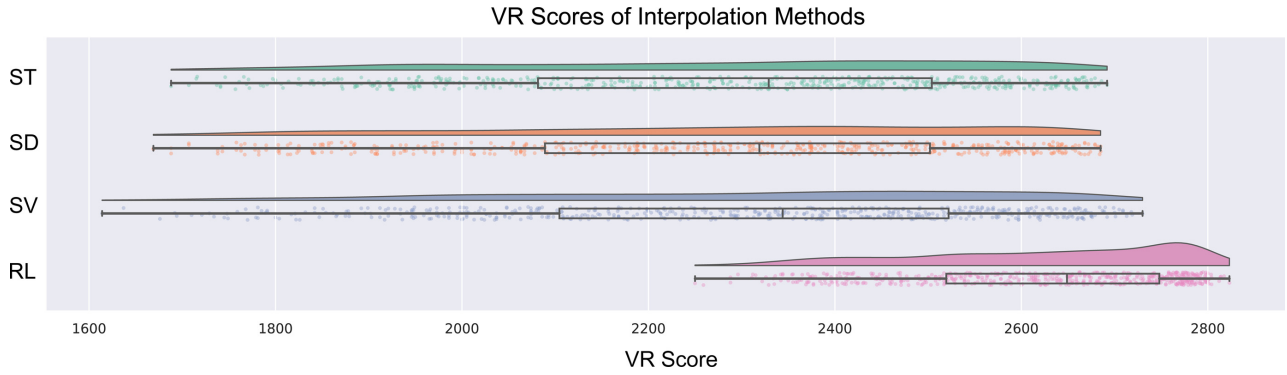


Fig. 6. The comparison of visibility scores per iteration using the four pose interpolation procedures (vertical axis). The horizontal axis represents $\frac{\sum_{j=1}^{N_{\mathbb{P}}} \mathcal{O}_{VR}(\mathbf{p}_j)}{N_{\mathbb{C}}}$. Note that the point cloud sequence contains $N_{\mathbb{P}} = 3200$ per frame.

3.1. *Goal pose derivation*

The goal camera poses are extracted by performing the Advanced Approximation Strategy goal pose optimization procedure iteratively on the 562th point cloud frame until convergence, as mentioned in Sec. 2.2. Details regarding the process are described in the previous work [6]. Table 1 shows the derived start and goal poses of each camera.

Table 1. Initial and goal camera poses.

Pose	Cam	$\mathcal{P}_X$ [mm]	$\mathcal{P}_Y$ [mm]	$\mathcal{P}_Z$ [mm]	$\mathcal{Q}_X$ [rad]	$\mathcal{Q}_Y$ [rad]
Initial	$\mathbf{c}_1$	0.00	40.0	281.8	−0.019	0.37
	$\mathbf{c}_2$	32.0	16.0	282.4	0.149	0.26
	$\mathbf{c}_3$	32.0	−23.0	281.8	0.149	0.05
	$\mathbf{c}_4$	0.00	−48.0	280.6	−0.019	−0.08
	$\mathbf{c}_5$	−40.0	−24.0	280.8	−0.23	0.05
	$\mathbf{c}_6$	−40.0	16.0	281.4	−0.23	0.26
†Goal	†$\mathbf{c}_1$	43.6	82.19	268.9	0.48	0.57
	†$\mathbf{c}_2$	−104.1	75.24	253.7	−0.68	0.47
	†$\mathbf{c}_3$	−191.2	−95.9	186.28	−0.96	−0.09
	†$\mathbf{c}_4$	−153.8	−78.8	225.4	−0.83	0.13
	†$\mathbf{c}_5$	52.4	−103.5	259.6	−0.18	−0.37
	†$\mathbf{c}_6$	163.1	−72.0	221.0	0.93	−0.06

Table 2. Pose interpolation performance.

	Pose interpolation method			
Metric	ST	SD	SV	RL
VR score	0.721	0.720	0.726	0.822
Interpolation time [ms]	3.1×10^{-4}	2.5×10^{-4}	1.75	0.113

3.2. *Pose interpolation*

At every time step, one camera, call it $\mathbf{c}_i$ where $i \in [1, N_{\mathbb{C}}]$, can choose to take either a translational or rotational movement step. A translational step can be in positive or negative X-, *Y*-direction with grid step size of $\frac{d_M}{100}$, whereas a rotational step can be a rotation of $\frac{\pi}{100}$ around the positive or negative *X*- or *Y*-axis. The three slerp baseline procedures — ST, SD, SV — were applied to determine a series of camera motion interpolation actions.

Meanwhile, the proposed double DQN approach RL is implemented under the same experimental conditions. Prior to training, a total of 3,500,000 state transition data steps were collected through random policy π for experimental replay. The model was later trained for a total of 15,000,000 epochs. A sequence of camera motion actions is then determined after the training process.

4. Results

The results from the runtime and visibility score experiments demonstrate improvements in required interpolation time and normalized VR score $\frac{\sum_{j=1}^{N_{\mathbb{P}}} \mathcal{O}_{VR}(\mathbf{p}_j)}{N_{\mathbb{P}} N_{\mathbb{C}}}$. These improvements are summarized in Table 2. The proposed method is promising towards real-time implementation with much improved 3D surgical scene reconstructability over slerp approaches with a modest cost to computational efficiency.

4.1. *Selected actions*

Figure 4 shows a comparison of the list of actions prescribed by the four interpolation procedures. In both ST and RL, each camera roughly shares an equal number of moves; the distribution of motion is fairly uniform across cameras. SD focuses on decreasing the distance to each camera goal pose. Thus, it is prone to focusing only on the furthest camera for a period of time until another camera overtakes it as the furthest from the goal pose. Finally, SV exhibits behavior as the most reluctant to change action near the end of the experiment, or once it discovers the greedy camera move that optimizes visibility.

For further reference, Fig. A.1 shows the camera trajectories resulting from the chosen actions for each motion component of interest. One can observe a preference of RL to maintain relatively neutral X, Y positions (close to 0) compared to the other interpolation methods. This in effect also provides flexibility to move positively in the Z-axis to increase scene coverage. Meanwhile, RL also showcases a tendency to select smaller rotations compared to the slerp approaches; this contributes to a smaller projection angle γ and therefore generally results in a better $\mathbb{R}$ score. These phenomena are apparent especially for cameras 2 and 5 in Fig. A.1.

4.2. *Runtime analysis*

Figure 5 shows the iteration runtime statistics. The median runtime for method ST, SD, SV, RL is roughly 0.31, 0.25, 1750, and 113 ms, respectively. ST and SD exhibit the best runtime figures, as both SV and RL execute additional processes to optimize visibility. While RL iterations require more time than ST, SD, they are over an order of magnitude more efficient than SV. RL also shows the best runtime consistency (smallest runtime variance). This characteristic is ideal for real-time application. Furthermore, computational efficiency gains are likely with the incorporation of hardware acceleration methods.

4.3. *Visibility analysis*

The aggregate visibility $\frac{\sum_{j=1}^{N_{\mathbb{P}}} \mathcal{O}_{VR}(\mathbf{p}_j)}{N_{\mathbb{C}}} \in [0.0, 1.0]$ for each iteration is depicted in Fig. 6. Each point on the graph corresponds to the overall visibility of the entire surgical scene given the current camera configurations. Method RL shows the highest visibility median and smallest variance, which again demonstrates the consistency of the proposed algorithm compared with the slerp baselines. Figure 7 lists related weighting and visibility metrics. Figure 9 shows the correlated distribution of weighting function $\mathcal{W}(\mathbf{p}_j)$ and visibility score $\mathcal{O}_{VR}(\mathbf{p}_j)$. This is shown for each point $j \in [1, N_{\mathbb{P}}]$ at frame 222. The color reflects the frequency of points in that part of the distribution. Note that since there are a total of $N_{\mathbb{C}} = 6$

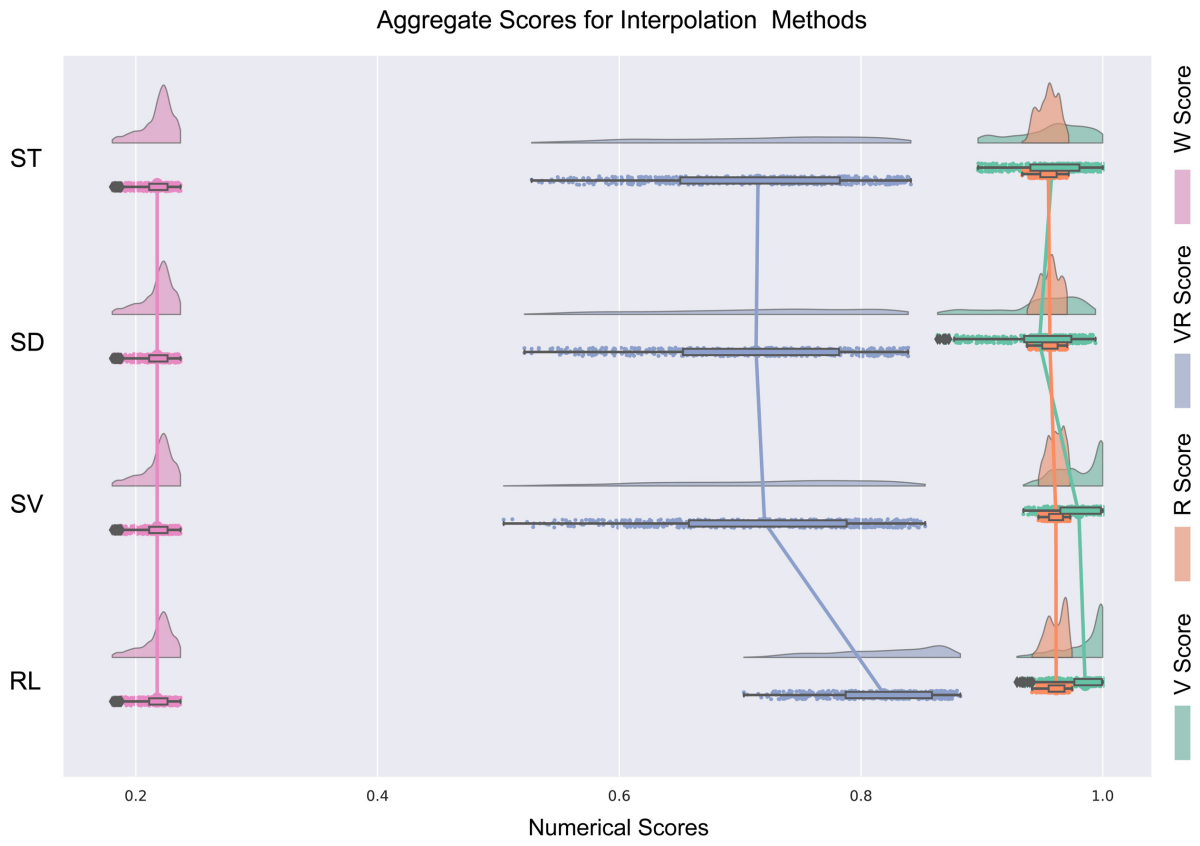


Fig. 7. The statistical comparison of four score types using different pose interpolation procedures (vertical axis). The horizontal axis represents score values. The normalized V score (green) $= \frac{\sum_{i=1}^{N_{\mathbb{C}}} \sum_{k=1}^{N_{\mathbb{K}}} \sum_{j=1}^{N_{\mathbb{P}}} \mathcal{V}(i,k,\mathbf{p}_j)\mathcal{C}_i(k)}{N_{\mathbb{P}} N_{\mathbb{C}}}$; the normalized R score (orange) $\frac{\sum_{i=1}^{N_{\mathbb{C}}} \sum_{k=1}^{N_{\mathbb{K}}} \sum_{j=1}^{N_{\mathbb{P}}} \mathcal{R}(i,k,\mathbf{p}_j)\mathcal{C}_i(k)}{N_{\mathbb{P}} N_{\mathbb{C}}}$; the normalized VR score (blue) $\frac{\sum_{j=1}^{N_{\mathbb{P}}} \mathcal{O}_{VR}(\mathbf{p}_j)}{N_{\mathbb{P}} N_{\mathbb{C}}}$; finally the normalized W score (pink) $\frac{\sum_{j=1}^{N_{\mathbb{P}}} \mathcal{W}(\mathbf{p}_j)}{d_M}$, where d_M is the maximum distance in the surgical scene (in mm).

cameras, the maximum visibility score $\mathcal{O}_{VR}(\mathbf{p}_j)$ for any given point is capped at 6. The higher the $\mathcal{O}_{VR}(\mathbf{p}_j)$, the greater chance that point will be visible multiple cameras and hence 3D reconstructed. The distribution of $\mathcal{W}(\mathbf{p}_j)$, on the other hand, reflects the relative importance of viewing and reconstructing a point, which is highly correlated to proximity to the tool tip location (thus the distribution of $\mathcal{W}(\mathbf{p}_j)$ is the same across methods). More visibility would result in more points near the top of the map. Considering the weighting function, ideally more points will appear at the top right corner (both high importance and high visibility) and only a few points in the bottom right corner (both high importance and poor visibility). Based on these criteria, RL method significantly outperforms the other methods. Out of the three slerp approaches, SV also shows a more desirable result compared with ST and SD.

The reconstructed points from the same surgical scene and initial and goal poses using the tested interpolation methods are depicted in Fig. 10. The total number of reconstructed points during the interpolation is shown in Table 3. The RL interpolation method demonstrates a greater reconstructability, and is consistent with the greater VR scores.

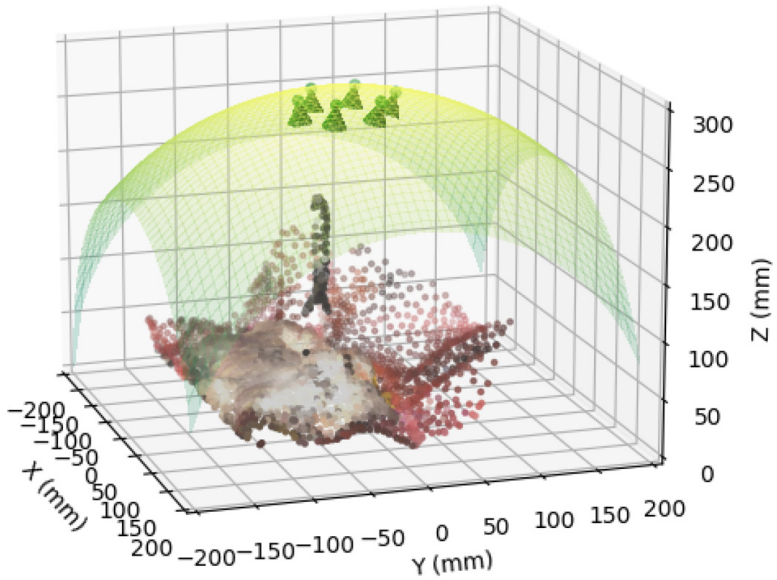


Fig. 8. The dynamic surgical scene, with initial poses of the six cameras marked with green cones. The abdominal wall, shown as the translucent green surface, translates periodically in the *Z*-direction to mimic patient breathing.

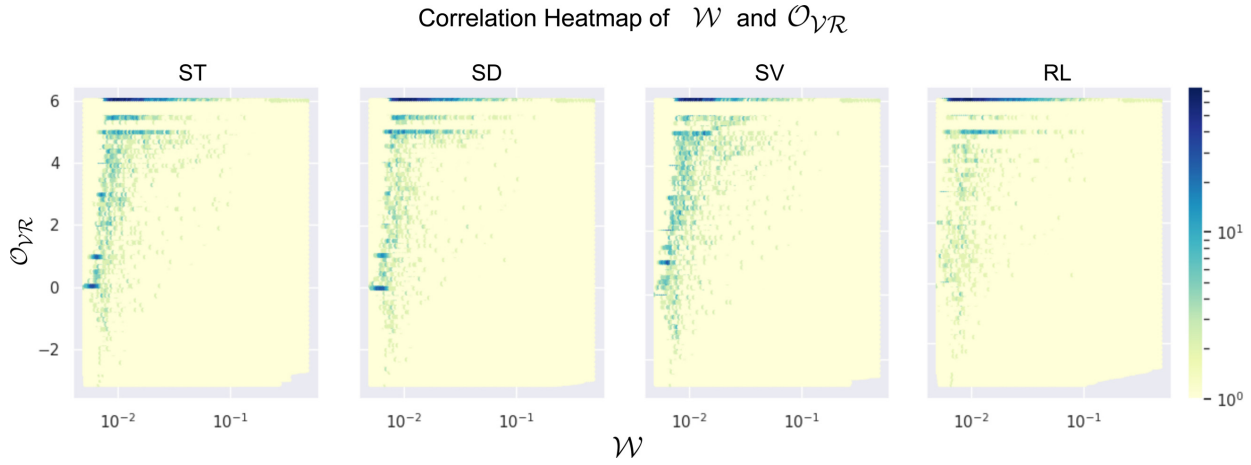


Fig. 9. The correlation heatmap between the point importance weights $\mathcal{W}(\mathbf{p}_j)$ and corresponding visibility scores $\mathcal{O}_{VR}(\mathbf{p}_j)$ using the four interpolation procedures.

Table 3. Reconstruction performance.

	Pose interpolation method			
	ST	SD	SV	RL
Aggregate points	1,385,073	1,383,188	1,393,485	1,577,422

5. Conclusion

Previously, the authors introduced a novel Advanced Approximation Strategy, $\mathcal{A}$, and assessed its viability compared with an exhaustive search approach. Relative improvements were stark, and insights as to the effects of increased camera number were also presented [6]. While accuracy and high reconstructability were observed, true real-time deployment is a challenge. To that end, this work presented a mixed approach, whereby iterations of goal camera poses are calculated via $\mathcal{A}$, and streamlined multicamera movement methods interpolate between assigned start and goal poses. The authors present a deep RL approach and compare with linear spherical interpolation. Cameras are initialized, as depicted in Fig. 8, and a goal pose is calculated using $\mathcal{A}$, as shown in Table 1.

The different slerp approaches, ST, SD, and SV, prescribe camera movements in various order. ST adheres to a rigorous turn-taking approach, as observed in Fig. 4. SD sought to prioritize movement of cameras furthest from the goal pose. Finally, SV chose camera slerp motions greedily to maximize (3). These three interpolation methods disregard any potentially better views between the initial and goal poses; movement candidates are predetermined by the initial and goal poses. The authors introduced a deep RL agent, RL, which is presented with the same task as the above methods. However, this method is trained to maximize longer term visibility while also serving cameras to the goal poses. As shown in Fig. 6, RL outperformed the linear interpolation methods for overall visibility score. The number of reconstructed points from the same surgical scene, camera configuration, and goal pose were tracked for all four methods, and are depicted in Fig. 10, and summarized in Table 3. The proposed RL method demonstrated greater reconstructability than the other approaches. The method comes at a cost of computational efficiency (iteration time of the method was measured at about 113 ms), yet is more efficient by over an order of magnitude over SV, as illustrated in Fig. 5. These results are also summarized in Table 2.

The work presented here introduces a deep RL method for adjusting multicamera viewpoints in a dynamic surgical scene to navigate from one set of camera poses to another. Results are encouraging, and suggest that when compared to linear interpolation methods, the proposed technique achieves much greater en route reconstructability visibility while introducing acceptable latency. Previous work offered a promising method for deriving optimal camera poses at a coarser time scale. This work shows that at a finer temporal scale, navigating multicamera systems in RMIS should still utilize intelligent movement algorithms such as the one presented here.

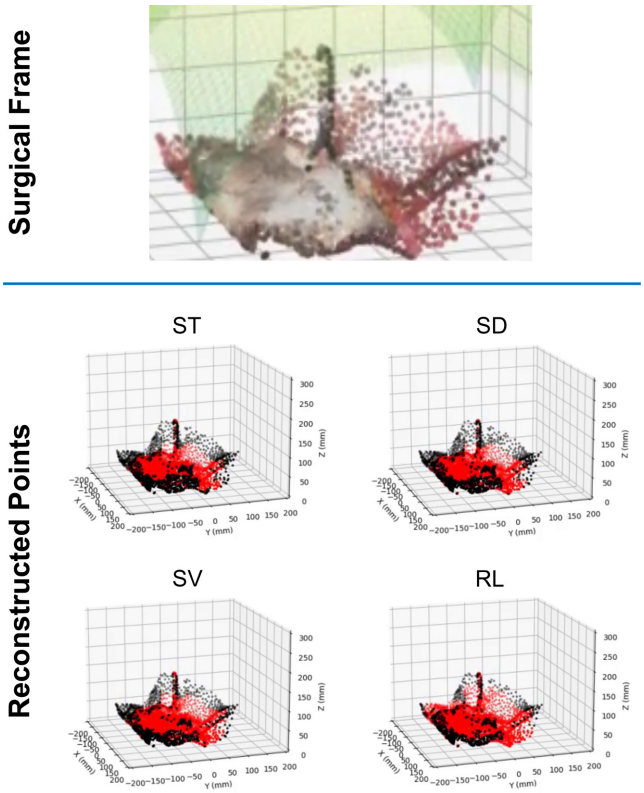


Fig. 10. Reconstructed points are depicted in red for the same surgical scene and start and end poses. The proposed RL interpolation method exhibits greater reconstructability score and number of reconstructed points.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. IIS-2101107. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Appendix A. Interpolation Trajectories

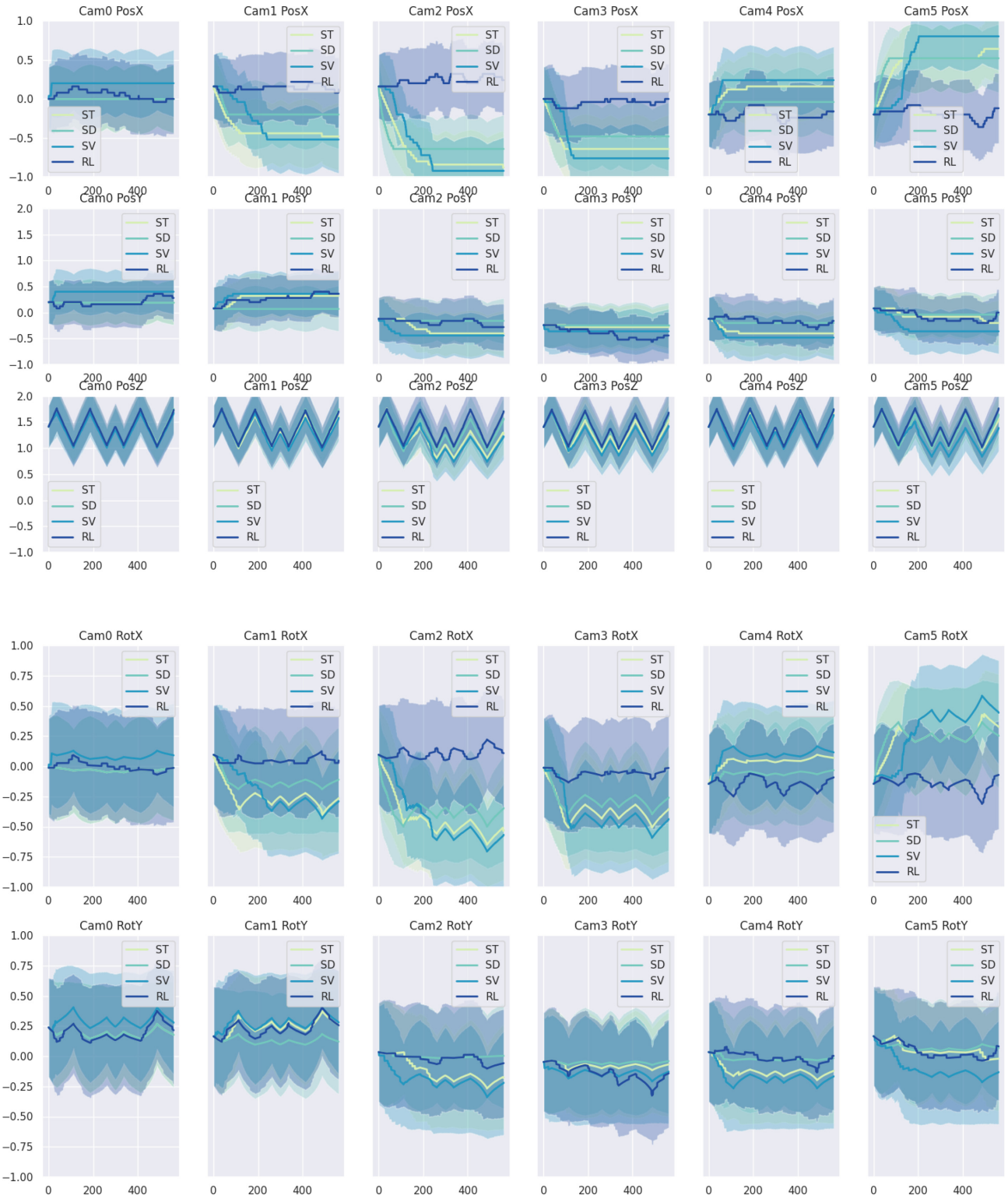
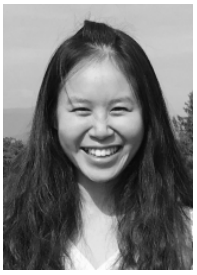


Fig. A.1. Recorded camera trajectories. In each graph, the horizontal axis denotes the evaluation point cloud frame $\tau \in [1, N_{\mathbb{E}}]$ and the vertical axis shows normalized poses. Subgraphs in the same row illustrate the same normalized camera pose component, whereas each column shows motion subgraphs of the same camera. The colored areas shows the normalized VR score at that particular iteration.

References

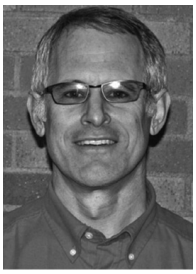
1. K. Fuchs, Minimally invasive surgery, *Endoscopy* **34**(02) (2002) 154–159.
2. A. Pandya, L. Reisner, B. King, N. Lucas, A. Composto, M. Klein and R. Ellis, A review of camera viewpoint automation in robotic and laparoscopic surgery, *Robotics* **3**(3) (2014) 310–329.
3. O. Weede, H. Mönnich, B. Müller and H. Wörn, An intelligent and autonomous endoscopic guidance system for minimally invasive surgery, in *2011 IEEE Int. Conf. Robotics and Automation* (IEEE, 2011), pp. 5762–5768.
4. K.-T. Song and C.-J. Chen, Autonomous and stable tracking of endoscope instrument tools with monocular camera, in *2012 IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics (AIM)* (IEEE, 2012), pp. 39–44.
5. L. Yu, Z. Wang, L. Sun, W. Wang and T. Wang, A kinematics method of automatic visual window for laparoscopic minimally invasive surgical robotic system, in *2013 IEEE Int. Conf. Mechatronics and Automation* (IEEE, 2013), pp. 997–1002.
6. Y. H. Su, K. Huang and B. Hannaford, Multicamera 3D reconstruction of dynamic surgical cavities: Autonomous optimal camera viewpoint adjustment, in *2020 Int. Symp. Medical Robotics (ISMR)* (IEEE, 2020), pp. 1–11.
7. M. Hu, G. P. Penney, D. Rueckert, P. J. Edwards, F. Bello, R. Casula, M. Figl and D. J. Hawkes, Non-rigid reconstruction of the beating heart surface for minimally invasive cardiac surgery, in *Int. Conf. Medical Image Computing and Computer-Assisted Intervention* (Springer, 2009), pp. 34–42.
8. Y. H. S. Isaac Huang, K. Huang and B. Hannaford, Comparison of 3D surgical tool segmentation procedures with robot kinematics prior, in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS) 2018* (IEEE, 2018), pp. 4411–4418.
9. Y. H. Su, K. Huang and B. Hannaford, Real-time vision-based surgical tool segmentation with robot kinematics prior, in *2018 Int. Symp. Medical Robotics (ISMR)* (IEEE, 2018), pp. 1–6.
10. Y. H. Su, K. Huang and B. Hannaford, Multicamera 3D reconstruction of dynamic surgical cavities: Camera grouping and pair sequencing, in *2019 Int. Symp. Medical Robotics (ISMR)* (IEEE, 2019), pp. 1–7.
11. O. G. Grasa, J. Civera and J. Montiel, EKF monocular SLAM with relocalization for laparoscopic sequences, in *2011 IEEE Int. Conf. Robotics and Automation (ICRA)* (IEEE, 2011), pp. 4816–4821.
12. T. Collins, B. Compte and A. Bartoli, Deformable shape-from-motion in laparoscopy using a rigid sliding window, in *Proc. Medical Image Understanding and Analysis (MIUA)* (2011), pp. 173–178.
13. H. Urey, K. V. Chellappan, E. Erden and P. Surman, State of the art in stereoscopic and autostereoscopic displays, *Proc. IEEE* **99**(4) (2011) 540–555.
14. M. Silvestri, T. Ranzani, A. Argiolas, M. Vatteroni and A. Menciassi, A multi-point of view 3D camera system for minimally invasive surgery, *Sens. Actuators A, Phys.* **202** (2013) 204–210.
15. G. Yin, W. K. Han, S. Faddegon, Y. K. Tan, Z.-W. Liu, E. O. Olweny, D. J. Scott and J. A. Cadeddu, Laparoendoscopic single site (LESS) *in vivo* suturing using a magnetic anchoring and guidance system (MAGS) camera in a porcine model: Impact on ergonomics and workload, *Urology* **81**(1) (2013) 80–84.
16. M. Morgan, E. O. Olweny and J. A. Cadeddu, Less and notes instrumentation: Future, *Curr. Opin. Urol.* **24**(1) (2014) 58–65.
17. N. Di Lorenzo, L. Cenci, M. Simi, C. Arcudi, V. Tognoni, A. L. Gaspari and P. Valdastri, A magnetic levitation robotic camera for minimally invasive surgery: Useful for notes?, *Surg. Endosc.* **31**(6) (2017) 2529–2533.
18. P. Valdastri *et al.*, A magnetic internal mechanism for precise orientation of the camera in wireless endoluminal applications, *Endoscopy* **42**(6) (2010) 481.
19. X. Bonaventura, M. Feixas, M. Sbert, L. Chuang and C. Wallraven, A survey of viewpoint selection methods for polygonal models, *Entropy* **20**(5) (2018) 370.
20. C. Song, L. Liu, G. Feng, Y. Fan and S. Xu, Coverage control for heterogeneous mobile sensor networks with bounded position measurement errors, *Automatica* **120** (2020) 109118.
21. R. Almadhoun, T. Taha, L. Seneviratne, J. Dias and G. Cai, A survey on inspecting structures using robotic systems, *Int. J. Adv. Robot. Syst.* **13**(6) (2016) 1729881416663664.
22. S. D. Roy, S. Chaudhury and S. Banerjee, Active recognition through next view planning: A survey, *Pattern Recognit.* **37**(3) (2004) 429–446.
23. M. Krainin, B. Curless and D. Fox, Autonomous generation of complete 3D object models using next best view manipulation planning, in *2011 IEEE Int. Conf. Robotics and Automation* (IEEE, 2011), pp. 5031–5037.
24. J. I. Vasquez-Gomez, L. E. Sucar, R. Murrieta-Cid and E. Lopez-Damian, Volumetric next-best-view planning for 3D object reconstruction with positioning error, *Int. J. Adv. Robot. Syst.* **11**(10) (2014) 159.
25. J. I. Vásquez and L. E. Sucar, Next-best-view planning for 3D object reconstruction under positioning error, in *Mexican International Conf. Artificial Intelligence* (Springer, 2011), pp. 429–442.
26. E. Dunn and J.-M. Frahm, Next best view planning for active model improvement, in *British Machine Vision Conf. (BMVC)* (BMVC, 2009), pp. 1–11.
27. E. Dunn, J. Van Den Berg and J.-M. Frahm, Developing visual sensing strategies through next best view planning, in *2009 IEEE/RSJ Int. Conf. Intelligent Robots and Systems* (IEEE, 2009), pp. 4001–4008.
28. J. Tisdale, A. Ryan, Z. Kim, D. Tornqvist and J. K. Hedrick, A multiple UAV system for vision-based search and localization, in *2008 American Control Conf.* (IEEE, 2008), pp. 1985–1990.
29. T. Furukawa, F. Bourgault, B. Lavis and H. F. Durrant-Whyte, Recursive Bayesian search-and-tracking using coordinated UAVs for lost targets, in *Proc. 2006 IEEE Int. Conf. Robotics and Automation, 2006 (ICRA 2006)* (IEEE, 2006), pp. 2521–2526.
30. J. Hu, L. Xie and J. Xu, Vision-based multi-agent cooperative target search, in *2012 12th Int. Conf. Control Automation Robotics & Vision (ICARCV)* (IEEE, 2012), pp. 895–900.
31. G. Chmaj and H. Selvaraj, Distributed processing applications for UAV/drones: A survey, in *Progress in Systems Engineering* (Springer, 2015), pp. 449–454.
32. K. McGuire, C. De Wagter, K. Tuyls, H. Kappen and G. de Croon, Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment, *Sci. Robot.* **4**(35) (2019) eaaw9710.
33. H. Chao, M. Baumann, A. Jensen, Y. Chen, Y. Cao, W. Ren and M. McKee, Band-reconfigurable multi-UAV-based cooperative remote sensing for real-time water management and distributed irrigation control, *IFAC Proc. Vol.* **41**(2) (2008) 11744–11749.
34. P. Odonkor, Z. Ball and S. Chowdhury, Distributed operation of collaborating unmanned aerial vehicles for time-sensitive oil spill mapping, *Swarm Evol. Comput.* **46** (2019) 52–68.
35. B. Argrow, D. Lawrence and E. Rasmussen, UAV systems for sensor dispersal, telemetry, and visualization in hazardous environments, in *43rd AIAA Aerospace Sciences Meeting and Exhibit* (2005), p. 1237.
36. B. Lin, Y. Sun, X. Qian, D. Goldgof, R. Gitlin and Y. You, Video-based 3D reconstruction, laparoscope localization and deformation recovery for abdominal minimally invasive surgery: A survey, *Int. J. Med. Robot. Comput. Assist. Surg.* **12**(2) (2016) 158–178.
37. B. Tamadazte, A. Agustinos, P. Cinquin, G. Fiard and S. Voros, Multi-view vision system for laparoscopy surgery, *Int. J. Comput. Assist. Radiol. Surg.* **10**(2) (2015) 195–203.

38. J.-J. Kim, A. Watras, H. Liu, Z. Zeng, J. A. Greenberg, C. P. Heise, Y. H. Hu and H. Jiang, Large-field-of-view visualization utilizing multiple miniaturized cameras for laparoscopic surgery, *Micromachines* **9**(9) (2018) 431.
39. A. Kanhere, K. L. Van Grinsven, C.-C. Huang, Y.-S. Lu, J. A. Greenberg, C. P. Heise, Y. H. Hu and H. Jiang, Multicamera laparoscopic imaging with tunable focusing capability, *J. Microelectromech. Syst.* **23**(6) (2014) 1290–1299.
40. A. Afifi, C. Takada, Y. Yoshimura and T. Nakaguchi, Real-time expanded field-of-view for minimally invasive surgery using multi-camera visual simultaneous localization and mapping, *Sensors* **21**(6) (2021) 2106.
41. X. Ma, C. Song, P. W. Chiu and Z. Li, Autonomous flexible endoscope for minimally invasive surgery with enhanced safety, *IEEE Robot. Autom. Lett.* **4**(3) (2019) 2607–2613.
42. J. M. Prendergast, G. A. Formosa, M. J. Fulton, C. R. Heckman and M. E. Rentschler, A real-time state dependent region estimator for autonomous endoscope navigation, *IEEE Trans. Robot.* (IEEE, 2020), pp. 1–17.
43. J. W. Martin, B. Scaglioni, J. C. Norton, V. Subramanian, A. Arezzo, K. L. Obstein and P. Valdastri, Enabling the future of colonoscopy with intelligent and autonomous magnetic manipulation, *Nature Mach. Intell.* **2**(10) (2020) 595–606.
44. T. Da Col, A. Mariani, A. Deguet, A. Menciassi, P. Kazanzides and E. De Momi, SCAN: System for camera autonomous navigation in robotic-assisted surgery, in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS) 2020* (IEEE, 2020), pp. 2996–3002.
45. A. Giaccone, P. Solli and L. Bertolaccini, Magnetic anchoring guidance system in video-assisted thoracic surgery, *J. Vis. Surg.* **3** (2017) 17, doi:10.21037/jovs.2017.01.13.
46. Y. H. S. Kevin Huang and B. Hannaford, Multicamera 3D reconstruction of dynamic surgical cavities: Non-rigid registration and point classification, in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS) 2019* (2019).
47. R. Cohen and L. Katzir, The generalized maximum coverage problem, *Inf. Process. Lett.* **108**(1) (2008) 15–22.
48. L. Tao, E. Elhamifar, S. Khudanpur, G. D. Hager and R. Vidal, Sparse hidden Markov models for surgical gesture classification and skill evaluation, in *Int. Conf. Information Processing in Computer-assisted Interventions* (Springer, 2012), pp. 167–177.
49. N. Ahmidi, L. Tao, S. Sefati, Y. Gao, C. Lea, B. B. Haro, L. Zappella, S. Khudanpur, R. Vidal and G. D. Hager, A dataset and benchmarks for segmentation and recognition of gestures in robotic surgery, *IEEE Trans. Biomed. Eng.* **64**(9) (2017) 2025–2041.
50. L. Tao, L. Zappella, G. D. Hager and R. Vidal, Surgical gesture segmentation and recognition, in *Int. Conf. Medical Image Computing and Computer-Assisted Intervention* (Springer, 2013), pp. 339–346.
51. K. Lindgren, K. Huang and B. Hannaford, Towards real-time surface tracking and motion compensation integration for robotic surgery, in *2017 IEEE/SICE Int. Symp. System Integration (SII)* (IEEE, 2017), pp. 450–456.
52. T. Ortmaier, M. Groger, D. H. Boehm, V. Falk and G. Hirzinger, Motion estimation in beating heart surgery, *IEEE Trans. Biomed. Eng.* **52**(10) (2005) 1729–1740.
53. J. Cardenas-Garcia, H. Yao and S. Zheng, 3D reconstruction of objects using stereo imaging, *Opt. Lasers Eng.* **22**(3) (1995) 193–213.
54. D. Stoyanov and G. Z. Yang, Removing specular reflection components for robotic assisted laparoscopic surgery, in *IEEE Int. Conf. Image Processing 2005*, Vol. 3 (IEEE, 2005), pp. III–632.
55. J. D. Stefansic, A. J. Herline, Y. Shyr, W. C. Chapman, J. M. Fitzpatrick, B. M. Dawant and R. L. Galloway, Registration of physical space to laparoscopic image space for use in minimally invasive hepatic surgery, in *Proc. 5th IEEE EMBS Int. Summer School on Biomedical Imaging, 2002* (IEEE, 2002), p. 12.
56. K. Shoemake, Animating rotation with quaternion curves, in *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques* (1985), pp. 245–254.
57. Y. Li, Deep reinforcement learning: An overview, arXiv:1701.07274.
58. M. Wiering and M. Van Otterlo, *Reinforcement Learning* (Springer, 2012).
59. J. D. R. Millán, D. Posenato and E. Dedieu, Continuous-action Q-learning, *Mach. Learn.* **49**(2–3) (2002) 247–265.
60. V. Mnih *et al.*, Human-level control through deep reinforcement learning, *Nature* **518**(7540) (2015) 529–533.
61. S. Zhang and R. S. Sutton, A deeper look at experience replay, arXiv:1712.01275.
62. T. Schaul, J. Quan, I. Antonoglou and D. Silver, Prioritized experience replay, arXiv:1511.05952.
63. M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. P. Abbeel and W. Zaremba, Hindsight experience replay, in *Advances in Neural Information Processing Systems* (2017), pp. 5048–5058.
64. H. Van Hasselt, A. Guez and D. Silver, Deep reinforcement learning with double Q-learning, arXiv:1509.06461.



Yun-Hsuan Su received the B.S. degree with concentration in Systems and Controls in Electrical and Computer Engineering from National Chiao Tung University, Hsinchu, Taiwan in 2016. She received her graduate degrees, the M.S. in Electrical Engineering and Ph.D. in Electrical and Computer Engineering, with focus on Systems, Controls and Robotics from the University of Washington, Seattle, WA in 2017 and 2020, respectively. In 2018, Dr. Su was a Research Engineer and worked on visual and force servoing for industrial robots at ABB robotics, leading to two patent applications. She is passionate about outreach STEM programs, has been closely involved in IEEE TryEngineering, and has organized and led multiple summer robotics camps. Currently, she is an Assistant Professor in the Department of Computer Science at Mount Holyoke College. She is a member of Tau Beta Pi, and was nominated for the Yang Award for Outstanding Doctoral Student. Dr. Su's research interests include span surgical robotics, vision-based force estimation, computer/machine vision, and haptic feedback.

Kevin Huang received the B.S. degree in Engineering and Mathematics from Trinity College, Hartford, CT, USA in 2012. He received his M.S. degree in Electrical Engineering and his Ph.D. degree in Electrical Engineering with a concentration in Systems, Controls and Robotics from the University of Washington, Seattle, WA in 2015 and 2017, respectively. During graduate studies, Dr. Huang received the National Science Foundation's Graduate Research Fellowship. Currently, he is an Assistant Professor in the Department of Engineering at Trinity College, where he is active in robotics outreach and undergraduate engineering education. He is an Affiliate Professor at the University of Washington and a Research Associate at Mount Holyoke College, and serves as the Chair for the IEEE Connecticut Robotics and Automation Society. Dr. Huang's research interests include haptic virtual fixtures, telerobotics, surgical robotics, and human–robot interaction.



Blake Hannaford (F'05) received the B.S. degree in Engineering and Applied Science from Yale University, New Haven, CT, USA, in 1977, and the M.S. and Ph.D. degrees in Electrical Engineering from University of California, Berkeley, CA, USA. He is currently a Professor of Electrical and Computer Engineering and an Adjunct Professor of Bioengineering, Mechanical Engineering, and Surgery with the University of Washington, Seattle, WA, USA. From 1986 to 1989, he worked on the remote control of robot manipulators in the Man–Machine Systems Group in the Automated Systems Section of the NASA Jet Propulsion Laboratory, Caltech, and supervised that group from 1988 to 1989. Since September 1989, he has been with University of Washington. His research interests include surgical robotics, surgical skill modeling, and haptic interfaces. He is currently splitting his time between UW and a spinout company called Applied Dexterity Inc. Dr. Hannaford received the National Science Foundation's Presidential Young Investigator Award and the Early Career Achievement Award from the IEEE Engineering in Medicine and Biology Society.