Motion Generation of Robotic Surgical Tasks: Learning From Expert Demonstrations

Carol E. Reiley, Erion Plaku, and Gregory D. Hager
Department of Computer Science, Johns Hopkins University
3400 N. Charles Street, Baltimore, MD 21218
{creiley, erion, hager}@jhu.edu

***Abstract*— Robotic surgical assistants offer the possibility of automating portions of a task that are time consuming and tedious in order to reduce the cognitive workload of a surgeon. This paper proposes using programming by demonstration to build generative models and generate smooth trajectories that capture the underlying structure of the motion data recorded from expert demonstrations. Specifically, motion data from Intuitive Surgical's da Vinci Surgical System of a panel of expert surgeons performing three surgical tasks are recorded. The trials are decomposed into subtasks or surgemes, which are then temporally aligned through dynamic time warping. Next, a Gaussian Mixture Model (GMM) encodes the experts' underlying motion structure. Gaussian Mixture Regression (GMR) is then used to extract a smooth reference trajectory to reproduce a trajectory of the task. The approach is evaluated through an automated skill assessment measurement. Results suggest that this paper presents a means to (i) extract important features of the task, (ii) create a metric to evaluate robot imitative performance (iii) generate smoother trajectories for reproduction of three common medical tasks.**

I. Introduction

Recent advances in surgical techniques have provided another avenue for research into machine learning of human skills. More clinical operations are being performed via teleoperation using surgical robots such as the da Vinci surgical robot (Intuitive Surgical, Sunnyvale, CA) [1]. A 2007 study stated that over 50,000 robotic-assisted prostatectomies have already been performed worldwide [2]. Robotic-assisted surgeries, however, present significant manipulation challenges which make it more demanding than open surgery. As an illustration, even a laparoscopic knot tie can take more than three minutes to complete compared to one second for a manually tied knot [3]. The future possibility of using a robot to automate portions of such tasks can assist the surgeon by reducing the cognitive workload and shortening the surgical procedure time. The work in [3] presented promising results from a system that automated the wind up portion of a knot tying task using recurrent neural networks trained from expert surgeons. The work in [4] used imaging techniques to plan motions which the robot can perform autonomously. Other approaches rely on hardwiring controllers for specialized stitching machines [5] and laparoscopic instruments [3] or optimizing controllers [6], [7].

A crucial aspect of further enhancing automation in robotic-assisted surgery relates to understanding and modeling expert skill to enable surgical robots to perform at similar levels as expert surgeons [8], [9], [10], [11], [12]. Toward this goal, this paper develops computational methods to automatically model expert skill for surgical tasks. The proposed approach leverages the idea of using machine learning to build generative models and generate from these models smooth trajectories that capture the underlying structure of the motion data recorded from expert demonstrations. Fig. 1 shows a flowchart of the proposed methodology. Our approach begins with recording raw motion data gathered from multiple surgeons of different expertise levels that perform surgical tasks on the da Vinci robotic surgical system, as shown in Fig. 2. Multiple demonstrations of the same task are needed to create a representative generalization of the skill. We extract inherent structure behind surgical motion by analyzing modular and flexible blocks of motions or sequence of states called "surgemes" (i.e., pick up needle, push needle through tissue) [8]. Fig. 3 shows an example of several surgemes and the transition flow between them. In this way, the trials from surgical tasks are decomposed into

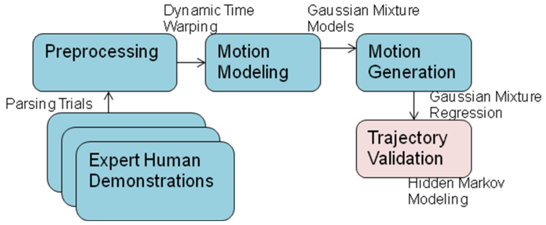

Fig. 1. Flowchart of the proposed method for motion modeling and trajectory generation.

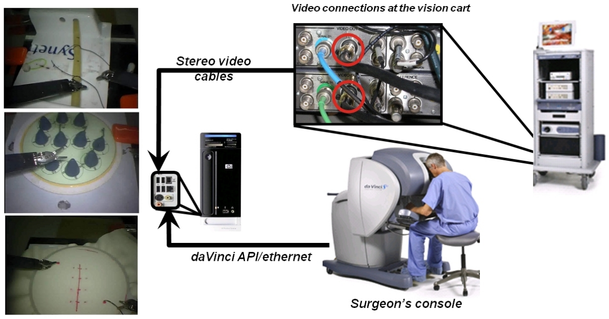

Fig. 2. Da Vinci surgical station and data recording process of API data and synchronized stero video. Left: A snapshot of the three surgical tasks.

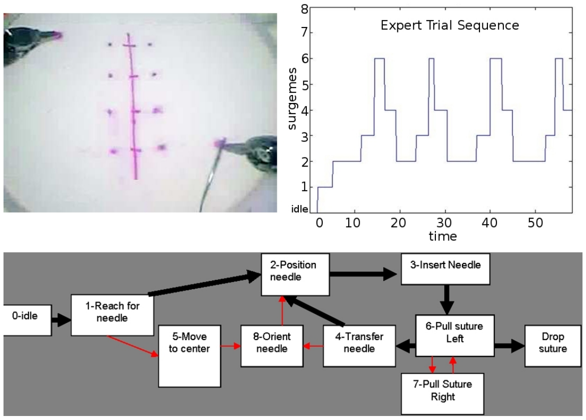

Fig. 3. Top right shows the transitions between surgemes through an example trial. Surgemes 2, 3, 6, 4 constitute a throw, which is repeated four times for each trial. Bottom shows this pictorally. The black thick lines indicate ideal trials. Red thin lines indicate trials with error (dropped needle) and extra transitions needed to compensate.

surgemes and are then temporally aligned through dynamic time warping. When surgemes are combined together, they can form very complex movements. Next, we encode the motions through a Gaussian Mixture Model (GMM) to capture the experts' underlying motion structure. Finally we are able to generate trajectories to perform gestures similar to the training examples using Gaussian Mixture Regression (GMR) which constrains smoothness. The approach is validated through classifying the skill of the generated motion through an automated skill-assessment procedure based on Hidden Markov models. Results suggest that this paper presents a means to (i) extract important features of the task, (ii) create a metric to evaluate robot imitative performance (iii) generate smoother trajectories for reproduction of common medical tasks, such as those illustrated in Fig. 2.

II. Data Acquisition

To facilitate presentation, before relating method details, we first describe the data acquisition process. The experimental setup consists of expert surgeons using the da Vinci surgical system to perform surgical tasks on the teleoperated system. The da Vinci surgical system is an FDA-approved seven degree of freedom (dof) robot that consists of three separate components: master-side console, vision cart, and patient-side manipulators. The surgeon manipulates the serial link inputs which relays the position, orientation, and grip commands sent from the surgeon to the robotic manipulators on the patient side. Motion data is collected through the da Vinci Application Programming Interface (API) using an Ethernet cable synchronized with high resolution stereoscopic video (1024x768) at 30 Hz. Each datapoint v_i consists of a $d = 83$ variable subset of the total 192 variables which include positions, rotation angles, and velocities of the master/patient side manipulators. We investigated three common surgical tasks: suturing, knot tying, and needle passing. These tasks are used in training to test psychomotor skill and are fundamental skills across all types of surgeries. Data was collected from 11 surgeons. For training the proposed method, we used 5 trials from expert surgeons for each task. To evaluate the proposed method, we used 5 new trials of each expertise (expert, intermediate, novice) for each task.

III. Methods

A. Multidimensional Dynamic Time Warping

Our approach begins with extracting surgemes from the raw motion data from expert trials [8]. Each expert trial is manually segmented into a sequence of 16 surgemes, as illustrated in Fig. 3. Dynamic Time Warping (DTW) [13] is then used to temporally align the surgemes. More specifically, for $h \in \{1, \ldots, 16\}$, let $s_h^1, \ldots, s_h^N$ denote the h-th surgeme across the different N trials of the same task. Then, each s_h^i is aligned to s_h^{med}, where s_h^{med} denote the h-th surgeme with the median time among the N trials. DTW provides a time-invariant method to synchronize two time series that vary in time and speed by finding a non-linear warping path. DTW sets each entry (i, j) of a matrix D to the distance between datapoints v_i, v_j, i.e., $D[i, j] = \sqrt{\sum_{\ell=1}^{d} (v_{i,\ell} - v_{j,\ell})^2}$, where d is the number of variables. The matrix D is then traversed through a dynamic programming approach to find the best synchronization by minimizing the sum of distances between all points that are warped onto each other. This preprocessing step is a way to extract task constraints irrespective of difficulties due to strong distortion in time to trajectories which softens key features.

B. Motion Modeling by Gaussian Mixture Model

The datapoints from the temporally aligned surgemes $\{s_h^1, \ldots, s_h^N\}$ are then modeled through a Gaussian Mixture Model (GMM). In learning from examples, a GMM produces a joint density estimate over the input and output space based on the training set. A GMM of K Gaussians is defined by the probability density function

$$p(v_j) = \sum_{k=1}^{K} p(k) p(v_j|k),$$

where v_j is the datapoint, $p(k)$ is the prior, and $p(v_j|k)$ is the conditional probability density function. Time is encoded directly into the GMM, (i.e., $v_j = (v_{jt}, v_{js})$, where v_{jt} and v_{js} correspond to time and spatial information, respectively) so that a smooth signal can be retrieved through regression, as described in the next section. Moreover,

$$p(v_j|k) = \mathcal{N}(v_j; \mu_k, \Sigma_k)$$

where $(p(k), \mu_k, \Sigma_k)$ are parameters of the Normal Gaussian distribution components k defined as the priors, means, and covariances, respectively. Maximum Likelihood Estimation of the mixture parameters is performed iteratively using the EM algorithm [14]. EM is a local search algorithm that guarantees finding a locally optimal fit of Gaussians to the data through increasing the likelihood of the training set during optimization. This algorithm requires an initial estimation, which is done through k-means clustering. We use the programming by demonstration toolkit [6] for the implementation of GMMs.

C. Motion Generation by Gaussian Mixture Regression

In order to generalize a path of the signals for a task we apply Gaussian Mixture Regression [15]. We can efficiently compute the output by conditioning the joint distribution on the input and take the expected value. Consecutive time series samples are queried for the corresponding spatial information estimated through regression. The temporal and spatial components are separated for the mean and covariance matrix of the Gaussian component k as

$$\mu_k = \{\mu_{t,k}, \mu_{s,k}\}, \quad \Sigma_k = \begin{pmatrix} \Sigma_{t,k} & \Sigma_{ts,k} \\ \Sigma_{st,k} & \Sigma_{s,k} \end{pmatrix}.$$

For each Gaussian k, the conditional expectation of $\varepsilon_{s,k}$ given time and the expected covariance is calculated as

$$\hat{\varepsilon}_{s,k} = \mu_{s,k} + \Sigma_{st,k} \left(\Sigma_{t,k}\right)^{-1} \left(\varepsilon_t - \mu_{t,k}\right)$$
$$\hat{\Sigma}_{s,k} = \Sigma_{s,k} + \Sigma_{st,k} \left(\Sigma_{t,k}\right)^{-1} \Sigma_{ts,k},$$

where $\hat{\varepsilon}_{s,k}$ and $\hat{\Sigma}_{s,k}$ are combined based on the probability that the Gaussian component k has

$$\beta_k = p(\varepsilon_t|k) / \sum_{i=1}^{K} p(\varepsilon_t|i).$$

Therefore, the conditional expectation and covariance of ε_t given time is as follows:

$$\hat{\varepsilon}_s = \sum_{k=1}^{K} \beta_k \hat{\varepsilon}_{s,k}, \quad \hat{\Sigma}_s = \sum_{k=1}^{K} \beta_k^2 \hat{\Sigma}_{s,k}.$$

Thus, if we evaluate $\{\hat{\varepsilon}_s, \hat{\Sigma}_s\}$ at different time steps ε_t, we can produce a generalized form of the motions to create output sequence O. With a probabilistic model, only the means and covariance matrices of each Gaussian are stored. We use the programming by demonstration toolkit [6] for the implementation of GMR.

D. Motion Evaluation by Hidden Markov Models

Hidden Markov Models (HMM) are used to evaluate the trajectories generated by the Gaussian Mixture Regression. An HMM is a doubly stochastic process [16]. It is modeled as a series of states that obey the Markov property, where these underlying states are not directly observable but can only be inferred through the observation of another set of stochastic processes. We formulate the learning problem as multiple discrete HMMs and develop a testbed based on the "most likely performance" criteria to have generated a sequence of actions based on [17]. This method was used to validate if the generated sequence could be correctly classified according to the appropriate skill level, i.e., expert, intermediate, novice. The signals are discretized using a sliding window fast fourier transform. From the discretized signals, an HMM is trained for each skill level $\lambda_s \in \{\lambda_{\text{expert}}, \lambda_{\text{intermediate}}, \lambda_{\text{novice}}\}$ using the Baum-Welch algorithm with no prior training labels as implemented in the Matlab statistics toolbox. Generalized Expectation Maximization algorithm was used to uncover the best structure of the hidden states. Given the trained HMMs, let O denote a trajectory generated by the Gaussian Mixture Regression, as described in section III-C. The trajectory O is discretized and an HMM is build from the discretized signal as described above in the training procedure. The trajectory O is then classified as an expert, intermediate, or novice based on the maximum log likelihood, i.e.,

$$\text{argmax}_{\lambda_s \in \{\lambda_{\text{expert}}, \lambda_{\text{intermediate}}, \lambda_{\text{novice}}\}} \log P(O|\lambda_s).$$

IV. Results

Our goal is to generate a smooth trajectory of a trial; therefore, the end of one surgeme must connect to the next seamlessly. To do so, we concatenated the Gaussian parameters μ and Σ of consecutive surgemes to generate new parameters μ_{new} and Σ_{new} between each transition. We averaged all the priors generated from EM of each surgeme. GMR was was sampled once using the parameters of each GMM for each of the 16 surgemes. The time was used to query the corresponding spatial information by computing the expected distribution. By providing temporal values as inputs, it thus outputs a smooth generalized version of the data encoded in GMM, and associated constraints expressed by covariance matrices. Fig. 4 shows 2 surgemes connecting together.

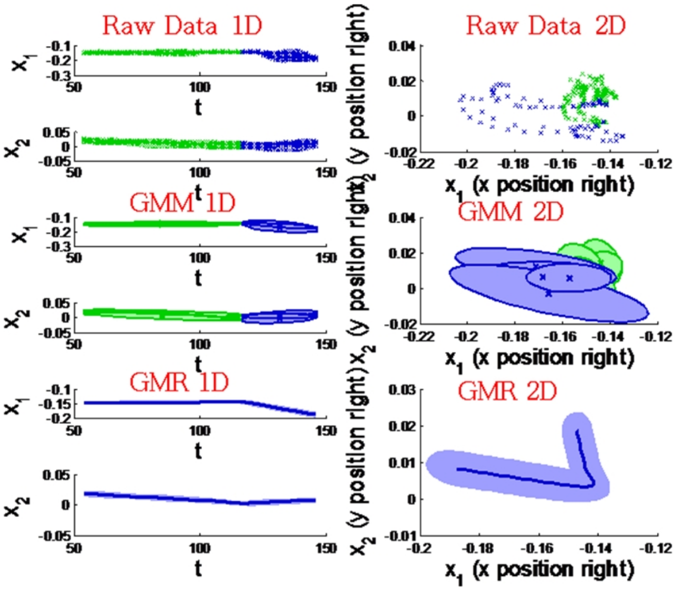

Fig. 4. Demonstrating the raw data of two surgemes (top in blue and in green), the GMM (center) modeled with 4 states, then a generated path using GMR concatenated together (bottom in dark blue and the standard deviation in light blue). The left hand side figures are 1D and right side figures are 2D corresponding to the position of the tooltip. The axis x1 and x2 are the x and y positions of the right tool respectively.

To test the accuracy of the GMM/GMR generative models, we ran a classification algorithm [17] that would classify trajectories generated by the GMM/GMR trained on expert data as either coming from expert, intermediate, or novice surgeons. The classification algorithm trains three HMMs (expert, intermediate and novice) from five new unseen trials for each skill level. Each trajectory O generated by the expert GMM/GMR is classified based on the log likelihood estimate to the nearest HMM, as described in section III-D. Table I shows that the trajectories generated by the GMM/GMR generative model in all three tasks are closest to the expert model. This indicates that the GMM/GMR generative model trained on expert data generates smooth trajectories that are classified as coming from experts.

TABLE I

LOG LIKELIHOOD DISTANCE OF TRAJECTORIES GENERATED BY GMM/GMR TRAINED ON EXPERT DATA TO HMMS TRAINED ON NEW EXPERT, INTERMEDIATE, AND NOVICE TRIALS.

	Suturing	Knot tying	Needle transfer
DTW Expert	−816.5	−756.2	−804.1
DTW Intermediate	−840.4	−822.1	−887.4
DTW Novice	−914.7	−897.0	−911.3
No DTW Expert	−813.9	−1311.9	−1301.2
No DTW Intermediate	−869.8	−1301.1	−1287.0
No DTW Novice	−937.1	−1438.8	−1118.1

We also tested the impact of DTW on the GMM/GMR process. Fig. 5 provides a qualitative illustration. Moreover, Table I indicates that when DTW is not used, then the trajectories generated by GMM/GMR are not classified as coming from experts. These results indicate that DTW considerably improves the GMM/GMR process to extract key features of a surgeme and generate smooth trajectories that are classified as coming from experts.

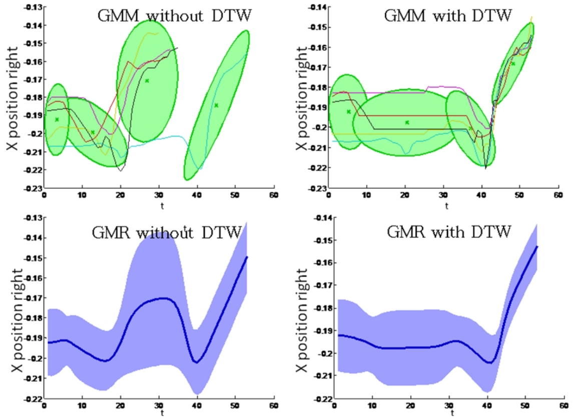

Fig. 5. An estimation of a single GMM parameter (top) and the GMR retrieval process (bottom) with raw data (left) of 3 representative suturing surgeme examples before (left) and after (right) multidimensional dynamic time warping.

V. DISCUSSION

This paper explores the problem of automating portions of a human-machine cooperative task and presents a method based on GMM/GMR to generate smooth trajectories that capture the underlying structure of the motion data recorded from expert demonstrations. Promising initial validation are obtained which show that the generated trajectories are classified as coming from expert surgeons. This is a distinct advantage over HMMs, since trajectories generated from HMMs (even those trained on expert data) are not smooth and are usually classified as coming from novice surgeons. Results in this paper also indicate that the GMM/GMR method is able to extract important features of the task, determine quantitative metrics to evaluate robot imitative performance, and reproduce several medical tasks. Starting with these initial results, we plan to have a teleoperated system perform simple surgical tasks. With the surgemes, it should be possible to reuse the skilled motion primitives to describe a large variety of tasks. The current technique is open loop as a method aimed at providing a solution for reproducing a generalization of the demonstrations for reproduction. The eventual goal of this research is to provide closed loop realtime assistance to surgeons in a teleoperated and cooperative manipulation environment.

ACKNOWLEDGMENT

This work was supported in part by NSF 0534359 (CER, GDH), NSF IIS-0748338 (EP, GDH), and the Graduate Research Fellowship Program (CER).

REFERENCES

[1] G. S. Guthart and J. K. Salisbury, “The intuitivetm telesurgery system: Overview and application,” *IEEE International Conference on Robotics and Automation*, pp. 618–621, 2000.

[2] *Intuitive Surgical Annual Report*, Intuitive Surgical Inc., 2007.

[3] H. Mayer, F. Gomex, D. Wierstra, I. Nagy, A. Knoll, and J. Schmidhuber, “A system for robotic heart surgery that learns to tie knots using recurrent neural networks,” *ICRA*, pp. 543–548, 2006.

[4] R. Taylor, B. Mittelstadt, H. Paul, W. Hanson, and P. Kazanzides, “An image-directed robotics system for precise orthopaedic surgery,” *IEEE Trans. on Robotics and Automation*, vol. 10(3), p. 261275, 1994.

[5] H. Kang and J. T. Wen, “Robotic knot tying in minimally invasive surgeries,” *IEEE International Conference on Intelligent Robots and Systems*, pp. 1421–1426, 2002.

[6] S. Calinon, F. Guenter, and A. Billard, “On learning, representing and generalizing a task in a humanoid robot,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 37, pp. 286–298, 2007.

[7] 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,” *ICRA*, 2010.

[8] H. Lin, I. Shafran, D. D. Yuh, and G. D. Hager, “Towards automatic skill evaluation: Detection and segmentation of robot-assisted surgical motions,” *Computer Aided Surgery*, vol. 11(5), pp. 220–223, 2006.

[9] C. Reiley, H. Lin, B. Varadarajan, S. Khudanpur, D. D. Yuh, and G. D. Hager, “Automatic recognition of surgical motions using statistical modeling for capturing variability,” *MMVR*, 2008.

[10] T. Kowalewski, J. Rosen, L. Chang, M. Sinanan, and B. Hannaford, “Optimization of a vector quantization codebook for objective evaluation of surgical skill,” *Studies in health technology and Informatics*, vol. 98, pp. 174–179, 2004.

[11] M. Nicolescu and M. Mataric, “Natural methods for robot task learning: Instructive demonstrations, generalization and practice,” *Intl Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 241–248, 2003.

[12] M. Pardowitz, R. Zoellner, S. Knoop, and R. Dillmann, “Incremental learning of tasks from user demonstrations, past experiences and vocal comments,” *IEEE Trans. on Systems, Man and Cybernetics, Part B,*, vol. 37(2), pp. 322–332, 2007.

[13] H. Sakoe and S. Chiba, “Dynamic programming algorithm optimization for spoken word recognition,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 26, pp. 43–49, 1978.

[14] A. Dempster and N. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the Royal Statistical Society*, vol. 39(1), pp. 1–38, 1977.

[15] D. Cohn, Z. Ghahramani, and M. Jordan, “Active learning with statistical models,” *Artificial Intelligence Research*, vol. 4, pp. 129–145, 1996, gMR citation.

[16] L. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” *IEEE*, vol. 77(2), pp. 257–286, 1989.

[17] C. E. Reiley and G. D. Hager, “Task versus subtask surgical skill evaluation of robotic minimally invasive surgery,” *Medical Image Computing and Computer Assisted Intervention*, vol. 5761, pp. 435–442, 2009.