AAAI Technical Report FS-12-07
Robots Learning Interactively from Human Teachers

Learned Partial Automation for Shared Control in Tele-Robotic Manipulation

Sebastian Bodenstedt, Nicolas Padoy, Gregory Hager
Johns Hopkins University
Baltimore, MD, USA

Abstract

When used in challenging applications like surgery or underwater maintenance, the use of tele-operated robots involves manipulations that are complex to perform on the master controllers due to restricted access and limited perception. In this paper, we investigate an assistance approach for tele-robotic manipulation, in which the robot automates several degrees of freedom (DOF) of the tools, such as their orientation. This automation requires the understanding of the intent of the operator, so as to not impede the natural manipulation of the remaining DOF. Our system is therefore based on the observation that in the aforementioned applications, the manipulation tasks have often a structure that can be learned from the daily usage of the robot.

We propose an approach that uses the typical motion performed by the operator during a given task, learned from demonstration, to automate the rotation of the manipulator in new instances of this task. The operator keeps control of the robot by manipulating the tool translation and can recover full control if needed. The learned motion model is based on Gaussian Mixture Regressions and combined with a 3D reconstruction of the environment to address variations in the task. We demonstrate our assistance approach using a da Vinci robot on a task consisting of moving a ring along a wire possessing a complex 3D shape.

Introduction

Tele-operated robots are often used to perform complex manipulations in challenging environments, where human perception is limited. In applications like deep underwater maintenance or minimally invasive surgery, where human access is restricted, the remote control of the robotic tools can be difficult or tedious for the operator, in spite of the relatively well defined task that needs to be performed.

One solution to facilitate the work of the operator is to automate portions of the tele-operated tasks (Nageotte et al. 2009; van den Berg et al. 2010; Padoy and Hager 2011). Complete automation is however very challenging, in particular due to the difficulty arising in modeling the complex decision processes and in perceiving accurately the environment. There is therefore a need for developing new tele-manipulation paradigms that simplify the remote control of the robotic manipulators. In (Padoy and Hager 2011), an approach has been proposed to trade the control back and forth between the robot and the operator, so as to automate the transport motions and to reduce the master manipulator's workspace. In this work, we investigate another method for sharing control between a tele-operated robot and an operator: we aim at automating several degrees of freedom (DOF) of the manipulator using real-time information about the other DOF that are controlled by the operator and information about the 3D environment. Compared to virtual fixtures (Rosenberg 1993), which are used to constrain the motion of the manipulators, for instance using force feedback, the operator is here totally relieved from controlling certain DOF. Compared to full automation (Nageotte et al. 2009; van den Berg et al. 2010), the operator keeps control of certain parameters of the motion and also of the speed with which the task is performed.

The main challenge is to compute values for the automated DOF that are compatible with the operator's intent and that do not make the rest of the manipulation more complex. For example, computing the automated DOF is not straightforward, since certain aspects of tele-operation are difficult to model, such as the ergonomics of the master-manipulators in the hands of the operator.

However, one main advantage of tele-manipulation is that motions are naturally demonstrated by regular operators. It is therefore possible to learn from the large database of motions coming from the daily usage of the robot how to augment the tele-operation with partial automation. Ideally, the learned models will also be operator-specific in order to adapt to the different ways employed to control the remote tools.

Our approach first learns from demonstration the typical manipulation motion required to perform a given task using Gaussian Mixture Regression (GMR). During assistance, it combines this a-priori information with a 3D reconstruction of the new environment in order to automate the rotation of the manipulator. A major difference to previous work in learning from demonstration (Calinon 2009; Argall et al. 2009), often targeted at humanoids (Ye and Alterovitz 2011), is the online use of the translation manipulated by the operator as input of the regression, as opposed to the time variable only.

To demonstrate this approach, we use a non-commercial

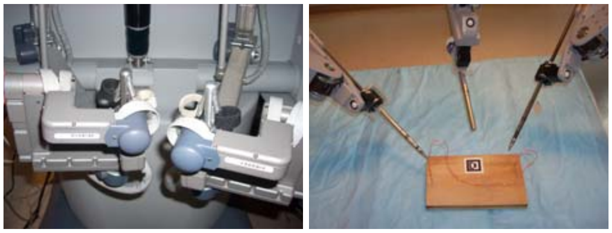


Figure 1: Master manipulators of the da Vinci robot (Left). Three robotic arms, with camera in the center (Right).

version of the da Vinci robot from Intuitive Surgical™ (Guthart and Jr. 2000) used exclusively for research purposes. It consists of four robotic arms that are remotely manipulated using two master manipulators. One of the four robotic arms holds a stereo camera imaging the environment (see fig. 1). The other arms can be used to hold tools with 7 DOF (3 for translation, 3 for rotation and 1 for the gripper opening angle).

This robot is used to perform a task in which a circular ring needs to be transferred from one side of a wire having a tortuous 3D shape to the opposite side. This task requires complex tool motions due to the curvature of the wire and is delicate to perform without colliding with or bending the wire because of the limited depth perception. We propose an assistance system that learns from an operator how to automate the tool orientation for the aforementioned task and show experiments in a situation where the environment is deformed.

Related Work

Tele-operated robots are a natural environment for experimenting with human-machine cooperation. In (Guo et al. 1995), an approach was presented in which a robotic arm performs autonomously a planned path under surveillance of an operator, who can take control of the arm and avoid obstacles if needed. Such a system does not use any machine intelligence model about the performed task. In (Yang, Xu, and Chen 1993), tool motions are learned from demonstration using Hidden Markov Model (HMM) in order to smooth out tremors and potential deviations within the trajectory. This leads to the concept of virtual fixtures (Marayong et al. 2003), which can be used for real-time assistance during remote manipulation, for instance by imposing no-go zones for the tools. Further information about the structure of the task can be used, as in (Kragic and Hager 2003) where a HMM is used to recognize the task context in order to implement different assistance modes during the manipulation.

Gaussian Mixture Regression is a widely used method for trajectory learning. It has been used in combination with Dynamic Time Warping (Sakoe and Chiba 1978) for learning tele-manipulated motions and for comparing skills in (Reiley, Plaku, and Hager 2010). (Calinon 2009; Argall et al. 2009) provide an extensive review of learning from demonstration techniques. A major difficulty with such approaches is to take into account variations in the environment and also to model the interactions between the robot and the environment. (Ye and Alterovitz 2011) presents an interesting approach for detecting such interactions and thereby the essential portions of the trajectories for path planning by using the covariances learned in the GMR.

Methods

Setup

The physical setup is illustrated in fig. 1. To compensate for camera motion and possible displacements of the task holder, the task coordinate system is tracked visually using a marker. In the following sections, we therefore assume that the 3D coordinates are provided in the same reference coordinate system, called the task coordinate system.

Motion learning

An instance of the task is denoted by $(\mathcal{T}, \mathcal{M})$, where $\mathcal{T} = (T_{1:\tau}, R_{1:\tau})$ is the 6 DOF tool trajectory, represented by the tool translation and rotation at each time step. $\mathcal{M} : u \in [0, 1] \longrightarrow \mathbb{R}^3$ is a 3D spline reconstruction of the wire used in the task.

Multiple task instances $(\mathcal{T}_i, \bar{\mathcal{M}})_{1\leq i\leq m}$ are used to build a representation of the task. The trajectories used in the learning process are assumed to have been performed on the same 3D model $\bar{\mathcal{M}}$. If variations in the 3D models exist, alignment techniques based on dynamic time warping (Sakoe and Chiba 1978) could be used as in (Reiley, Plaku, and Hager 2010) to align the trajectories on a reference 3D model prior to the learning.

The learning approach uses Gaussian Mixture Regression to construct a function $g : T \in \mathbb{R}^3 \longrightarrow R \in SO(3)$. Rotations are represented by quaternions and each trajectory consists of multiple 7-dimensional vectors. The construction approach is similar to the one presented in (Calinon, Guenter, and Billard 2006). A main difference is the use of the translation vector as input of our regression instead of the time step, since our objective is to perform partial automation using real-time input from the operator.

We first calculate a Gaussian Mixture Model (GMM) consisting of n Gaussians $G_k = \{\vec{\mu}_k, \mathbf{\Sigma}_k\}$, with associated priors π_k, from the vectors of the trajectories $\mathcal{T}_i$. The model is initialized using k-Means and trained with the Expectation-Maximization algorithm. The regression g associating a rotation R to a translation T is then computed from projections of the GMM: for each Gaussian, the covariance matrices and the mean vectors are decomposed into their translational and rotational parts

$$\mathbf{\Sigma}_k = \begin{pmatrix} \mathbf{\Sigma}_k^T & \mathbf{\Sigma}_k^{TR} \\ \mathbf{\Sigma}_k^{RT} & \mathbf{\Sigma}_k^R \end{pmatrix} \tag{1}$$

$$\vec{\mu}_k = \begin{pmatrix} \vec{\mu}_k^T \\ \vec{\mu}_k^R \end{pmatrix}. \tag{2}$$

The partial contributions of each Gaussian to R are

$$R_k = \mathbf{\Sigma}_k^{RT} {\mathbf{\Sigma}_k^T}^{-1} (T - \vec{\mu}_k^T) + \vec{\mu}^R. \tag{3}$$

Using the technique for averaging quaternions outlined in (Markley et al. 2007), the result of the regression is the weighted average of these contributions. The contribution of each Gaussian G_k is weighted by the a posteriori probability of T: $w_k = \frac{\pi_k \mathcal{N}(T, \mathbf{\Sigma}_k^T, \vec{\mu}_k^T)}{\sum_{i=1}^N \pi_i \mathcal{N}(T, \mathbf{\Sigma}_i^T, \vec{\mu}_i^T)}$.

Environment modeling

To cope with variations of the environment amongst different task instances, the task is associated with an environment model. The model consists of a 3D reconstruction of the scene obtained by using the calibrated stereo camera system of the da Vinci robot. Using the implementation of the Semi-Global Matching (Hirschmueller 2008) provided in the OpenCV library, a correspondence analysis of the stereo image pair is performed. A point cloud of the wire is then computed via stereo triangulation. Due to the small baseline of the camera (5mm), small correspondence errors, resulting from an inaccurate calibration or from illumination artifacts, lead to large errors in the reconstruction, which cause the point cloud to be noisy. We therefore fit a spline $\mathcal{M}(u)$ to the point cloud to obtain a smoother reconstruction.

Automation

We control the tool in a cartesian coordinate frame. Automating its rotation is equivalent to computing a function

$$f(t, T_{1:t}, R_{1:t-1}, \mathcal{M}) \longrightarrow R_t$$

where $T_{1:t}$ are all the observed translations up to current time t, $R_{1:t-1}$ are all the previously set rotations and $\mathcal{M}$ is the model of the environment for the task at hand.

In this paper, we investigate two methods for automatically rotating a tool being translated by a user. *Method 1* uses the model of the environment and no other prior knowledge to automate the rotation. By doing this, *Method 1* basically tries to stubbornly follow the wire. *Method 2* maps current observations to the learned representation of the task in order to retrieve a learned orientation for the tool. This rotation is then adapted to suit the actual model of the environment, which may differ from the model used in the learning process. While, like *Method 1*, *Method 2* also follows the wire, it tries to remain close to the original demonstrations.

Method 1: Automation from 3D only

When automating the rotation of the tool from 3D information only, no a-priori information regarding the task is required. Let p be the projection of current position T_t of the tool on the spline with $p = \mathcal{M}(u_p)$ and $\vec{v}_p$ be the gradient of the spline at parameter u_p.

As can be seen in fig. 2, when holding the ring, the x-axis of the tool is parallel to the direction of travel, if the wire passes through the ring.Therefore, to follow the curvature of the wire, the x-axis of the tool has to be parallel to the wire. To achieve this, we extract the x-axis $\vec{R}_{t-1}^x$ from the previous rotation R_{t-1} and, using the angle $\cos^{-1}\left(\frac{\vec{R}_{t-1}^x \cdot \vec{v}_p}{||\vec{v}_p||}\right)$ and the axis $\vec{R}_{t-1}^x \times \frac{\vec{v}_p}{||\vec{v}_p||}$, calculate a rotation ΔR that rotates the tool orientation given by R_{t-1} so that the x-axis is parallel to the wire. The new orientation of the tool becomes $R_t = \Delta R \cdot R_{t-1}$.

Method 2: Automation from Learning and 3D

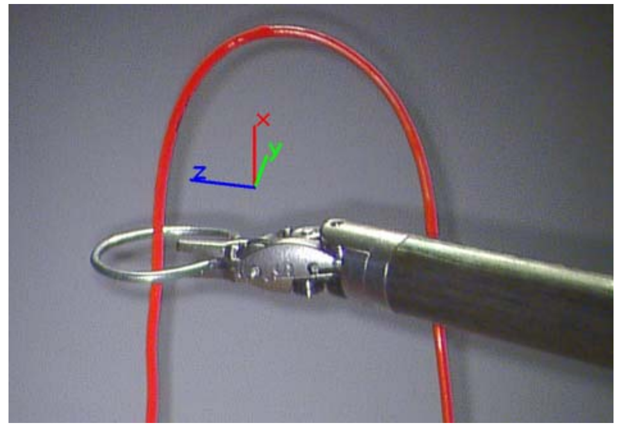


Figure 2: Coordinate system of the tool.

This method makes use of the previously learned regression and of the reference environment model $\bar{\mathcal{M}}$, which was recorded at the same time as the user demonstrations. To account for changes in the environment, a mapping $w : [0,1] \longrightarrow [0,1]$ between $\mathcal{M}$ and $\bar{\mathcal{M}}$ is obtained by first rigidly registering the models and then by applying Dynamic Time Warping on the two splines. A non-rigid mapping between two models is illustrated in fig. 3.

Using w and the projected point p of T_t onto the current model of the environment, we compute the corresponding point $\bar{p} = \bar{\mathcal{M}}(w(u_p))$ on model $\bar{\mathcal{M}}$. This point is then used to regress a rotation $g(\bar{p})$. Since the returned rotation does not take into account any potential physical changes between $\mathcal{M}$ and $\bar{\mathcal{M}}$, the orientation of the tool will not necessarily follow properly that of the new wire. Hence, large changes in the model curvature may impede a successful passing of the ring. We therefore transform the rotation returned by the regression with the technique outlined under method 1, so that its x-axis is parallel to the spline. This results in rotation R_t.

By fusing the result from regression with the result from the 3D-reconstruction, the rotation returned by the GMR can be adapted to account for changes in the environment, while still staying close to a trajectory learned from demonstration. The objective is to allow the ring to be passed along a deformed wire, while still keeping the trajectory intuitive and comfortable for the operator using the master manipulators.

Experiments

Learning

Before actually training the GMR, we determine the optimal number of Gaussians needed for modeling the task to avoid over-fitting. To this end, we divide our training data into a training set and a testing set. We then train multiple GMRs with up to 40 Gaussians and calculate the average and minimum angular error between the quaternions of the demonstrated and of the regressed rotation on both sets over 50 k-means initializations. These errors are indicated in fig. 4. Based on these results, we choose to use a GMR with 28

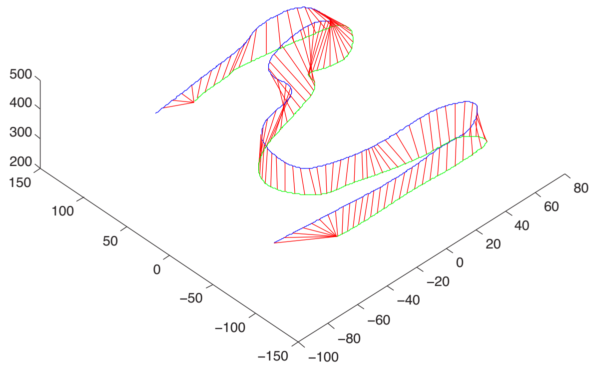


Figure 3: Visual correspondences between two 3D models.

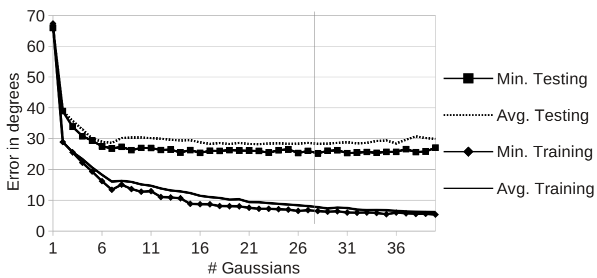


Figure 4: Angular error as a function of the number of Gaussians.

Gaussians in the following experiments. To train the Gaussians, 6 demonstrations from an expert user were used.

Results

To test the two methods outlined above, we constructed two tortuous wires with a similar but different curvature (see fig. 5). One wire is used to learn both the GMR and the reference model of the environment for the task representation. The other wire is used for testing.

Testing with *method 1* shows that while the information provided by the 3D reconstruction of the current wire is enough to successfully transfer the ring from one end of the wire to the other, the resulting trajectory is not necessarily intuitive or comfortable for the operator. The method calculates a minimal rotation to change the tool's orientation, but does not take into consideration whether the trajectory dictated by the new orientation can be easily followed by the master controllers.

We are able to transfer the ring more conveniently using *method 2*. The adaption of the data provided by the GMR through the current model of the environment assures that the resulting orientations stay as close as possible to the demonstration, making the experience of following the wire more intuitive and natural for the operator than when using *method 1*.

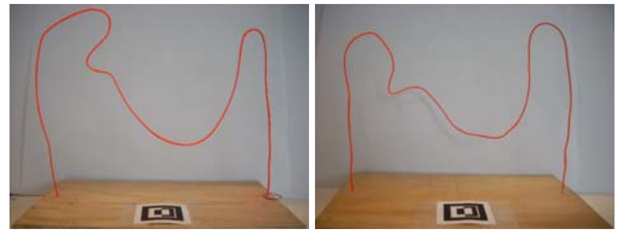


Figure 5: The two wires used in the experiments.

The differences in the trajectories for the two methods are illustrated in fig. 6. The figure shows how the two methods handle the same portion of the wire. With *method 1*, the instrument is passed over the wire, while method 2 rotates the tool in such a way that its trajectory runs beside the wire.

When taking into consideration that the arm of the operator has to mimic the trajectory of the tool, *method 1* has the disadvantage that the movement relies completely on a wrist roll. Since the hand of the operator will be parallel to the wire, the wrist can only be rolled 90° in either direction, which means that the operator would have to use the upper arm as well.

Method 2 makes use of wrist roll, yaw and pitch, therefore enabling the user to traverse the curve using only the lower arm, which is more natural.

(Bodenstedt, Padoy, and Hager 2012) presents a study on 10 users of the aforementioned approaches. The study shows that methods based on prior demonstration not only feels more comfortable for the user, but also decrease the time required to complete the task when compared to a method based solely on vision.

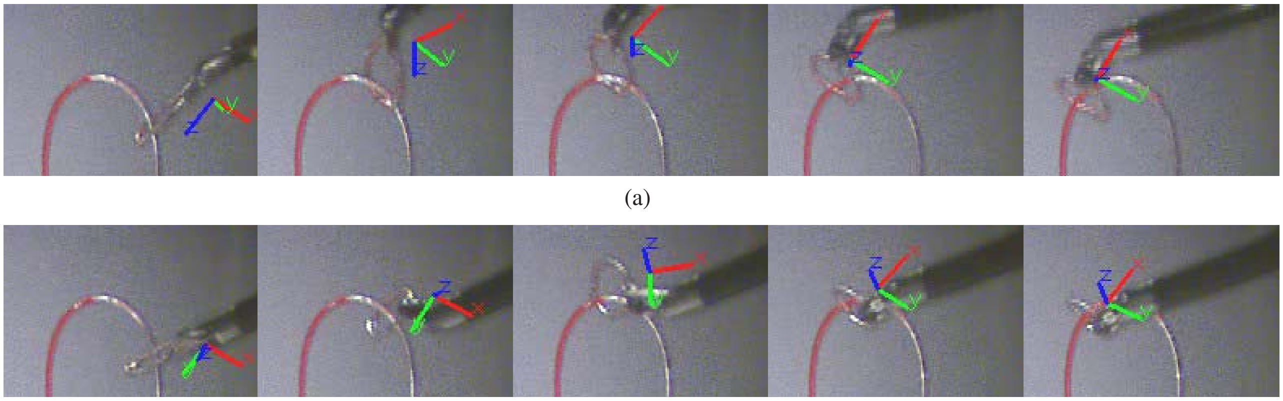
Discussion

Due to the fact that our chosen method for the correspondence analysis of the stereo images searches along epipolar lines for candidate matches, parts of the wire that ran along an epipolar line cannot be unambiguously reconstructed in all situations, leaving gaps in the point cloud. Errors in the camera calibration can also contribute to mismatches in the correspondence analysis and to noise in the stereo triangulation. Since these gaps and erroneously reconstructed points are taken into account when fitting the spline, the resulting spline may contain bumps or could otherwise be incomplete, so that it would not model reality correctly. If a faulty reconstruction is obtained, this can cause issues with the passing of the ring, such as small unnecessary rotations or, in the worst case, larger rotations that can make the passing impossible.

Due to small errors in the kinematic chain, the position returned for the tool is also not completely accurate. This sometimes results in incorrect projections along the splines. Tool tracking from the videos would help to reduce this inaccuracy.

Conclusion

In this work, we present a method for sharing control between an operator and a tele-manipulated robot, in which the



(a)

(b)

Figure 6: Visual comparison of *method 1* (top) and *method 2* (bottom) along one portion of the wire. (a) shows a trajectory that minimizes the difference between consecutive rotations. The trajectory displayed in (b) consists of a sequence of rotations that feels more natural to the operator.

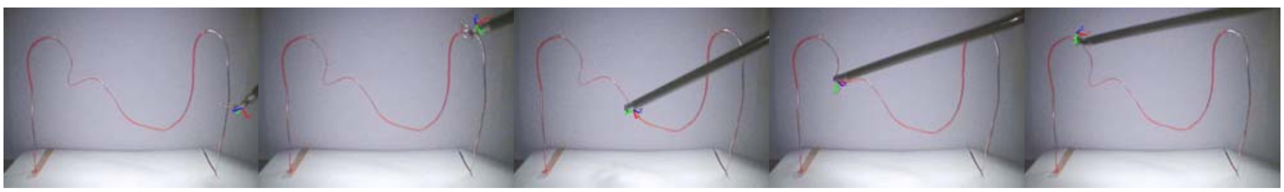


Figure 7: Picture sequence obtained when employing *method 2* for assistance.

operator controls a subset of the degrees of freedom while the robot automates the others based on the operator's intent. The objective is to learn from the daily usage of the robot how to simplify the tele-operation in repetitive tasks, while leaving partial control to the operator. If required, the operator can immediately recover full control of the remote tools. We use demonstrated data to build an a-priori model of the motion involved in the task manipulation. To perform the automation, we propose an approach that combines a 3D model of the environment with a learned regression from the tool translation to the tool rotation. The learned model permits to provide a partial automation that makes the remaining control of the master manipulators closer to the usual operator's experience. The approach is demonstrated on a difficult manipulation task involving the dexterous transfer of a ring along a tortuous wire, using one wire for learning and a deformed wire for demonstrating the actual transfer.

There exists several ways to extend this work. Since most tasks have a natural structure based on subtasks, it would be interesting to contextualize the partial automation based on the current subtask, so as to automate the most relevant DOF for the particular subtask objective. The most challenging aspect is however the perception of the environment and the learning from demonstration of generic models. A future direction is therefore to extend the GMR modeling so that it directly includes parameters describing the environment deformations.

References

Argall, B. D.; Chernova, S.; Veloso, M.; and Browning, B. 2009. A survey of robot learning from demonstration. *Robotics and Autonomous Systems* 57:469–483.

Bodenstedt, S.; Padoy, N.; and Hager, G. 2012. Evaluation of a learning-based partially automated approach for assistance in tele-robotic manipulation. In *Proceedings of MICCAI Workshop on Modeling and Monitoring of Computer Assisted Interventions (M2CAI), to appear*.

Calinon, S.; Guenter, F.; and Billard, A. 2006. On Learning the Statistical Representation of a Task and Generalizing it to Various Contexts. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2978–2983.

Calinon, S. 2009. *Robot Programming by Demonstration - a Probabilistic Approach*. EPFL Press.

Guo, G.; Tarn, T. J.; Xi, N.; and Bejczy, A. K. 1995. Fusion of human and machine intelligence for telerobotic systems. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 3110–3115.

Guthart, G., and Jr., J. K. S. 2000. The intuitive$^{\text{tm}}$ telesurgery system: Overview and application. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 618–621.

Hirschmueller, H. 2008. Stereo processing by semi-global matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30:328–341.

Kragic, D., and Hager, G. 2003. Task modeling and specification for modular sensory based human-machine cooperative systems. In *Proceedings of the IEEE/RSJ (IROS) International Conference on Intelligent Robots and Systems*, 3192–3197.

Marayong, P.; Li, M.; Okamura, A. M.; and Hager, G. D. 2003. Spatial motion constraints: Theory and demonstrations for robot guidance using virtual fixtures. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1954–1959.

Markley, F. L.; Cheng, Y.; Crassidis, J. L.; and Oshman, Y. 2007. Averaging Quaternions. *Journal of Guidance Control and Dynamics* 30:1193–1197.

Nageotte, F.; Zanne, P.; Doignon, C.; and de Mathelin, M. 2009. Stitching planning in laparoscopic surgery: Towards robot-assisted suturing. *The International Journal of Robotics Research* 28(10):1303–1321.

Padoy, N., and Hager, G. D. 2011. Human-machine collaborative surgery using learned models. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 5285–5292.

Reiley, C. E.; Plaku, E.; and Hager, G. D. 2010. Motion generation of robotic surgical tasks: Learning from expert demonstrations. In *32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*.

Rosenberg, L. B. 1993. Virtual fixtures: Perceptual tools for telerobotic manipulation. In *IEEE Annual International Symposium on Virtual Reality*, 76–82.

Sakoe, H., and Chiba, S. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoust. Speech Signal Process.* 26(1):43–49.

van den Berg, J.; Miller, S.; Duckworth, D.; Hu, H.; Wan, A.; Fu, X.-Y.; Goldberg, K.; and Abbeel, P. 2010. Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2074–2081.

Yang, J.; Xu, Y.; and Chen, C. S. 1993. Hidden markov model approach to skill learning and its application in telerobotics. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 396–402.

Ye, G., and Alterovitz, R. 2011. Demonstration-guided motion planning. In *International Symposium on Robotics Research (ISRR)*.