Journal of Mechanisms and Robotics. Received October 13, 2016;
Accepted manuscript posted January 11, 2017. doi:10.1115/1.4035684

Force-Controlled Exploration for Updating Virtual Fixture Geometry In Model-Mediated Telemanipulation

Long Wang[1], Zihan Chen[2], Preetham Chalasani[2],
Rashid M. Yasin[1], Peter Kazanzides[2], Russell H. Taylor[2] and Nabil Simaan[1]

[1]Advanced Robotics and Mechanism Applications
Department of Mechanical Engineering
Vanderbilt University
long.wang,rashid.m.yasin,
nabil.simaan@Vanderbilt.edu

[2]Laboratory for Computational Sensing and Robotics
Department of Computer Science
Johns Hopkins University
zihan.chen, pchalas1,
pkaz,rht@jhu.edu

ABSTRACT

This paper proposes an approach for using force-controlled exploration data to update and register an a-priori *virtual fixture geometry to a corresponding deformed and displaced physical environment. An approach for safe exploration implementing hybrid motion/force control is presented on the slave robot side. During exploration, the shape and the local surface normals of the environment are estimated and saved in an exploration data set. The geometric data collected during this exploration scan is used to deform and register the* a-priori *environment model to the exploration data set. The environment registration is achieved using a deformable registration based on the coherent point drift method. The task-description of the high-level assistive telemanipulation law, called a virtual fixture (VF), is then deformed and registered in the new environment. The new model is updated and used within a model-mediated telemanipulation framework. The approach is experimentally validated using a da-Vinci research kit (dVRK) master interface, a dVRK patient side manipulator, and a Cartesian stage robot. Experiments demonstrate that the updated VF and the updated model allow the users to improve their path following performance and to shorten their completion time when the updated path following VF is applied. The approach presented has direct bearing on a multitude of surgical applications including force-controlled ablation.*

1 INTRODUCTION

During robot-assisted and computer-aided surgery, surgeons attempting to carry out path following tasks such as ablation or dissection along a desired anatomical path are challenged by the flexibility of the underlying anatomy. Examples of this task can be found in cardiac ablation for electrophysiology and in cholecystectomy where dissection to expose the hepatic and cystic ducts are required. The introduction of image-guided surgery assists surgeons in avoiding critical anatomical structures. In addition, robot assisted image-guided surgery improves the coupling between surgical pre-planning and surgical execution. The success of this coupling hinges on successful registration between the *a-priori* model of the surgical plan and the anatomy as obtained from pre-operative imaging. One key challenge to the paradigm of image-guided surgery is the fact that flexible organs are susceptible to deformation due to gravitational forces or changes in their boundary conditions when the connective tissues around the target organ are removed or displaced to gain access to that organ.

This challenge of dealing with an environment that deforms relative to an *a-priori* model has led to a plethora of works on deformable registration methods (see [1] for an up to date review). Typically these methods focus on how to register intra-operative to pre-operative images. The pre-operative images usually include CT, MRI and ultrasound while intra-operative images use ultrasound primarily. Other approaches included the use of vision [2] or conoscopy [3–5] for updating the environment shape. While these methods work, they are typically encumbered with cost and difficulty in employing intra-operative imaging. Furthermore, vision-based techniques only work for non-obfuscated fields with a line-of-sight restriction. In this work, we take a different approach which uses information collected through robotic force-controlled exploration to obtain a model of the environment. This approach overcomes some of the above mentioned challenges but presents new challenges in designs of safe exploration algorithms in the absence of an environment model.

In addition to the registration challenges, tasks requiring force-controlled interaction with the anatomy present additional challenges within the paradigm of telemanipulation. Direct force feedback from the slave force sensing to the master force rendering presents challenges of stability and robustness against registration errors, environment impedance parameters and more-importantly time delays. One approach that helps overcome some of these challenges is model-mediated telemanipulation in which the user interacts with a haptic model of the environment and the slave interacts with the environment [6]. However, current frameworks for model-based telemanipulation suffer from dependency on the haptic interaction with the environment model which may be inaccurate. Previous works on model based telemanipulation demonstrated improved user performance despite large time delays, in which hybrid force/motion control was used to accommodate misalignment of the environment with respect to the slave robot (e.g. [7, 8]). A key limitation to model-based telemanipulation is the process of using exploration data to update the model in the case of deformable environments.

The methodology described in this work is as follows: given an *a-priori* model and an associated telemanipulation virtual fixture descriptor, devise a framework allowing collection of data during force-controlled slave robot exploration and propose a method for correcting the *a-priori* model for deformation and registration errors. The following assumptions will be used: i) an *a-priori* model is given with the virtual fixture descriptor; ii) an impedance master and an admittance slave robot capable of force sensing are provided.

Researchers have been investigating the use of contact information during interaction with organs. Extraction of stiffness information using mechanical imaging via tactile sensor arrays were demonstrated in [9–11]. Rolling mechanical imaging was obtained using a force-sensitive probe in [12] and probing motion (indenting tissue in the depth direction) was investigated in [13, 14]. Force controlled telemanipulation of continuum robots was used in [15] to map the geometry and stiffness of the environment. Dynamic excitation of tissue was applied to estimate impedance parameter in [16]. This work differs from these previous works by attempting to use geometric information obtained through force-controlled exploration to register and correct a preoperative *a-priori* model of the surgical plan.

In this paper, the use of force-controlled scanning of tissue is explored as a means of gathering information for registering and updating the pre-operative model. The most relevant works on this topic include [17] where constrained Kalman filtering was employed to use the contact and estimated stiffness information to obtain a rigid registration of the model. Similarly, the complementary model update (CMU) method [18] was presented as a more robust approach for rigid registration using stiffness and geometry information to improve registration. In addition, Gaussian processes were used to simultaneously estimate the stiffness and surface of an organ using continuous palpation motion [19]. In [20], a Bayesian optimization framework was introduced to guide probing to maximize information gain, thus avoid probing the entire organ, while registering the predicted stiffness to an *a-priori* geometric model. These works account for local deformation induced during the probing process yet disregard a potential global deformation of the organ. Global deformation of a model is addressed in [21, 22]. This work complements these efforts by testing a naive approach which relies on a force-controlled scan of the organ and which is coupled with a deformable registration. The method is not time-efficient so we anticipate using it to initialize our registration while subsequently achieving a continuous model update using the other approaches listed above.

This work was initially presented in [23]. Compared to our prior work in [23], this work presents the same two key contributions while extending the work in two ways. First, the work provides additional evaluation of our VF geometry update using force-controlled exploration on a clinically deployable system (the da Vinci Research Kit (dVRK)). Second, the work provides a more thorough exposition of our exploration and deformable registration and the high-level system integration approach.

Two key contributions are presented in this work. First, a method for force-controlled telemanipulated exploration is proposed to collect geometric data of the deformed environment. With the exploration data, an *a-priori* model of the environment is registered and corrected using deformable registration based on coherent point drift [21]. Using this approach, a flexible environment model is updated for deformation and registration errors.

The second contribution is technical in nature and is the presentation of a highly modularized framework of system integration using the *cisst* package and Simulink® Real-Time. This framework provides assistive virtual fixtures on the master side while supporting model-mediated telemanipulation. It also provides intelligent control behaviors on the slave side to support force-controlled exploration and telemanipulation.

2 TELEMANIPULATION FRAMEWORK

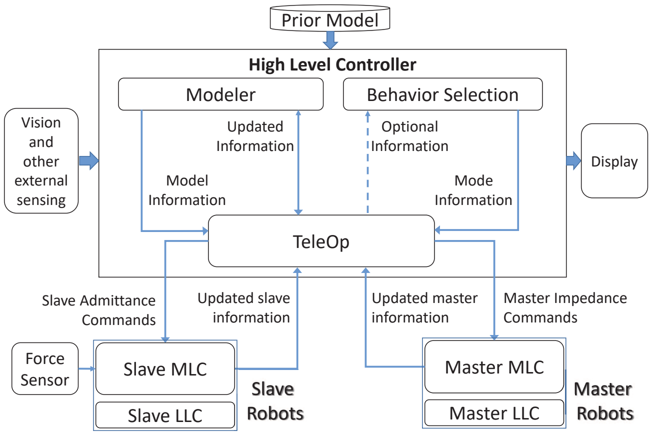

FIGURE 1. System Architecture

Our telemanipulation framework, as shown in Fig. 1, is based on the JHU "Surgical Assistant Workstation (SAW)"

Journal of Mechanisms and Robotics. Received October 13, 2016;
Accepted manuscript posted January 11, 2017. doi:10.1115/1.4035684

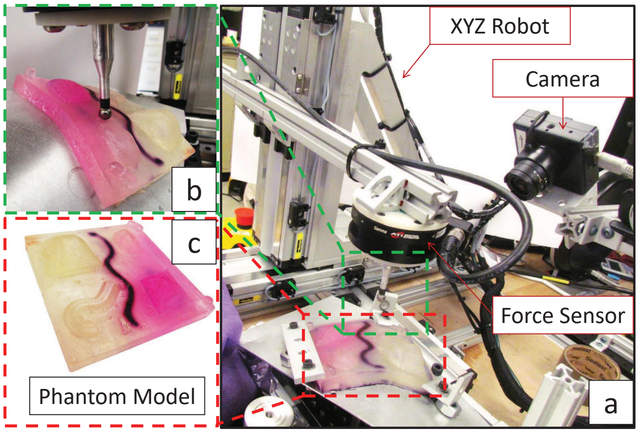

FIGURE 2. A custom Cartesian Slave Robot System: (a) Experiment setup, (b) Ball Probe Finger ATI Force Torque Sensor, (c) A phantom model used in experiment

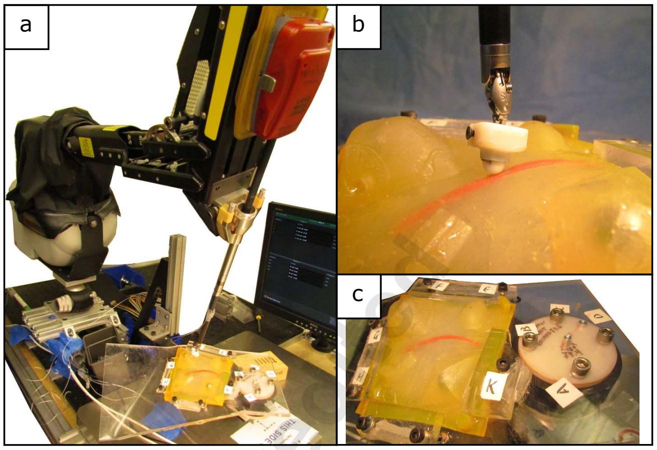

FIGURE 3. PSM Experiment Setup: (a) Experiment setup, (b) Ball probe finger adapter integrated with EM tracker, (c) A phantom model mounted on a force plate

software environment [24] and the da Vinci Research Toolkit (dVRK) [8, 25]. This environment supports multiple telemanipulation hardware and software components in a mix-and-match fashion. For our current research, we use dVRK master tool manipulators (MTMs) and slave manipulators choosing from either dVRK patient side manipulators (PSMs, as in Fig. 3) or a custom Cartesian robot seen in Fig. 2. Each slave system manipulates a force-sensing probe comprising either ATI Nano-17 or an ATI Gamma-SI-130-10 Force/Torque sensor with a ball probe finger for contacting tissue, as in Fig. 2. In Fig. 3, the phantom model is equivalently mounted on an ATI Gamma Gamma-SI-130-10 Force/Torque sensor for implementation ease to verify the framework of this paper. The results reported in section 5 were obtained with both the Cartesian robot and a dVRK PSM. The component-based SAW software is very modular and its processes may be run on a highly distributed computing environment. Several key processes are discussed below.

The *Master Controller* process is responsible for the control of the Master manipulator hardware. This process consists of two sub-processes: a *Master Mid-Level Controller (MLC)* which communicates with the *TeleOp* process (described below) and a *Master Low-Level Controller (LLC)* which communicates with the Master hardware and performs basic joint-level servo control functions. The *Master MLC* runs as a clock-driven process at a sampling rate of 500 Hz and the *Master LLC* runs at 1.5 kHz. The *Master MLC* receives impedance specification commands from the *TeleOp* process and translates them into an appropriate form for execution by the *Master LLC*. The *Master MLC* process also returns state information to the *TeleOp* process, including joint and Cartesian positions and velocities, Master gripper openings, and forces and torques exerted by the Master on the surgeon's hand.

The *Slave Controller* process is responsible for control of the Slave hardware. Like the Master Controller, this process consists of a *Slave Mid-Level Controller (MLC)* which communicates with the *TeleOp* process and a *Slave Low-Level Controller (LLC)* which communicates with the Slave hardware. The *Slave MLC* runs as a clock-driven process at a sampling rate of 500 Hz and the *Slave LLC* runs at 1000 Hz. The Slave Controller also contains a force sensing component that reads the slave's force sensor and computes forces exerted on the finger probe. The *Slave MLC* receives admittance commands and virtual fixture specifications from the *TeleOp* process and translates them into Cartesian or joint position/velocity commands that are passed on to the *Slave LLC*. The *Slave MLC* receives state information from the *Slave LLC*, combines this information with other Slave Controller information (e.g., forces, contact information) and passes the combined state information back to the *TeleOp* process.

The *TeleOp* process is the central control point for the system. This process runs as a real-time, clock driven process (at 500 Hz). It is responsible for managing communications among the *Master Controller*, *Slave Controller*, *Modeler*, and higher-level *Behavior Selection* processes. It is also directly responsible for real-time telemanipulation behavior. The *TeleOp* process receives state information from the *Master MLC* and *Slave MLC* and passes this information on to the Modeler and the Behavior Selection Process. Based on the entire combined state information (Master, Slave, Model, etc.) and the current behavior mode, the *TeleOp* component determines appropriate admittance commands and sends them to the Slave Controller. Similarly, it also determines appropriate impedance commands and sends them to the *Master Controller*. The *TeleOp* component also has a special "autonomous scanning" behavior in which the slave is issued a series of admittance commands causing the slave manipulator to move at a constant velocity across a surface while exerting regu-

Journal of Mechanisms and Robotics. Received October 13, 2016;
Accepted manuscript posted January 11, 2017. doi:10.1115/1.4035684
Copyright (c) 2017 by ASME

lated force normal to the surface.

The *Behavior Selection* process runs in the background and communicates with the *TeleOp* process to inform it of changes in desired behavior (e.g., simple position following telemanipulation, model-mediated telemanipulation, telemanipulation with force bias, telemanipulation with superimposed palpation motion, etc.). It receives state information from the *TeleOp* process and the Modeler, as well as direct input from the user. It also will manage information displays and other user interfaces not directly involving telemanipulation.

The *Modeler* process is responsible for maintaining a model of the manipulation environment. In the current implementation, the model consists of a triangulated surface mesh representation of an anatomic organ or phantom object. This mesh is augmented with a spline curve representing a path on the surface that the robot is to follow. In future versions, the mesh will also be annotated with stiffness information associated with each triangle in the mesh. In our prior work [16] we demonstrated force-controlled estimation of flexible environment constraints and impedances. In [17] we adapted the constrained extended Kalman filter to allow taking into account geometric and stiffness information to benefit registration in flexible environments. In this work we are extending these results to demonstrate the utility of force-controlled exploration for updating the model. The *Modeler* process also is responsible for maintaining the registration between the slave robot and the model, based on surface contact information provided by the Slave process.

2.1 Slave Robot Controllers

To enable hybrid force/motion capabilities, two slightly different versions of Mid-Level Controllers are developed in Matlab Simulink® Real-Time™environment, for a dVRK PSM and a custom Cartesian robot, shown as in Fig. 4 and Fig. 5, respectively. The dVRK PSM *MLC-LLC* is implemented as an admittance type controller while the Cartesian robot is implemented as a hybrid motion/force with dynamic compensation controller. The motion/force projection mechanisms and the Cartesian robot controller are both motivated by the works of Khatib [26] and Featherstone [27].

The MLC accepts commands from *TeleOp* in a format either of position, admittance, or desired force. A velocity command $\dot{\mathbf{x}}_{des}$ is generated based on a resolved rates algorithm [28], given the desired position $\mathbf{x}_{des}$ and the current slave position. Another velocity command $\dot{\mathbf{x}}_{adm}$ is generated given an admittance force command (or force error in PSM's case) and admittance gains. In the case of PSM controller (Fig. 4), $\dot{\mathbf{x}}_{des}$ and $\dot{\mathbf{x}}_{adm}$ are decomposed by the *Motion Force Projection* block, resulting in $\dot{\mathbf{x}}_{des_\perp}$ and $\dot{\mathbf{x}}_{adm_\perp}$ respectively. The added velocity command $\dot{\mathbf{x}}_{cmd}$ is then sent to the PSM LLC. In the case of Vanderbilt Cartesian robot controller, $\dot{\mathbf{x}}_{des}$ and $\dot{\mathbf{x}}_{adm}$ are first combined to be the motion command $\dot{\mathbf{x}}_{cmd}$. This motion command $\dot{\mathbf{x}}_{cmd}$ and the desired force $\mathbf{f}_{des}$ are decomposed by the *Motion Force Projection* block. The projected velocity and force command $\dot{\mathbf{x}}_{cmd_\perp}$ and $\mathbf{f}_{cmd_\perp}$ are then sent to the LLC of the Cartesian robot. The projection matrices in Fig. (4, 5) are given as follows [26, 27].

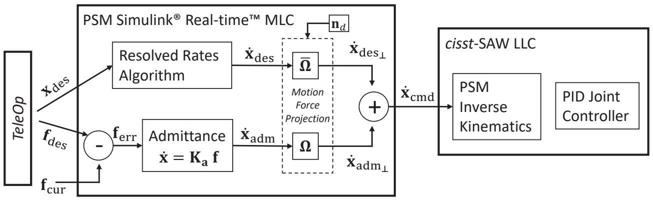

FIGURE 4. PSM Force-controlled Slave MLC-LLC Controller

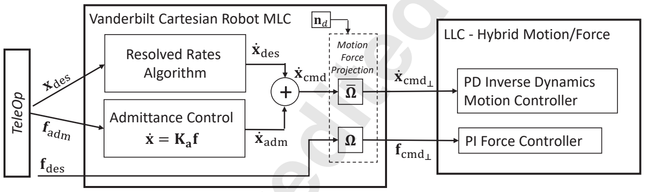

FIGURE 5. Vanderbilt Slave MLC-LLC Controller

$$
\begin{aligned}
\Omega = \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T &= \mathbf{I} - \bar{\Omega}, \\
\bar{\Omega} = \mathbf{T}(\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T &= \mathbf{I} - \Omega, \\
\mathbf{N} \in \mathbb{R}^{m \times r}, \mathbf{T} &\in \mathbb{R}^{m \times (m-r)}
\end{aligned}
\tag{1}
$$

where m is the total task space dimension and r is the force/torque controlled space dimension, in our research $m = 3, r = 1$. As a result, $\mathbf{N} = \mathbf{n}_d = [n_x, n_y, n_z]^T$ specifies the *desired* force control direction.

2.2 Master Impedance Controller

The MLC is implemented as an impedance type controller, which allows combining different control goals by simply adding desired joint torques computed separately. As shown in Fig. 6, gravity compensation is rendered at any time and an impedance type virtual fixture controller is running in parallel, taking commands from the teleoperation component.

To define the virtual fixture controller behavior, *TeleOp* sets force position compliance frame $\mathbf{F}_c = [\mathbf{R}_c, \mathbf{p}_c]$ defined in master base frame. The virtual fixture law also uses position stiffness gain vectors $\mathbf{k}^{(+)}, \mathbf{k}^{(-)}$, position damping gain vectors $\mathbf{b}^{(+)}, \mathbf{b}^{(-)}$ and force bias terms $\mathbf{a}^{(+)}, \mathbf{a}^{(-)}$. The pairs are used to distinguish between movement toward the virtual fixture v.s. away from the virtual fixture boundary. Algorithm 1 shows how the desired force applied on the master tip is computed.

One advantage of this design is that it permits very fast haptic rendering of discontinuous impendence environments when the slave end effector is near the virtual fixture boundary, such as encountered when one is palpating or following an organ surface. It also permits very versatile descriptions of local virtual

Journal of Mechanisms and Robotics. Received October 13, 2016;
Accepted manuscript posted January 11, 2017. doi:10.1115/1.4035684
Copyright (c) 2017 by ASME

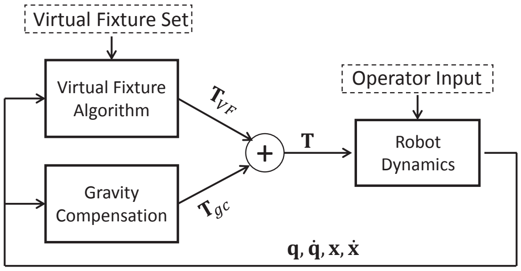
Virtual Fixture Set
Virtual Fixture Algorithm
Operator Input
T_{VF}
+
T
Robot Dynamics
Gravity Compensation
T_{gc}
$\mathbf{q}, \dot{\mathbf{q}}, \mathbf{x}, \dot{\mathbf{x}}$

FIGURE 6. Master MLC Impedance Type Controller: $\mathbf{q}$ - joint position; $\dot{\mathbf{q}}$ - joint velocity; $\mathbf{x}$ - cartesian position; $\dot{\mathbf{x}}$ - cartesian velocity; T - total joint torque applied to robot; T_{VF} - joint torque from virtual fixture controller T_{gc} - joint torque from gravity compensation

Algorithm 1 Master Virtual Fixtures Controller

```
Given
$\mathbf{F} = [\mathbf{R},\mathbf{p}]$: current pose   $\dot{\mathbf{p}}$: current velocity        ▷ $\mathbf{F} \in SE(3)$
$\mathbf{F}_c = [\mathbf{R}_c,\mathbf{p}_c]$: position compliance frame w.r.t master
$\mathbf{k}^{(+)},\mathbf{k}^{(-)}$: stiffness gains  $\mathbf{b}^{(+)},\mathbf{b}^{(-)}$: damping gains
$\mathbf{a}^{(+)},\mathbf{a}^{(-)}$: force bias terms                   ▷ $\mathbf{k},\mathbf{b},\mathbf{a} \in \mathbb{R}^3$
Compute
if (Enabled) then
    $\mathbf{e} = \mathbf{F}_c^{-1}\mathbf{p} = \mathbf{R}_c(\mathbf{p}-\mathbf{p}_c)$                  ▷ position error
    ${}^c\mathbf{v} = \mathbf{R}_c^{-1}\dot{\mathbf{p}}$                         ▷ velocity written in $\{C\}$
    for $i \in \{x,y,z\}$ do                            ▷ each component
        if $(e_i \le 0)$ then $g_i = a_i^{(-)} + k_i^{(-)} e_i + b_i^{(-)c} v_i$
        else $g_i = a_i^{(+)} + k_i^{(+)} e_i + b_i^{(+)c} v_i$
        end if                          ▷ gains selection depends on error sign
    end for
    $\mathbf{g} = [g_x, g_y, g_z]^T$                       ▷ virtual fixture force in $\{C\}$
    $\tau = \mathbf{R}_c\mathbf{g}$              ▷ virtual fixture force in master base frame
end if
```

fixtures behavior, such as encountered in curve following. Further, it permits simple combinations of virtual fixture elements, such as combining surface following with curve following. It is simple to implement and provides a versatile command interface between the *TeleOp* process and the Master Controller.

2.3 Model-based virtual fixtures for surface following, palpation, and surface feature tracking

Although the Slave Controller is capable of implementing virtual fixtures incorporating both positional and force constraints using the methods described in [29], for the current paper, we rely on impedance commands to exert feedback forces on the surgeon's hands with Master manipulator, based on the current registered model. To simplify the discussion we will treat the Master, Slave, and Model coordinate systems as equivalent, i.e., a position $\vec{p}$ in the Master manipulator coordinates corresponds to position $\vec{p}$ in the Slave and Model. Thus, we will say that the Master is "in contact" with the Model if its current position $\vec{p}$ is on or below the surface of the Model.

For *surface following*, our goal is to exert a constant force normal to the surface while permitting the surgeon to move the robot freely across it. In this mode, *TeleOp* determines the closest point on the surface from the Master manipulator. This closest point is chosen as the origin of the compliance reference frame, as shown in Fig. 7, along with the surface normal at this point as positive Z axis with 0 positive gain and large negative gain. In the mean while, the X and Y axes can be chosen freely, with 0 gains, since motion along the surface is not limited. For *surface feature tracking* our goal is to assist the surgeon in tracing a predefined curve across the surface while still maintaining contact with the surface with a constant normal force. Along with the surface following virtual fixture, a preregistered curve guidance virtual fixture also starts rendering whenever the robot is close to the curve. Once started, *Teleop* then determines the closest point on the curve and the tangent direction of the curve at the closest point. Similarly, the surface normal at the closest point is picked as the positive Z axis of the compliance reference frame. The tangent direction serves as the X axis with zero gains. The Y axis is determined from the X and Z axes, with large gains for positive and negative directions, as shown in Fig. 8. The use of the model-mediated virtual fixture is demonstrated in *Multimedia Extension I*[1].

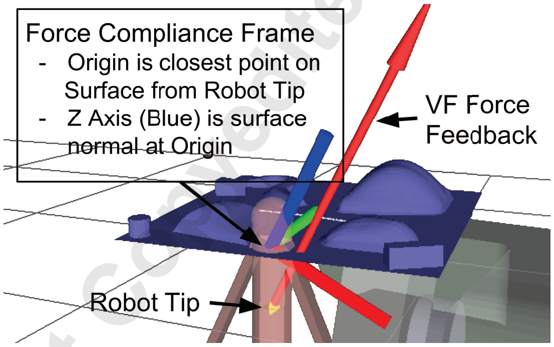
Force Compliance Frame
- Origin is closest point on Surface from Robot Tip
- Z Axis (Blue) is surface normal at Origin
VF Force Feedback
Robot Tip

FIGURE 7. Surface following frame with master robot tip and force feedback

3 FORCE CONTROLLED EXPLORATION

Using hybrid force/motion controller of the slave robots described in section 2 we can achieve a force-controlled exploration of the environment, i.e. a *surface following* function. Our previous work [16] used a similar *surface following* function where a Cartesian robot was controlled given a constant predefined force desired direction (i.e. $\mathbf{n}_d$ in Fig. 5 is constant and specified). In this work, the force desired direction is updated based on current

[1] `http://arma.vuse.vanderbilt.edu/images/stories/videos/long_jmr_mme1.mp4`

Journal of Mechanisms and Robotics. Received October 13, 2016;
Accepted manuscript posted January 11, 2017. doi:10.1115/1.4035684
Copyright (c) 2017 by ASME

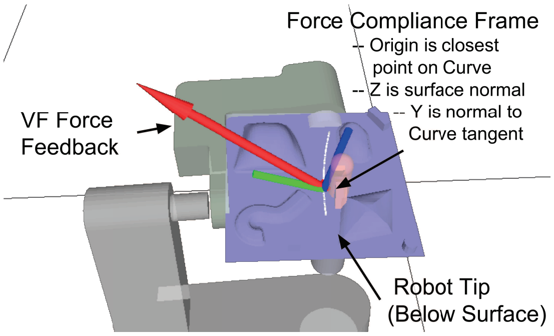
Force Compliance Frame
-- Origin is closest point on Curve
-- Z is surface normal
-- Y is normal to Curve tangent
VF Force Feedback
Robot Tip (Below Surface)

FIGURE 8. Curve following frame with master robot tip and force feedback

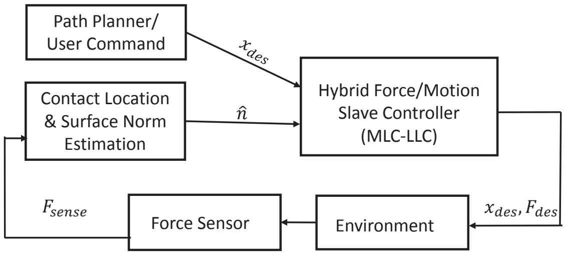
Path Planner/ User Command
x_{des}
Contact Location & Surface Norm Estimation
$\hat{n}$
Hybrid Force/Motion Slave Controller (MLC-LLC)
F_{sense}
Force Sensor
Environment
x_{des}, F_{des}

FIGURE 9. Force-controlled Exploration Strategy

estimation of the environment and robots with wrist orientation control are considered. The surface geometry is estimated and used as data for registration, which will be discussed in section 4. The exploration control strategy is described in Fig. 9. The hybrid force/motion slave controller accepts the position command $\mathbf{x}_{des}$ from either user command or path planning and the force regulating direction $\hat{\mathbf{n}}$ is set to be the current estimated contact surface normal.

3.1 Contact and Surface Normal Estimation

The contact location and surface normal estimation is shown in Fig. 10. The surface normal is computed using a highly simplified model and the force sensed from the environment, $\hat{\mathbf{n}} = \mathbf{f}_s/||\mathbf{f}_s||$. The calculation of $\hat{\mathbf{n}}$ is obtained through a moving average filter with a width of 30 samples obtained at a frequency of 1kHz. This model assumes negligible contact friction. During experiments, Glycerin was used as a highly lubricious medium to approximate this assumption. This is a reasonable approximation to lubricious tissue covered with bodily fluids during surgery.

Because friction does not dominate, this assumption does not impede task execution. However, for a more robust/generalized formulation there are a number of options to incorporate friction compensation for better surface normal execution in task completion. A simple method is to remove force projections in the direction of the tool tip velocity [30]. Other methods include the estimation of a constraint jacobian [31], force and position sensor fusion [32], or an adaptive learning controller for surface normal estimation [33]. These and similar methods can be adjusted and extended to fit the presented task of palpation of a flexible environment with unknown geometry. However, a full exploration of this topic is beyond the scope of this paper.

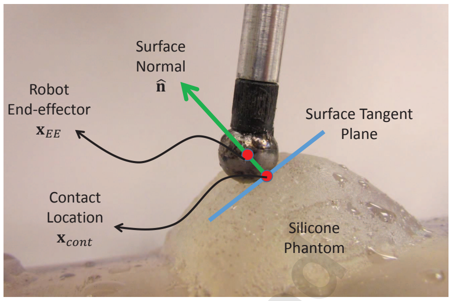

FIGURE 10. Contact Location and Surface Norm Estimation

The offset of the contact location with respect to the robot end-effector can be computed as $\mathbf{x}_{cont} = \mathbf{x}_{EE} - \hat{\mathbf{n}}r$. As shown in Fig. 10, when the robot is in contact with environment during the exploration, it is controlled such that its motion is constrained in the surface tangential plane and its force projection onto the surface normal is regulated to a specified magnitude by *TeleOp*.

3.2 Wrist Orientation Optimizer for Exploration

Different from the Cartesian robot in Fig. (2, 10) where only positions can be controlled, in the case of PSM control, the wrist orientation may be optimized for exploration advantage. As shown in Fig. 11, the robot base frame is denoted as $\{0\}$, the gripper frame as $\{G\}$ and the exploration environment frame as $\{E\}$. The $\hat{\mathbf{z}}$ axis of frame $\{E\}$ is determined by the surface normal direction $\hat{\mathbf{n}}$ while the $\hat{\mathbf{x}}$ axis is defined as the projection of exploration moving direction onto the surface tangential plane.

We propose two criterions to optimize the wrist orientation while exploring the environment. The first criterion is to align the force probe direction (gripper frame $\hat{\mathbf{z}}_g$) with the surface normal direction $\hat{\mathbf{n}}$. This is illustrated in Fig. 11(a), where the angle γ is marked as the minimization goal. It is desired because it minimizes the chance of other area outside of the probing ball touching the environment, which guarantees exploration contacts to be accurately on the probe sphere surface. The second criterion is to align the highest wrist stiffness direction with the exploration moving direction, which is illustrated as minimizing the angle ϕ in Fig. 11(b). The angle ϕ is defined between the axis $\hat{\mathbf{y}}_e$ and the projection of the gripper frame axis $\hat{\mathbf{x}}_g$ onto the surface tangential plane. We formulate this as a constrained optimization problem:

$$\begin{aligned} \underset{\mathbf{R}_g}{\text{maximize}} \quad & \hat{\mathbf{n}}^{\mathrm{T}}(-\hat{\mathbf{z}}_g) + \hat{\mathbf{y}}_e^{\mathrm{T}}\,\hat{\mathbf{x}}_{g_\perp} \\ \text{subject to} \quad & \dot{\mathbf{x}} = \dot{\mathbf{x}}_{des} \end{aligned} \tag{2}$$

Journal of Mechanisms and Robotics. Received October 13, 2016;
Accepted manuscript posted January 11, 2017. doi:10.1115/1.4035684
Copyright (c) 2017 by ASME

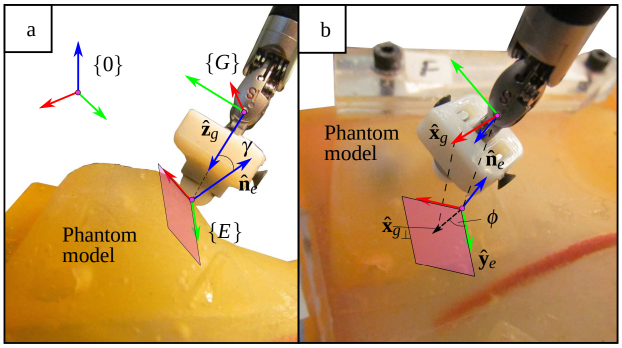

FIGURE 11. Orientation optimization for force-controlled exploration using a robot with wrist.

where $\hat{\mathbf{x}}_{g_\perp} = \bar{\Omega}\hat{\mathbf{x}}_g$ and $\bar{\Omega}$ is defined in equation (1).

3.3 Surface Exploration

Fig. 12a shows the path planning to explore the entire area of interest. The path can be given in an arbitrary plane and in experiments to optimize the scan resolution we chose a plane that was parallel to XY plane in the robot base. The user selected a starting point location and several via points as the reference points, shown as a red point and several green points in Fig. 12a. A 2D projection onto the robot base XY plane of these reference points were used to automatically generate a raster scan pattern coordinates $\mathbf{P}_{XY} \in \mathbb{R}^{N_p \times 2}$ that enclosed several "patches" where N_p is the number of reference points, e.g. in Fig. 12a $N_p = 148$. And $\mathbf{P}_{XY}$ combined with the current Z coordinate of the robot in real-time were sent as command positions to the slave MLC, which implements hybrid motion/force control for the Cartesian slave robot and hybrid position/admittance control for the dVRK PSM.

The 2D scan pattern $\mathbf{P}_{XY}$ was executed at a constant velocity of 1 mm/sec in the Cartesian robot case while the exploration speed in PSM was 4 mm/sec. Currently in the Cartesian robot, without acceleration measurement feedback, a very slow execution speed was selected to avoid pseudo force disturbance to contact and surface normal estimation stemming from loading mass dynamics. In future work, an inexpensive accelerometer may be installed to provide acceleration measurement. Hence the compensation of the dynamic effect from the loading mass can be provided in real-time, enabling a much faster scanning capability. Fig. 12 shows the actual estimated contact locations during the force-controlled exploration using the Cartesian robot and Fig. 13 shows the results using dVRK PSM robot. Demonstrations of force-controlled explorations using Cartesian robot and dVRK PSM are available in *Multimedia Extension II*[2] and *Multimedia Extension III*[3] respectively.

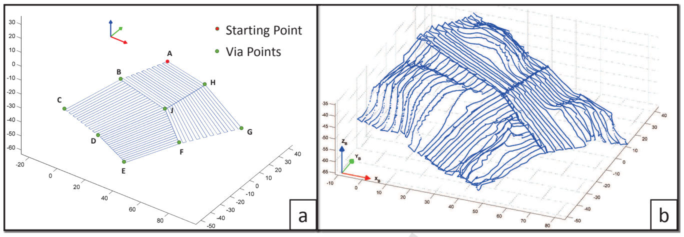

FIGURE 12. Force-controlled exploration using custom Cartesian Robot: (a) is the planned scan pattern, (b) is the actual scanned point cloud.

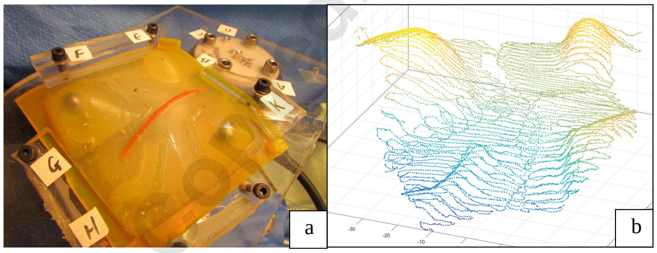

FIGURE 13. Force-controlled exploration using dVRK PSM: (a) is the silicone phantom organ, (b) is the actual collected point cloud from exploration.

4 DEFORMABLE MODEL REGISTRATION

Given the environment geometry based on force-controlled exploration data, this section presents the use of this data to update a pre-planned VF descriptor. In this paper, a 3D *target curve* is used to describe a VF representing a pre-planned ablation path.

4.1 Incorporating the Virtual Fixture Target Curve to an *a-priori* Model

The following is a description of how the pre-planned VF target curve was incorporated into an *a-priori* model. An STL file representing a non-deformed silicone phantom model was obtained from a CAD model using Creo Parametric™. This STL file will henceforth be called the *a-priori* model having a corresponding point cloud (P_a). The non-deformed silicone model was laser scanned using a Faro Arm Fusion® resulting in a point cloud model (P_{ls}). A VF curve (C_{dig}) denoting a mockup preoperative plan was also marked on the non-deformed silicone model and digitized using the Faro Arm. The laser scanned point cloud (P_{ls}) of the non-deformed silicone model and the the digitized VF curve points (C_{dig}) are shown in Fig. 14.

[2] `http://arma.vuse.vanderbilt.edu/images/stories/videos/long_jmr_mme2.mp4`

[3] `http://arma.vuse.vanderbilt.edu/images/stories/videos/long_jmr_mme3.mp4`

Journal of Mechanisms and Robotics. Received October 13, 2016;
Accepted manuscript posted January 11, 2017. doi:10.1115/1.4035684
Copyright (c) 2017 by ASME

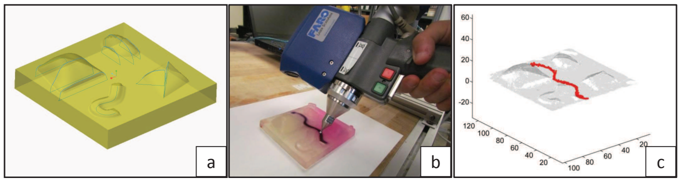

FIGURE 14. Creating an *a-priori* model of the silicone phantom a) *a-priori* STL model (P_A); b) digitizing the target curve (C_{dig}) using Faro Arm; c) laser scan (P_{ls}) and the digitized curve (C_{dig}) in red

A deformable registration method (denoted by *DReg*) based on Coherent Point Drift [21] was used. Given two point clouds or models X_1 and X_2, this method produces a deformable registration transformation $\mathrm{T}(\cdot) \leftarrow DReg(X_1, X_2)$, such that $\mathrm{T}(X_2) \approx X_1$. Using this approach, the target curve was registered to the *a-priori* model using the laser scan (P_{ls}), by following three steps:

i) The laser scanned point cloud of the non deformed phantom (P_{ls}) was registered to the *a-priori* STL point cloud (P_a) using *DReg*. This step resulted in a transformation T_1, such that $P_a = T_1(P_{ls})$.
ii) T_1 was used to transform the digitized target curve (C_{dig}), resulting in a registered digitized target curve (C_{dig_a}) in the frame of the *a-priori* model.
iii) A polynomial basis was used to fit a smooth target curve (C_a) to (C_{dig_a}). The curve C_a is represented by a high-density of points having constant arc-length spacing along the smooth curve. The root mean square (RMS) error of target curve fitting process was recorded.

Although in i) one could have used rigid-body point-cloud registration, it was easy to use *DReg* because it also covers the special case of rigid point cloud registration while dealing with the fact that the laser point cloud contains noisy data. The final output is the *a-priori* model M_a including P_a and C_a registered in the *a-priori* model frame.

4.2 Updating the Virtual Fixture Curve

The approach to registering and updating the surgical plan as represented by the target curve (VF curve) is depicted graphically in Fig. 15. The deformed silicone model was explored using the robot and a point cloud $\left(\tilde{P}_a\right)$ was obtained[4]. The same deformable registration method (*DReg*) was used between the *a-priori* model point cloud (P_a) and the robot exploration data of the deformed phantom model ($\tilde{P}_a$):

$$[T_2,\, L_{\mathrm{corr}}] \leftarrow DReg\left(\tilde{P}_a, P_a\right) \tag{3}$$

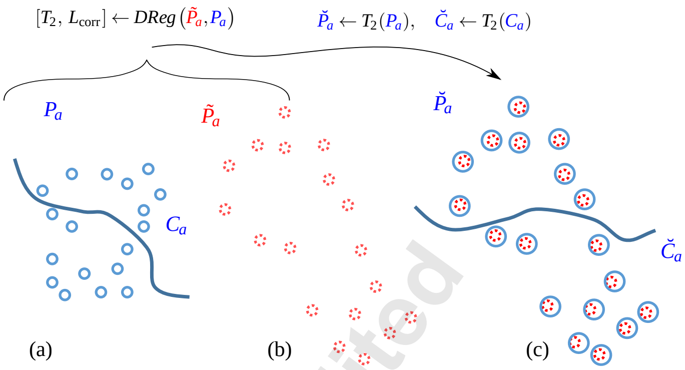
$[T_2,\, L_{\mathrm{corr}}] \leftarrow DReg\left(\tilde{P}_a, P_a\right)$
$\breve{P}_a \leftarrow T_2(P_a)$, $\breve{C}_a \leftarrow T_2(C_a)$
P_a
$\tilde{P}_a$
$\breve{P}_a$
C_a
$\breve{C}_a$
(a)
(b)
(c)

FIGURE 15. The process of updating the virtual fixture (VF) geometry: (a) the a-priori model (pre-operative model) with a VF curve, (b) the deformed environment obtained from exploration data, (c) using correspondence list to find the VF points in the exploration data set that match the curve from a-priori data set, (d) the transformed and registered VF geometry in the deformed environment

where T_2 is a deformable registration transformation and L_{corr} is a list of point indices relating points in P_a to their corresponding points in $\tilde{P}_a$. Applying T_2 to P_a and C_a results in the model point cloud and the VF curve points ($\breve{P}_a$) and ($\breve{C}_a$), respectively[5].

$$\breve{P}_a \leftarrow T_2(P_a), \qquad \breve{C}_a \leftarrow T_2(C_a) \tag{4}$$

Figure 16 illustrates the deformable registration process using a data set from based on dVRK PSM force-controlled exploration. The blue point cloud is the exploration data, the red point cloud is the updated *a-priori* model and the green curve is the updated VF curve. A sequence of the intermediate registered results from different iterations throughout the process are presented, from which one can see the converging of the updated model to the exploration data.

To evaluate the performance of the exploration and registration, the actual VF curve on the deformed environment is digitized and a fitted ground truth curve $\tilde{C}_{gt}$ is obtained. The **ground truth VF curve fitting residual error** ε_{gt} is defined as the least-squares curve fitting residual, which is calculated as the root mean squares (RMS) error between the digitized points and their corresponding closest points on $\tilde{C}_{gt}$. The **VF curve registration error** ε_o is captured between the updated VF curve $\breve{C}_a$ and the ground truth fitted curve $\tilde{C}_{gt}$ as the following:

$$\varepsilon_o \triangleq \sqrt{\frac{1}{N}\sum_{i=1}^{N}\|\breve{c}_{a_i}-\tilde{c}^*_{gt_i}\|^2},\ \tilde{c}^*_{gt_i} = \underset{n}{\operatorname{argmin}}\|\tilde{c}_{gt_n}-\breve{c}_{a_i}\| \tag{5}$$

where N designates the number of sample points along the deformably registered VF curve $\breve{C}_a$ and $\tilde{c}^*_{gt_i}$ is a point along the

[4] The wave accent (˜) denotes data obtained from the deformed phantom model using either force-controlled exploration or laser scanning

[5] The breve accent (˘) denotes the updated *a-priori* data.

Journal of Mechanisms and Robotics. Received October 13, 2016;
Accepted manuscript posted January 11, 2017. doi:10.1115/1.4035684
Copyright (c) 2017 by ASME

ground truth VF curve $\tilde{C}_{gt}$ that is closest to the i^{th} sample point along the deformably registered VF curve $\check{C}_a$.

4.3 Validation of VF Update and Discussion

The VF update strategy was validated using a custom Cartesian robot and the da Vinci research kit patient side manipulator (dVRK PSM). For both experiments we used two similar silicone phantom models made using the same mold, but with slight variations due to molding accuracy and placement/deformation of the phantom model when fixed to a deformed base as seen in Fig. 13. It was not possible to use the same exact phantom model due to the degradation of the first silicone phantom model during the period of porting our experiments to the dVRK PSM.

The Cartesian robot was used initially because it provides both high rigidity and positional accuracy during force-controlled exploration. The force-controlled robot exploration strategy was presented in section 3 and the data collection can be seen in Fig. 12 and in *Multimedia Extension II*[6]. The same strategy was repeated on another robot - dVRK PSM, to test feasibility of our approach on a clinically relevant setup and to test whether the compliance of the dVRK PSM would prohibit the deployment of our approach for exploration-based VF update. *Multimedia Extension III*[7] shows the force-controlled exploration using the dVRK PSM.

To determine whether the force-controlled exploration affects the registration due to local deformation caused by the exploration probe, we carried out two other experiments using non-contact laser scanning. These two laser scans of each phantom model were carried out to provide comparison baselines for both of the Cartesian and PSM force-controlled exploration of the deformed phantom models. Given the laser scanned model and the force-controlled exploration data, the steps described above in equations (3)-(4) were carried out. The resulting deformed VF curve $\check{C}_a$ based on the laser scan was then used to calculate the registration error according to equation (5).

Using the Faro Arm Fusion®, the ground truth VF curve $\tilde{C}_{gt}$ was digitized for phantom 1. This was possible because the Cartesian robot was calibrated and its geometry allows easy registration of its base frame to the base of the Faro arm. This was not the case for phantom model 2 used with the dVRK PSM since it is an un-calibrated robot presenting substantial difficulties in registering its base frame to the base frame of the Faro arm. We therefore used the dVRK PSM as a digitizer to obtain the ground truth VF curve $\tilde{C}_{gt}$.

Table 1 shows the comparison of the VF curve registration errors for both the Cartesian and the dVRK PSM (also see Fig. 17). Since we used two similar - yet different - phantom models we had to laser scan and digitize the VF curve on each phantom model. Therefore, the table is split into left and right two-column blocks associated with each phantom model. Phantom 1 was used with the Cartesian stage robot and Phantom 2 was used with the dVRK PSM. The right and left columns of each of the two-column blocks report the VF curve registration errors using non-contact laser scan and force-controlled exploration, respectively.

TABLE 1. Deformable registration results for the Cartesian robot and dVRK PSM. For each phantom model the first column shows VF update errors based on force-controlled exploration and the second column shows the same based on laser scanning

Error Types	RMS Errors for Different Registrations [mm]			
	Phantom #1		Phantom #2	
	Cartesian	Laser	PSM	Laser
Overall ε_o	3.393	3.069	3.386	1.295
Ground Truth ε_{gt}	0.998	0.998	1.711	0.560

The overall error ε_o for different experiments should be reviewed while considering the ground truth digitization curve fitting error ε_{gt}. For example in the PSM exploration case, ε_o has a value of 3.4 mm while ε_{gt} is 1.7 mm which contributes to the overall error. This ground truth fitting error is also consistent with the finding in [34] which shows that PSM has an approximate fiducial localization error of 1 mm. In the experiments associated with Phantom 1, the similar registration errors between the the Cartesian robot and the laser scan confirms that the effect of force-controlled exploration on the registration was negligible. However, comparing the PSM and its Laser baseline, we find that the errors when using the PSM were significantly larger compared to when using the laser scanner. Two possible sources of error that explain this phenomenon were noticable: 1) the dVRK PSM has lateral compliance of its long and slender arms and these deflections are not observable by the robot's encoders when used as a digitizer and also when carrying out the force-controlled exploration, 2) the dVRK PSM used a wrist which had some slack due to over-use. This slack also contributes to positional error of the robot's tip. Both of robots provide overall VF update errors smaller than 3.5 mm, which is acceptable for the required registration accuracy for many abdominal surgical applications [35, 36].

5 EVALUATION OF UPDATED VIRTUAL FIXTURES FOR MOCKUP ABLATION

In addition to testing the feasibility of VF update using force-controlled exploration, we tested the feasibility of using such updated VF in an assistive VF in which lateral deviation from the desired VF path are resisted by applying a corrective force at the impedance master of the dVRK. The motion of the slave robot along the local surface normal is regulated by a force controller law that maintains a constant ablation force along the

[6] http://arma.vuse.vanderbilt.edu/images/stories/videos/long_jmr_mme2.mp4

[7] http://arma.vuse.vanderbilt.edu/images/stories/videos/long_jmr_mme3.mp4

Journal of Mechanisms and Robotics. Received October 13, 2016;
Accepted manuscript posted January 11, 2017. doi:10.1115/1.4035684
Copyright (c) 2017 by ASME

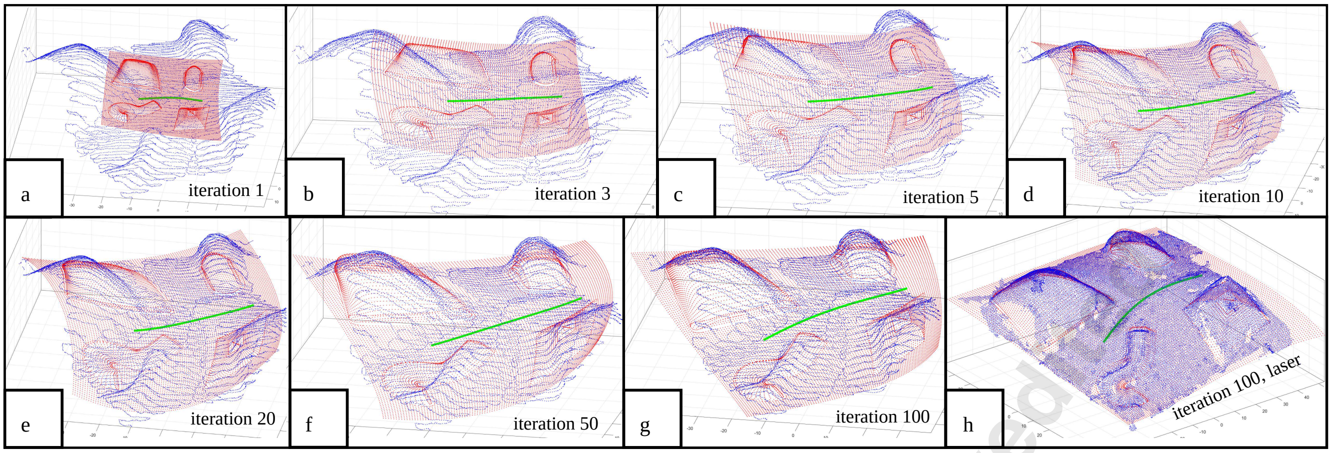

FIGURE 16. Deformable registration using exploration data from dVRK PSM robot and ground truth: (a)-(g) show iterations of deformable registration using PSM robot data where iteration numbers are $\{1, 3, 5, 10, 20, 50, 100\}$, (h) is the deformable registration result using laser scan data.

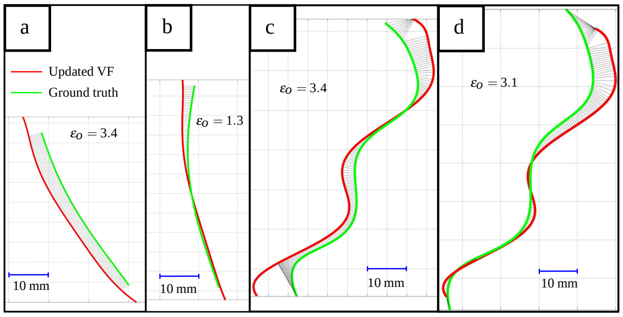

FIGURE 17. Errors between the updated VF curve and the digitized ground truth fitted curve: (a) PSM robot case, (b) PSM laser comparison, (c) Cartesian robot case, (d) Cartesian laser comparison.

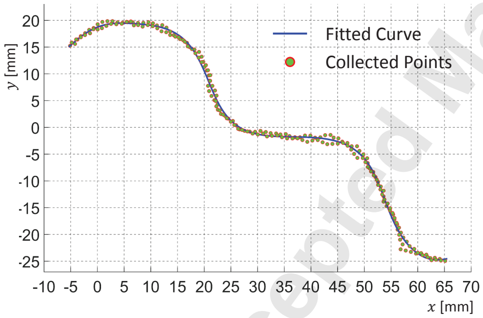

FIGURE 18. Ground truth digitization data points and the smooth curve fit

pre-planned ablation path.

The experimental validation was carried out using the Cartesian robot with a dVRK master. Three users participated the experiments: one user was experienced with the system and the other two were not. All users were given 30 minutes each to warm up and get used to telemanipulating the system with and without VF assistance. In both cases the hybrid force/motion controller was used with a force reference command of 0.7 N normal to the silicone phantom surface. Each user was instructed to follow a target curve back and forth twice while not paying attention to stopping exactly at the ends of the curve but rather trying to follow the curve the best they can with minimal time. Visualization was provided through an HD camera. The package *rosbag* was used to record the time and slave pose and slave command. Using the digitized data for the ground truth curve, we fitted a 10^{th} order Bernstein polynomial curve to have a smooth descriptor of the ground truth curve. This fitting process is shown in Fig. 18. For each user, the distance from the actual curve to the target curve was calculated after truncating the curve edges represented by the x coordinate $-5 < x < 65mm$. This effectively eliminated the edge effects since the users were not instructed to stop at the end of the curve. We also projected the data points onto a plane that best fits the target data and calculated the average RMS error along the sampled points of the curve.

TABLE 2. Trial completion time for each user (subject) with and without virtual fixture assistance.

	Trial Completion Time [sec]					
	Without virtual fixture			With virtual fixture		
Trial #	User 1	User 2	User 3	User 1	User 2	User 3
1	18.21	22.53	36.24	17.37	14.19	28.81
2	15.68	19.37	38.01	17.20	9.32	20.67
3	13.65	15.34	32.21	15.13	11.47	25.79
4	12.14	17.33	38.69	14.52	10.39	27.12
5	11.90	14.86	29.69	10.94	10.58	26.93
Average	14.32	17.89	34.97	15.03	11.19	25.86

The RMS tracking error and the completion time of each trial by every user is shown in Table 2 and Table 3 respectively.

Journal of Mechanisms and Robotics. Received October 13, 2016;
Accepted manuscript posted January 11, 2017. doi:10.1115/1.4035684
Copyright (c) 2017 by ASME

TABLE 3. RMS target curve tracking errors for each user (subject) with and without virtual fixture assistance.

	Average Trial RMS Error [mm]					
	Without virtual fixture			With virtual fixture		
Trial #	User 1	User 2	User 3	User 1	User 2	User 3
1	5.40	5.87	5.06	4.54	4.51	6.37
2	4.98	5.63	5.13	4.45	4.85	4.21
3	5.32	5.30	5.00	4.56	4.47	4.40
4	5.22	5.56	4.63	4.61	4.51	4.32
5	5.11	4.85	4.73	4.61	4.71	4.42
Average	5.21	5.44	4.91	4.55	4.61	4.75

A comparison of each user's performance with and without VF assistance is also presented in Fig. 19. These results show that all users benefited from reduced time for each trial and increased tracking accuracy when the curve-following VF law was implemented. A paired t-test between both RMS error groups resulted in a p score of 0.0031 thus rejecting the null hypothesis that both data sets come from the same distribution. Similarly, the results for time resulted with a p score of 0.0033.

6 CONCLUSIONS

The paper presented a framework for updating the geometry of a virtual fixture in a deformed environment by using information from a force-controlled exploration. A modular telemanipulation framework was presented within the context of model-mediated telemanipulation. The model update employed a deformable registration based on the coherent point drift algorithm, registering an *a-priori* model of the environment and an associated virtual fixture using the exploration data. The approach was successfully demonstrated using a custom Cartesian slave robot and a da Vinci Research Kit Patient Side Manipulator which were tele-manipulated using a da Vinci Research Kit Master Tool Manipulator. The results show that the assistive behaviors after the model update benefit the users in both speed and accuracy. We believe that this framework will benefit future surgical applications where force-controlled ablation and dissection along anatomical paths is required. Future work will include using stiffness information to drive the registration method and to update the model of the environment.

ACKNOWLEDGMENT

The research reported in this paper has been supported in part by NSF grants #IIS-1327566, #IIS-1327657 and #IIS-1208540 through the National Robotics Initiative and in part by institutional funds. We are thankful to Dr. Jason Pile who provided expertise and assistance in evaluating the user study results and Dr. Haoran Yu for his help in preparing figures of experimental setup.

REFERENCES

[1] Sotiras, A., Davatzikos, C., and Paragios, N., 2013. "Deformable medical image registration: a survey.". *IEEE transactions on medical imaging,* **32**(7), July, pp. 1153–90.

[2] Mirota, D. J., Ishii, M., and Hager, G. D., 2011. "Vision-based navigation in image-guided interventions.". *Annual review of biomedical engineering,* **13**, Aug., pp. 297–319.

[3] Cash, D. M., Sinha, T. K., Chapman, W. C., Terawaki, H., Dawant, B. M., Galloway, R. L., and Miga, M. I., 2003. "Incorporation of a laser range scanner into image-guided liver surgery: Surface acquisition, registration, and tracking". *Medical Physics,* **30**(7), pp. 1671–1682.

[4] Hayashibe, M., 2006. "Laser-scan endoscope system for intraoperative geometry acquisition and surgical robot safety management". *Medical Image Analysis,* **10**, pp. 509–519.

[5] Lathrop, R. A., Rucker, D. C., and Webster III, R. J., 2010. "Guidance of a steerable cannula robot in soft tissue using preoperative imaging and conoscopic surface contour sensing". IEEE, pp. 5601–5606.

[6] Xia, T., Leonard, S., Kandaswamy, I., Blank, A., Whitcomb, L. L., and Kazanzides, P., 2013. "Model-based telerobotic control with virtual fixtures for satellite servicing tasks". In 2013 IEEE International Conference on Robotics and Automation, IEEE, pp. 1479–1484.

[7] Mitra, P., and Niemeyer, G., 2008. "Model-mediated Telemanipulation". *The International Journal of Robotics Research,* **27**(2), Feb., pp. 253–262.

[8] Kazanzides, P., Chen, Z., Deguet, A., Fischer, G. S., Taylor, R. H., and DiMaio, S., 2014. "An open-source research kit for the da Vinci® surgical robot". In IEEE Intl. Conf. on Robotics and Auto. (ICRA).

[9] Galea, A., and Howe, R., 2002. "Tissue stiffness from tactile imaging". In Engineering in Medicine and Biology, 2002. 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society EMBS/BMES Conference, 2002. Proceedings of the Second Joint, Vol. 2, IEEE, pp. 935–936.

[10] Kesner, S. B., and Howe, R. D., 2011. "Discriminating tissue stiffness with a haptic catheter: Feeling the inside of the beating heart". In World Haptics Conference (WHC), 2011 IEEE, IEEE, pp. 13–18.

[11] Egorov, V., Van Raalte, H., and Sarvazyan, A. P., 2010. "Vaginal tactile imaging". *Biomedical Engineering, IEEE Transactions on,* **57**(7), pp. 1736–1744.

[12] Liu, H., Noonan, D. P., Challacombe, B. J., Dasgupta, P., Seneviratne, L. D., and Althoefer, K., 2010. "Rolling mechanical imaging for tissue abnormality localization during minimally invasive surgery". *Biomedical Engineering, IEEE Transactions on,* **57**(2), pp. 404–414.

[13] Xu, K., and Simaan, N., 2010. "Intrinsic wrench estimation and its performance index for multisegment continuum robots". *Robotics, IEEE Transactions on,* **26**(3), pp. 555–561.

[14] Nichols, K., Okamura, A. M., et al., 2013. "Autonomous robotic palpation: Machine learning techniques to identify hard inclusions in soft tissues". In Robotics and Automation (ICRA), 2013 IEEE International Conference on, IEEE, pp. 4384–4389.

[15] Bajo, A., and Simaan, N., 2016. "Hybrid motion/force control of multi-backbone continuum robots". *Int. J. Rob. Res.,* **35**(4), Apr., pp. 422–434.

[16] Goldman, R. E., Bajo, A., and Simaan, N., 2013. "Algorithms for autonomous exploration and estimation in compliant environments". *Robotica,* **31**(1), Mar., pp. 71–88.

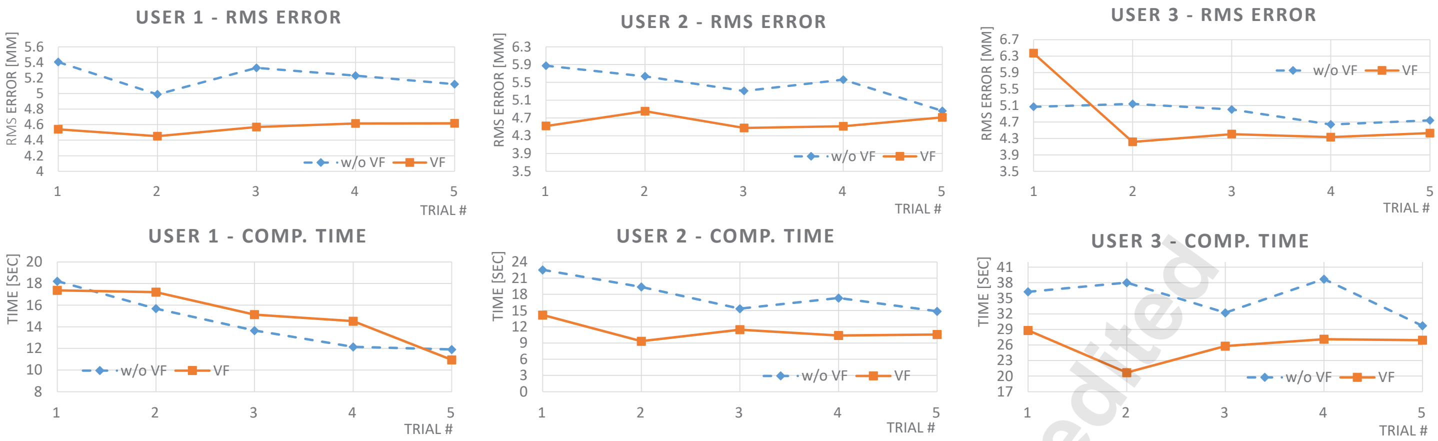

FIGURE 19. Comparisons of the users' performance with and without virtual fixture on target curve tracing RMS errors and completion time.

[17] Sanan, S., Tully, S., Bajo, A., Simaan, N., and Choset, H., 2014. "Simultaneous compliance and registration estimation for robotic surgery". In Proceedings of Robotics: Science and Systems.

[18] Srivatsan, R. A., Ayvali, E., Wang, L., Roy, R., Simaan, N., and Choset, H., 2016. "Complementary Model Update: A Method for Simultaneous Registration and Stiffness Mapping in Flexible Environments". In Proceedings 2016 IEEE International Conference on Robotics and Automation, IEEE (accepted).

[19] Chalasani, P., Wang, L., Roy, R., Simaan, N., Taylor, R. H., and Kobilarov, M., 2016. "Concurrent Nonparametric Estimation of Organ Geometry and Tissue Stiffness Using Continuous Adaptive Palpation". In Proceedings 2016 IEEE International Conference on Robotics and Automation, IEEE (accepted).

[20] Ayvali, E., Srivatsan, R. A., Wang, L., Roy, R., Simaan, N., and Choset, H., 2016. "Using Bayesian Optimization to Guide Probing of a Flexible Environment for Simultaneous Registration and Stiffness Mapping". In Proceedings 2016 IEEE International Conference on Robotics and Automation, IEEE (accepted).

[21] Myronenko, A., and Song, X., 2010. "Point set registration: coherent point drift.". *IEEE transactions on pattern analysis and machine intelligence*, **32**(12), Dec., pp. 2262–75.

[22] Billings, S. D., Boctor, E. M., and Taylor, R. H., 2015. "Iterative most-likely point registration (imlp): a robust algorithm for computing optimal shape alignment". *PloS one*, **10**(3).

[23] Wang, L., Chen, Z., Chalasani, P., Pile, J., Kazanzides, P., Taylor, R. H., and Simaan, N., 2016. "Updating virtual fixture from exploration data in force-controlled model-based telemanipulation". In ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, p. In press.

[24] Jung, M. Y., Deguet, A., and Kazanzides, P., 2010. "A component-based architecture for flexible integration of robotic systems". In Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, IEEE, pp. 6107–6112.

[25] Chen, Z., Deguet, A., Taylor, R., DiMaio, S., Fischer, G., and Kazanzides, P., 2013. "An open-source hardware and software platform for telesurgical robot research". In MICCAI Workshop on Systems and Arch. for Computer Assisted Interventions.

[26] Khatib, O., 1987. "A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation". *IEEE Journal of Robotics and Automation*, **RA-3**(1), Feb., pp. 43–53.

[27] Featherstone, R., Thiebaut, S., and Khatib, O., 1999. "A general contact model for dynamically-decoupled force/motion control". In Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C), Vol. 4, IEEE, pp. 3281–3286.

[28] Whitney, D. E., 1969. "Resolved motion rate control of manipulators and human prostheses.". *IEEE Transactions on man-machine systems*.

[29] Kapoor, A., Li, M., and Taylor, R., 2006. "Constrained control for surgical assistant robots". In Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on, pp. 231–236.

[30] Yoshikawa, T., and Sudou, A., 1993. "Dynamic hybrid position/force control of robot manipulators-on-line estimation of unknown constraint". *IEEE Transactions on Robotics and Automation*, **9**(2), pp. 220–226.

[31] Namvar, M., and Aghili, F., 2005. "Adaptive force-motion control of coordinated robots interacting with geometrically unknown environments". *IEEE Transactions on Robotics*, **21**(4), pp. 678–694.

[32] Yin, Y., Hu, H., and Xia, Y., 2004. "Active tracking of unknown surface using force sensing and control technique for robot". *Sensors and Actuators A: Physical*, **112**(2), pp. 313–319.

[33] Karayiannidis, Y., and Doulgeri, Z., 2009. "Adaptive control of robot contact tasks with on-line learning of planar surfaces". *Automatica*, **45**(10), pp. 2374–2382.

[34] Kwartowitz, D. M., Herrell, S. D., and Galloway, R. L., 2006. "Toward image-guided robotic surgery: determining intrinsic accuracy of the da vinci robot". *International Journal of Computer Assisted Radiology and Surgery*, **1**(3), pp. 157–165.

[35] Linte, C. A., Moore, J., and Peters, T. M., 2010. "How accurate is accurate enough? a brief overview on accuracy considerations in image-guided cardiac interventions". In Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE, IEEE, pp. 2313–2316.

[36] Krücker, J., Xu, S., Glossop, N., Viswanathan, A., Borgert, J., Schulz, H., and Wood, B. J., 2007. "Electromagnetic tracking for thermal ablation and biopsy guidance: clinical evaluation of spatial accuracy". *Journal of Vascular and Interventional Radiology*, **18**(9), pp. 1141–1150.

Journal of Mechanisms and Robotics. Received October 13, 2016;
Accepted manuscript posted January 11, 2017. doi:10.1115/1.4035684

List of Figures

Journal of Mechanisms and Robotics. Received October 13, 2016;
Accepted manuscript posted January 11, 2017. doi:10.1115/1.4035684
Copyright (c) 2017 by ASME

List of Tables