

A Case Study of Trajectory Transfer Through Non-Rigid Registration for a Simplified Suturing Scenario

John Schulman Ankush Gupta Sibi Venkatesan Mallory Tayson-Frederick Pieter Abbeel

Abstract—Suturing is an important yet time-consuming part of surgery. A fast and robust autonomous procedure could reduce surgeon fatigue, and shorten operation times. It could also be of particular importance for suturing in remote tele-surgery settings where latency can complicate the master-slave mode control that is the current practice for robotic surgery with systems like the da Vinci[®].

We study the applicability of the *trajectory transfer* algorithm proposed in [12] to the automation of suturing. The core idea of this procedure is to first use non-rigid registration to find a 3D warping function which maps the demonstration scene onto the test scene, then use this warping function to transform the robot end-effector trajectory. Finally a robot joint trajectory is generated by solving a trajectory optimization problem that attempts to find the closest feasible trajectory, accounting for external constraints, such as joint limits and obstacles.

Our experiments investigate generalization from a single demonstration to differing initial conditions. A first set of experiments considers the problem of having a simulated Raven II system [5] suture two flaps of tissue together. A second set of experiments considers a PR2 robot performing sutures in a scaled-up experimental setup. The simulation experiments were fully autonomous. For the real-world experiments we provided human input to assist with the detection of landmarks to be fed into the registration algorithm. The success rate for learning from a single demonstration is high for moderate perturbations from the demonstration’s initial conditions, and it gradually decreases for larger perturbations.

I. INTRODUCTION

Robotic systems for minimally invasive surgery, such as the da Vinci[®] system, are becoming increasingly widespread. Currently, these systems are operated in master-slave mode—the surgeon completely controls the movement of the robot. There are several reasons why it would be useful to perform surgical subtasks such as suturing autonomously. First, some automation of repetitive tasks would give the surgeon some rest and reduce fatigue. Second, computer control could enable these tasks to be performed faster and more precisely by overcoming the inherent limitations of the human nervous system for speed and precision of motion. Third, master-slave mode tele-operation performance degrades with increased transmissions delays. Autonomous low-level control could enable tele-surgery in the presence of longer transmission delays by having the surgeon intermittently issues high-level commands, like “cut here”, “suture here”, which would then be controlled at a finer timescale by software.

Suturing, along with other surgical tasks, requires a long series of complex motions, and the margin of error is

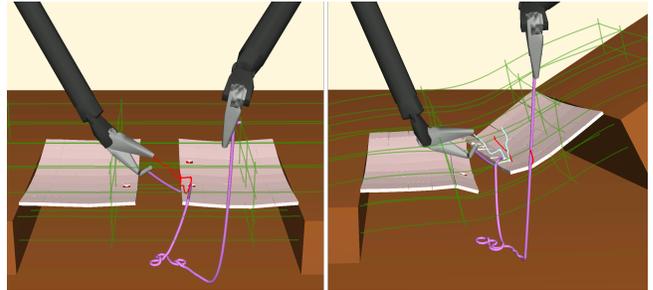


Fig. 1. Image on the left shows the suturing setup as it was in the demonstration. Image on the right shows a test scene in which the right suturing pad is in a different location. The grid visualizes the warping function, which is used to transfer the demonstrated trajectory to the test setting. The grid on the left is uniform box-like, showing the identity transform, while it is non-rigidly bent and rotated on the right. The suture is shown in pink. The training scene shows the demonstrated trajectory (in red). The test scene shows the transferred trajectory (in light blue) as well as the original demonstrated trajectory (in red).

small. The operating environment, including the tissue to be sutured, may vary widely in position, shape, and material properties.

Building on recent advances in learning from demonstrations and trajectory optimization, this paper presents promising results that might bring us closer to real-world autonomous suturing capabilities. At the core of our procedure is the *trajectory transfer* algorithm recently proposed by the authors in [12], which takes trajectories from human demonstrations and adapts them to a new environment geometry.

Our experiments consider a simplified suturing scenario, both with a simulated Raven II system [5] and with a real PR2, and show that a single demonstration (provided by a human) can be generalized to a variety of initial conditions.

Videos of our results are available at: <http://rll.berkeley.edu/iros2013suturing>

II. RELATED WORK

Some of the earliest work on automating laparoscopic procedures was performed by Kang et al. [6], [7]. They focused on the mechanical design and low-level control of a surgical robot. Recent work has addressed tying knots in the surgical setting based on human demonstrations. Mayer et al. [8] used ideas from fluid dynamics to modify trajectories to avoid obstacles. Mayer et al. [9] used a recurrent neural network to as a controller for part of a surgical knot-tying task, based on demonstrations. van den Berg et al. [14] used an iterative learning procedure to learn to tie an overhand

Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, CA, USA. {joschu, gupta, sibi, mtaysonfrederick, pabbeel}@berkeley.edu

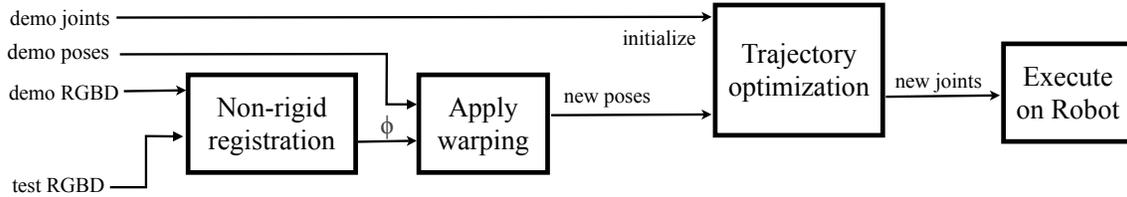


Fig. 2. Schematic diagram of the procedure applied at testing time for each trajectory segment. “Demo” is short for “demonstration”. “Poses” and “joints” refer to pose trajectories and joint trajectories, respectively.

knot rapidly with an imprecise robot. Padoy and Hager [10] automated several subtasks in surgery, such as pulling a needle out of tissue, and they developed a system that automatically recognizes from the surgeon’s motion when these subtasks can be performed. Staub et al. [13] described a method to automate needle piercing in laparoscopic surgery. Their technique involves using a laser pointer to specify the stitching area and a switching visual servoing approach for positioning the needle precisely.

There is a large body of work on learning from demonstrations. In this work we perform an empirical study of the applicability of the trajectory transfer approach presented in [12] to the task of suturing. We refer the reader to this work for a discussion on its relationship to other work on learning from demonstrations.

III. BACKGROUND: TRAJECTORY TRANSFER THROUGH NON-RIGID REGISTRATION

Our method for generating end-effector trajectories in this paper is based on the previous work of the authors in [12]. This section reviews that method; the reader is referred to the original paper for a more complete exposition.

The overall pipeline is illustrated in Figure 2, and consists of three steps:

Step 1: Find a transformation \mathbf{f} from the demonstration scene to the test scene.

We assume that there is a list of 3D points $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K$ in the demonstration scene and a list of 3D points $\mathbf{x}'_1, \mathbf{x}'_2, \dots, \mathbf{x}'_L$ in the test scene. When starting out, however, their correspondence is unknown. We use the method of Thin Plate Spline Robust Point Matching [1] to find both their correspondences, encoded by an index-mapping function $\Pi(\cdot)$ (which maps from indices of points in demonstration scene to indices of points in the test scene) and a warping function \mathbf{f} mapping $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K$ to their partners $\mathbf{x}'_{\Pi(1)}, \mathbf{x}'_{\Pi(2)}, \dots, \mathbf{x}'_{\Pi(K)}$.

The method of thin plate splines [4], [15] solves the following optimization problem:

$$\underset{\mathbf{f}, \Pi}{\text{minimize}} \sum_{i=1}^K \|\mathbf{x}'_{\Pi(i)} - \mathbf{f}(\mathbf{x}_i)\|^2 + \text{REGULARIZER}(\mathbf{f}) \quad (1)$$

\mathbf{f} is required to be an expansion in terms of radial basis functions

$$\mathbf{f}(\mathbf{x}) = \sum_{i=1}^K \mathbf{a}_i K(\mathbf{x}, \mathbf{x}_i) + \mathbf{B}\mathbf{x} + \mathbf{c} \quad (2)$$

where K is the 3D thin plate spline kernel $K(\mathbf{x}, \mathbf{x}') = -\|\mathbf{x} - \mathbf{x}'\|$. The regularization term takes the form

$$\text{REGULARIZER}(\mathbf{f}) = \lambda \text{tr}(\mathbf{A}^T \mathbf{K} \mathbf{A}) + \beta \|\log \mathbf{s}\|^2 \quad (3)$$

where \mathbf{s} is the vector of singular values of \mathbf{B} ; \mathbf{K} is the kernel matrix, with entries $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$; $\mathbf{A} = \begin{pmatrix} \mathbf{a}_1^T \\ \vdots \\ \mathbf{a}_K^T \end{pmatrix}$;

λ and β are parameters. The $\|\log \mathbf{s}\|^2$ term of this regularizer encourages \mathbf{B} to be a rotation with unit singular values. This regularization on the singular values is not typically used with thin plate splines, so we had to modify the standard fitting procedure to account for this. Our technical report [11] describes the details of this modification.

Step 2: Apply transformation \mathbf{f} to the demonstrated gripper trajectory. The gripper poses along the demonstration trajectory are specified by positions $\mathbf{x}_1, \dots, \mathbf{x}_T$ and orientations $\mathbf{R}_1, \dots, \mathbf{R}_T$. We transform the positions and orientations as follows, to adapt the trajectory to the test situation:

$$\mathbf{x}_t \rightarrow \mathbf{f}(\mathbf{x}_t) \quad (4)$$

$$\mathbf{R}_t \rightarrow \text{orth}(\mathbf{J}_f(\mathbf{x}_t)\mathbf{R}_t). \quad (5)$$

Here, \mathbf{J}_f is the 3×3 Jacobian matrix

$$\mathbf{J}_f = \begin{pmatrix} \partial f_x / \partial x & \partial f_x / \partial y & \partial f_x / \partial z \\ \partial f_y / \partial x & \partial f_y / \partial y & \partial f_y / \partial z \\ \partial f_z / \partial x & \partial f_z / \partial y & \partial f_z / \partial z \end{pmatrix}, \quad (6)$$

and $\text{orth}(\cdot)$ is a function that orthonormalizes a 3×3 matrix (e.g. using the SVD).

Equation (4) says that we apply the warping function \mathbf{f} to all of the positions. To transform rotations, we first observe that a reasonable way to transform a vector \mathbf{v} at a point \mathbf{p} through a function \mathbf{f} is to multiply it by $\mathbf{J}_f(\mathbf{p})$, the Jacobian. Equation (5) applies this transformation to the x , y , and z axes of the gripper (which are the columns of matrix \mathbf{R}_t), and then orthogonalizes the resulting basis so it corresponds to a gripper pose (i.e., it is a rotation matrix).

Step 3: Convert the end-effector trajectory into a joint trajectory. To enable the robot to follow the trajectory as closely as possible while satisfying constraints, we formulate the following optimization problem on the joint trajectory

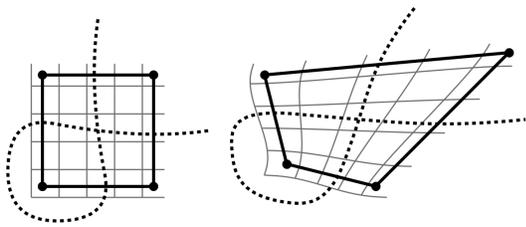


Fig. 3. Illustration of trajectory warping procedure on a cartoon 2-D example. Left: training situation. Right: testing situation.

$\theta_{1:T}$:

$$\underset{\theta_1, \dots, \theta_T}{\text{minimize}} \left[\sum_{t=1}^{T-1} \|\theta_{t+1} - \theta_t\|^2 + \mu \sum_{t=1}^T \|\text{err}(\tilde{\mathbf{T}}_t^{-1} \cdot \text{fk}(\theta_t))\|_{\ell_1} \right]$$

subject to

No collisions, with safety margin d_{safe}

$$\theta_{\min} \leq \theta_{1:T} \leq \theta_{\max} \quad (\text{Joint limits})$$

(7)

Here, $\tilde{\mathbf{T}}_t$ is the desired end-effector pose at time t , $\text{fk}(\cdot)$ indicates the robot's forward kinematics function applied to θ_t , and μ is a scalar parameter. $\text{err}(\cdot)$ is an error function that maps a pose in $SE(3)$ to an error vector in \mathbb{R}^6 . In particular, after decomposing a pose \mathbf{T} into translation \mathbf{p} and quaternion rotation \mathbf{q} , the error vector is simply given by $(p_x, p_y, p_z, q_x, q_y, q_z)$, i.e., the translation and the rotation part of the quaternion.

We will illustrate steps 1 and 2 of the above procedure with a two-dimensional toy example, where the task is to draw a two-dimensional curve through four guide-points. Note that this example merely illustrates the transformation of end-effector positions, not orientations. The left image of Figure 3 shows the training situation, environment shown in solid lines, gripper tip trajectory shown as a dotted line, coordinate grid lines shown as thin solid lines. The right image shows the test situation for which we want to predict a good gripper trajectory. The registered points are the four corners. First, we use the method of thin plate splines [15] to find a function that maps the four corners of the square in the training situation to the four vertices of the new quadrilateral. Then we apply the found warping function to the demonstrated path to obtain a new path (dotted line), which has the same topological characteristics. The warped coordinate-grid lines are shown.

IV. SUTURING EXPERIMENTS WITH RAVEN II IN SIMULATION

We performed a set of experiments in a physics simulation environment, using the Raven II [5]. Compared to the PR2 experiments of the next section, these simulation experiments are interesting because (1) being in simulation, they are more readily reproducible and a larger number of experiments can be run, (2) being a surgical robot, the Raven II has different kinematics than the PR2: non-redundant arm kinematics and a remote center of motion.

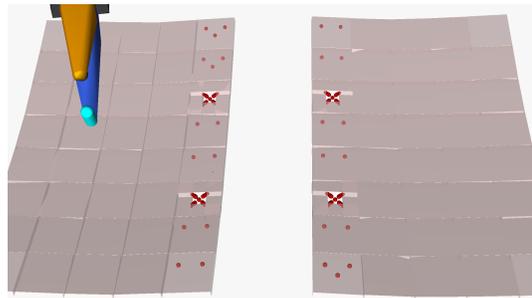


Fig. 4. Points (red) on the surface of the suturing pad used for registration.

The task we study in this paper is to perform a suture stitch (with puncture and knot), and we assume that the robot has a mechanism to pass the needle between its gripper tips such as in the Endo StitchTM system.

We used OpenRAVE [3] to load the model of the robot and do kinematics calculations, and we used a custom simulation environment based on the Bullet Physics Library [2].

A. Training

The simulated Raven II is tele-operated using a pair of Phantom OMNITM.¹

We studied trajectory transfer from a *single demonstration*. The demonstration consists of a sequence of segments (i.e., subtasks, each taking several seconds), which are illustrated in Figure 5. The segmentation is specified at the time of demonstration by pressing a foot-pedal to indicate the start of a new segment. The segmentation is crucial because at test time, trajectory transfer is performed several times—once per segment. At the beginning of each new segment the current scene is registered with the beginning of the corresponding demonstration segment, and then the trajectory from that demonstration segment is transferred.

B. Registration

We use full state information from the simulation to obtain a point cloud for registration. The demonstrator specifies which set of points matter for registering each segment. If the segment involves manipulation of the rope, for example when tying the knot, only points along the rope are used. If the trajectory segment involves interacting with the suturing pad, as when puncturing, points on the surface of the suturing pad are used. Figure 4 visualizes the points on the suturing pad used for registration.

To find the correspondences and the registration function f we use the TPS-RPM algorithm [1], as described in [12].²

¹This was a challenging task, and it took several hours of practice to obtain a good demonstration.

²While full state information gives us access to correspondences, we found that using these correspondences in the naive way, e.g., putting the suture thread into correspondence based on arc-length, did not necessarily result in the “right” registration. Using arc-length for correspondences does not take into account where the suture crosses a flap or itself. On the other hand, TPS-RPM will correctly match together points whose local neighborhoods are similar.

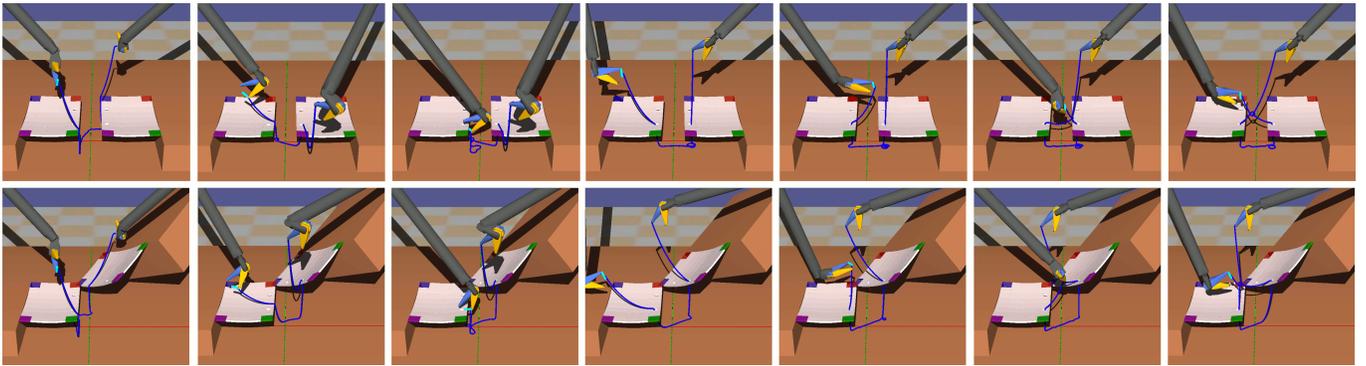


Fig. 5. Snapshots of the scene at the start of each segment (left to right). The final scene shows a knot. Top row shows the segments from the human demonstration. Bottom row shows the transferred trajectory when the right suturing pad was perturbed by $\Delta\mathbf{T} = (-2, 0, 0)\text{cm}$, $\Delta\mathbf{R} = (0^\circ, 40^\circ, 40^\circ)$.

C. Experiments

To test trajectory generalization in the presence of non-rigid perturbations, the left flap of the suturing pad was kept unchanged but the right flap was perturbed by translations and rotations. We scaled the perturbations in the different translation and rotation directions so that the individual perturbations are about equally challenging. Specifically, we chose 0.25cm for x translation, 0.5cm for y, z translation, and 5° for rotation around x, y, z . See Figure 6 for a visualization of the perturbations applied.

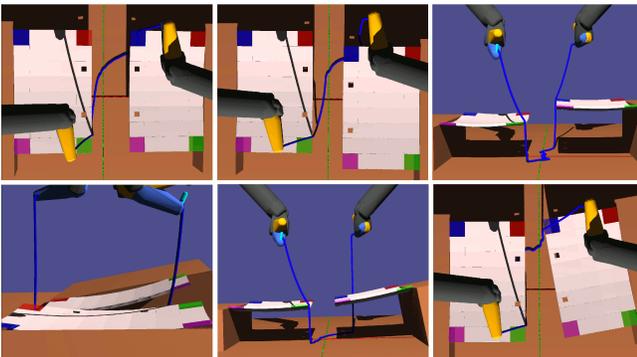


Fig. 6. Perturbations applied to each of the six degrees of freedom of the right suturing pad. Top row shows translation perturbations of -0.5cm along x and 1cm along y and z , in sequence from left to right. Similarly, the bottom row shows rotation perturbations of 10° around each axis.

We considered all the $2^6 = 64$ possibilities for each unit perturbation to be active or not. Then we considered re-scaling each of these 64 perturbations by a factor of $1, 2, \dots, 10$, giving 640 trials in total. A test run was considered a success if the suture was correctly routed through each flap’s piercing and a knot was tied at the end.

Figure 7 reports the success rate as a function of distance from the single demonstration. The graph on the left shows the success rate as a function of the scaling factor. The graph on the right shows the success rate as a function of *perturbation distance* of the applied perturbation vector, which is defined as the number of unit perturbations that need to be composed to obtain the test perturbation. For

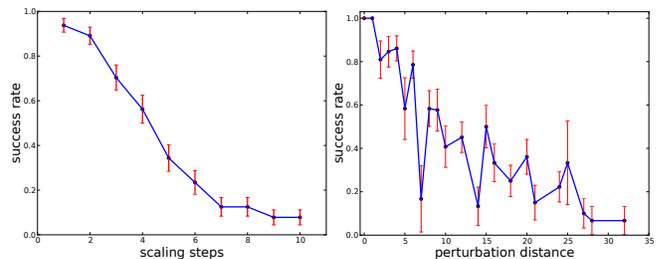


Fig. 7. Performance on Raven II suturing experiments. Left: *scaling steps* refer to the scale i used when generating the perturbation vector (see text for details). For each scaling step, a set of 64 tests were done corresponding to all the 2^6 perturbations of that scale. Right: Success rate versus Perturbation distance. In both cases the error-bars indicate \pm one standard deviation calculated as $\sqrt{\frac{r(1-r)}{N}}$, where r is the success ratio and N is the number of runs from which r was computed.

example, a perturbation of $(0.5, 0, 1.5, 5^\circ, 10^\circ, 0)$ would have a perturbation distance of $2+0+3+1+2+0 = 8$. The results show that the task succeeds reliably for small perturbations but fails as the size of the perturbations is increased.

The failures occurred when grasping the suture thread and passing the thread through the holes; these parts of the task require high precision in position and orientation. The most common underlying cause of these failures was that the registration gave an excessively distorted transformation \mathbf{f} due to the configuration of irrelevant portions of the scene—in particular, the suture thread—causing a bad gripper trajectory. In future work, we will try to address this problem by modifying the registration procedure to emphasize the parts of the scene that are relevant to the current motion. A second most common cause of failure was that the suture moved while executing a segment in open-loop. This problem could potentially be solved by frequent re-planning.

Figure 8 illustrates the dependence of task success on warping cost (as defined in Equation (1)) and the maximum position and orientation error of the grippers. The maximum position error is computed as the maximum over all segments and timesteps of the distance from the gripper to its target position in transferred trajectory. (Recall that the transferred

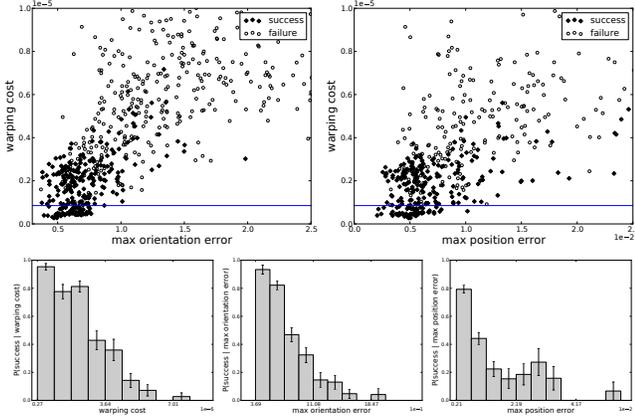


Fig. 8. The dependence of success rate on warping cost and gripper pose error. In the top row, all test runs below the horizontal blue lines were successful. Bottom row shows the distribution of success rate as a function of (a) warping cost (b) gripper orientation error (c) gripper position error (left to right). These plots are based on the results of 640 test runs on a Raven II surgical robot in simulation. The error-bars indicate \pm one standard deviation calculated as in Figure 7.

end-effector trajectory may be infeasible, as discussed in Section III.) Orientation error is defined similarly.

Figure 5 illustrates our algorithm in action. The top row shows a snapshot at the start of each segment during the demonstration. In the first segment of the task, the right arm moves to the front and the left arm punctures the right flap of the tissue. In the second and the third segment, the second flap is punctured. In the fourth and the fifth segment, the suture loops around once. In the last segment, the suture loops around a second time, thereby completing a knot.

Note that the task is broken into segments at the points in time when a high-precision interaction between the robot and its environment needs to occur. For example, segment three starts before the second puncture and segment five starts before the first loop. A video of the trial from Figure 5 and other illustrative videos are available at the URL provided in the Introduction.

V. SUTURING EXPERIMENTS WITH PR2

The experimental procedure on the PR2 is similar to the one used for the experiments in simulation. The key difference is that we do not have access to full state information. We provide additional information by having a human in the loop, who annotates the scenes with the relevant information for registration.

A. Overview

The sequence of segments for tying one suture stitch are as follows:

- Pick up the needle and move into view.
- Pick up one piece of tissue.
- Pierce held piece of tissue.
- Release held piece of tissue.
- Hold down second piece of tissue and pierce.
- Pull the needle and thread through both pieces of tissue.
- Tie a surgeon’s instrument-tied knot.

We did not include the knot tie in the quantitative experiment.

The segment boundaries were chosen qualitatively, where the main consideration was that the subsequent movement could be executed in open loop based on the 3D data obtained at the beginning of the segment.

We obtained 3D data through three different methods — AR markers, ROS interactive markers and a simple point and click interface.

The experimental setup at the start of each segment, along with the relevant annotations, is shown in Figure 9.

B. Training and Keypoint Annotation

At training time, we record a single high-quality demonstration of the complete task. This recording includes the joint states of the robot and an RGBD video of the task execution taken from a fixed camera.³

After data collection, we break up the procedure into separate segments as listed in the previous section. We then manually annotate each individual segment. These annotations indicate the parts of the scene that are most relevant to the following segment of the task.

C. Incorporating Normals and Orientations into Registration

We incorporate surface normals and orientation constraints into the registration by adding extra points. For a normal, two points are used: the first point is on the surface (from which the normal originates) and the second point is $\epsilon = 1\text{cm}$ away from the surface along the normal direction. Similarly, for orientations we use four points: a first point at the origin, and three points at a distance of 1cm from the origin in the direction of local x, y, z . These additional points are illustrated in Figure 10. Figure 9 shows several snapshots of this procedure from PR2’s perspective during execution and illustrates the points, normals and orientations used for each segment.

D. End-Effector Trajectory Generation

We apply the method described in III, but without the need to infer correspondences (which are provided by the human-in-the-loop), to fit a 3D non-rigid transformation and use it to warp the trajectory of the end-effectors. Note that some trajectory segments move only one arm, while others move both arms simultaneously.

The end-effector is either the robot’s gripper or the needle tip. In the annotation stage, we indicate what the end-effector is for each segment. It is important to note that whenever the end-effector is the needle tip, we need to know the precise pose of the needle relative to the robot’s gripper. To this end, all segments in which the robot grabs the needle are followed by “look-at-needle” segments, in which the robot holds up the needle, allowing to acquire the pose of the tip.

³In the PR2 experiments, we used an ASUS Xtion-Pro mounted to the robot’s head as our camera.

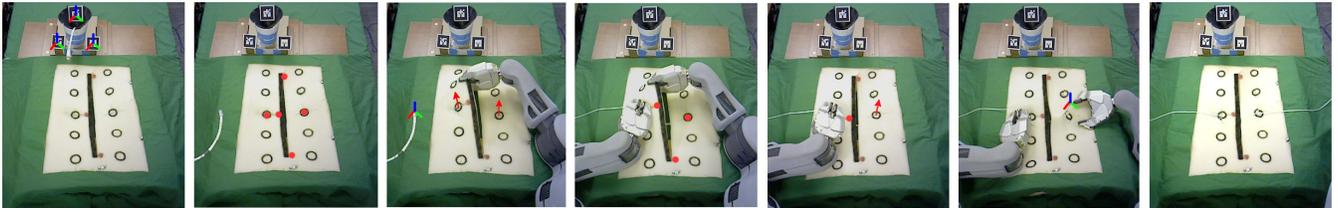


Fig. 9. Snapshots of the PR2 suturing setup at the start of each segment (left to right). The setup has been rotated by -10° around y with respect to the demonstration setup. The relevant annotations in each segment are shown: (1) Orientations of AR markers on the needle stand (2) Points at the top, middle and bottom of the cut and at the two puncture points (3) Orientation of the needle tip and normals at the two puncture points (4) Points on the middle and bottom of the cut and at the right puncture point (5) Point on the middle of the cut and the normal at the right puncture point (6) Orientation of the needle tip. The last image shows the setup after finishing the experiment. All the points and normals are colored in red. Normal directions are marked with arrows. Orientations are shown by red, green and blue lines corresponding to local x, y, z respectively.

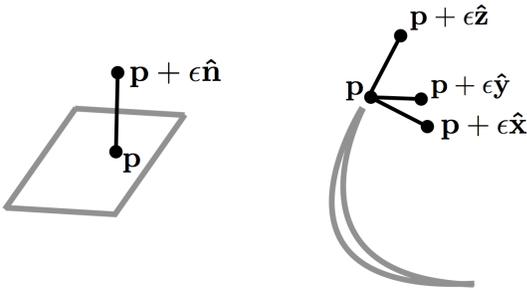


Fig. 10. Extra points are added when normal and orientation information at the keypoints are relevant. Left: extra point added for a surface normal. Right: three extra points added for orientation (at needle tip, needle shown in gray).

E. Experimental Results

Our experiments on the PR2 consider the task of grasping the suturing needle, piercing through both sides of tissue along the cut and pulling the needle through the tissue. This procedure is illustrated in Figure 9.

The suturing needle is a bent, rectangular shaft with 5mm side length and has a 3.2mm diameter rope attached to one end. It sutures through a foam pad with pre-cut holes. The needle rests in a stand which forces it to stay in a particular orientation.

We recorded one demonstration with manual annotations as described above. To investigate the extent to which the learned trajectory generalizes to different configurations of the suturing scene, we applied rigid and non-rigid perturbations to the suturing setup and measured the success rate. We experimented on $\pm 10^\circ$ and $\pm 15^\circ$ rotations along each of the x, y and z axes. We also applied non-rigid deformations to our setup, including bending along the x and y axes and a diagonal pair of holes. Figure 12 shows some of the experimental setups we worked with.

We did not consider translations in our quantitative evaluation as success in those experiments merely reflects whether the new setups were within the reachable workspace of the gripper or not.

The results of our experiments are outlined in Figure 11. The overall rate of success was 87%. For rotations of 10° along the three main axes, we found that the robot

Perturbation	Success Rate
10° x rotation	2/2
15° x rotation	2/2
-10° x rotation	2/2
-15° x rotation	1/2
10° y rotation	2/2
15° y rotation	2/2
-10° y rotation	2/2
-15° y rotation	2/2
10° z rotation	2/2
15° z rotation	0/2
-10° z rotation	2/2
-15° z rotation	2/2
Bend x -axis	2/2
Bend y -axis	2/2
Diagonal holes	1/2

Fig. 11. Experimental results for PR2 performing pierce and regrasp subject to different perturbations. Two trials were performed in each experimental setting. In the bending experiments, the middle of the cut was raised by 7.5 cm. For the diagonal holes experiments, the distance between the holes was increased by 2.5 cm. The failure cases were as follows: (1) one $-15^\circ x$ rotation experiment failed during the needle pull-through (2) $15^\circ z$ rotation experiments failed once in the second pierce and once during the needle pull-through (3) one diagonal-holes experiment failed in the first needle pierce.

succeeded 100% of the time in our experiments. However, as the perturbation of the environment was increased, different sources error in the overall pipeline became apparent. For the case of the $15^\circ z$ rotation experiments, the trajectory required was close to the joint limits of the PR2. This caused the PR2 to miss the second pierce in one experiment and the needle pull-through in the second experiment. The diagonal hole experiments were challenging because the testing scene differed substantially from the demonstration; one of the experiments failed due to the needle piercing the first hole at a slightly oblique angle. For the case of the $-15^\circ x$ rotation experiment, the needle got caught in the foam and slipped out of the gripper while being pulled out. Also note that the success rate is not the same on mirrored perturbations: $+15^\circ$ and -15° along x or z do not succeed with the same rate. This is because the task is asymmetric with the left and right hands performing different actions.



Fig. 12. Snapshots of a subset of the different initial conditions we experimented on. The perturbations in the figures are (from the top left): (1) No rotation or non-rigid deformation (2) $+15^\circ$ along x (3) $+15^\circ$ along y (4) -15° along y (5) $+15^\circ$ along z (6) Bend along x raised by 15 cm at center (7) Bend along y raised by 15 cm at center (8) Diagonal holes, as shown by the arrow, with distance of 18 cm between them.

In summary, our experiments showed a good success rate even when the experimental setup was rotated or bent. Failures could be attributed to limitations of our setup that are either relatively easy to fix or inherent to the scale and material properties of the experimental setup.

Videos of a few of our experiments are available at the URL provided in the Introduction.

VI. CONCLUSION

We have studied the applicability of the trajectory transfer method introduced in [12] to the problem of autonomous suturing. Our experimental results with the simulated Raven II surgical robot and a real PR2 suggest that trajectory transfer could be a powerful building block to bring us closer to having robots learn to perform challenging manipulation tasks from demonstrations. Our experimental results showed that, as one might expect, success rate is highly correlated with the similarity between the demonstration scene and the test scene. Using multiple demonstrations could potentially allow to better cover the set of possible test scenes. Some important questions for future work include how to optimally leverage multiple demonstrations in the context of trajectory transfer, how many demonstrations are needed, and how to systematically and efficiently obtain a representative set of training data.

VII. ACKNOWLEDGEMENTS

We thank Jonathan Ho and Alex Lee for their contributions to the software we used. We thank Sachin Patil, Zoe McCarthy, and Preetum Nakkiran for insightful comments and advice. This research has been funded in part by the Intel Science and Technology Center on Embedded Computing, by Air Force Office of Scientific Research Young Investigator Program (AFOSR-YIP) award # FA9550-12-1-0345, by NSF under award # IIS-1227536, and by a Sloan Foundation Fellowship.

REFERENCES

- [1] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding*, vol. 89, no. 2, pp. 114–141, 2003.
- [2] E. Coumans, "Bullet physics library," 2012, www.bulletphysics.org.
- [3] R. Diankov and J. Kuffner, "OpenRAVE: A planning architecture for autonomous robotics," *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-08-34*, 2008.
- [4] J. Duchon, "Splines minimizing rotation-invariant semi-norms in sobolev spaces," *Constructive theory of functions of several variables*, pp. 85–100, 1977.
- [5] B. Hannaford, J. Rosen, D. Friedman, H. King, P. Roan, L. Cheng, D. Glozman, J. Ma, S. Nia Kosari, and L. White, "Raven-II: an open platform for surgical robotics research," *Transactions on Biomedical Engineering*, vol. 10, no. 10, 2012.
- [6] H. Kang and J. T. Wen, "Autonomous suturing using minimally invasive surgical robots," in *Control Applications, 2000. Proceedings of the 2000 IEEE International Conference on*. IEEE, 2000, pp. 742–747.
- [7] —, "Robotic assistants aid surgeons during minimally invasive procedures," *Engineering in Medicine and Biology Magazine, IEEE*, vol. 20, no. 1, pp. 94–104, 2001.
- [8] H. Mayer, I. Nagy, A. Knoll, E. Braun, R. Lange, and R. Bauernschmitt, "Adaptive control for human-robot skill transfer: Trajectory planning based on fluid dynamics," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 1800–1807.
- [9] H. Mayer, F. Gomez, D. Wierstra, I. Nagy, A. Knoll, and J. Schmidhuber, "A system for robotic heart surgery that learns to tie knots using recurrent neural networks," *Advanced Robotics*, vol. 22, no. 13-14, pp. 1521–1537, 2008.
- [10] N. Padoy and G. D. Hager, "Human-machine collaborative surgery using learned models," in *In the proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011.
- [11] J. Schulman, "Notes on iterative registration with thin plate splines," July 2013. [Online]. Available: <http://eecs.berkeley.edu/~joschu/docs/registration-techreport.pdf>
- [12] J. Schulman, J. Ho, C. Lee, and P. Abbeel, "Generalization in Robotic Manipulation Through The Use of Non-Rigid Registration," *Submitted. Manuscript at http://rll.berkeley.edu/isrr2013/fd/*, 2013.
- [13] C. Staub, T. Osa, A. Knoll, and R. Bauernschmitt, "Automation of tissue piercing using circular needles and vision guidance for computer aided laparoscopic surgery," in *In the proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2010.
- [14] J. van den Berg, S. Miller, D. Duckworth, H. Hu, A. Wan, X.-Y. Fu, K. Goldberg, and P. Abbeel, "Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2010.
- [15] G. Wahba, *Spline models for observational data*. Siam, 1990, no. 59.