**Proceedings of the ASME 2016 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
IDETC 2016
August 21-24, 2016, Charlotte,NC, USA**

[DRAFT - IDETC2016-59305]
UPDATING VIRTUAL FIXTURES FROM EXPLORATION DATA IN FORCE-CONTROLLED MODEL-BASED TELEMANIPULATION

Long Wang
ARMA Laboratory[*]
Department of Mechanical Engineering
Vanderbilt University
Nashville, Tennessee 37212
long.wang@vanderbilt.edu

Zihan Chen
Preetham Chalasani
LCSR[†]
Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218
{zihan.chen, pchalas1}@jhu.edu

Jason Pile[*]
ARMA Laboratory [*]
Department of Mechanical Engineering
Vanderbilt University
Nashville, Tennessee 37212
jason.pile@gmail.com

Peter Kazanzides
Russell H. Taylor
LCSR [†]
Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218
{pkaz, rht}@jhu.edu

Nabil Simaan[‡]
ARMA Laboratory [*]
Department of Mechanical Engineering
Vanderbilt University
Nashville, Tennessee 37212
nabil.simaan@vanderbilt.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 (virtual fixture) 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 and a Cartesian stage robot. Experiments demonstrate that the updated virtual fixture and the updated model allow the users to improve their path following performance and to shorten their completion time when the updated path following virtual fixture 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

[*] ARMA, Advanced Robotics and Mechanism Applications. Jason Pile graduated in June 2015 and is currently at Intuitive Surgical Inc., Sunnivale, CA
[†] LCSR, Laboratory for Computational Sensing and Robotics
[‡] Corresponding author

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, 4] 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 [5]. 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. [6, 7]). 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 [8–10]. Rolling mechanical imaging was obtained using a force-sensitive probe in [11] and probing motion (indenting tissue in the depth direction) was investigated in [12, 13]. Dynamic excitation of tissue was applied to estimate impedance parameter in [14]. This work differs from these previous works by attempting to use geometric information obtained through force-controlled exploration to register and correct a pre-operative *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 [15] 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 [16] 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 [17]. In [18], 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 [19, 20]. 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.

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 [19]. 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-based telemanipulation. It also provides intelligent control behaviors on the slave side to support force-controlled exploration and telemanipulation.

2 TELEMANIPULATION FRAMEWORK

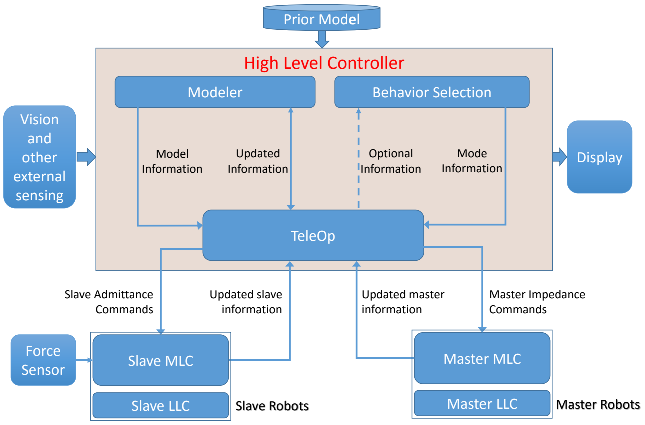


FIGURE 1. System Architecture

Our telemanipulation framework, as shown in Fig. 1, is based on the JHU "Surgical Assistant Workstation (SAW)" software environment [21] and the da Vinci Research Toolkit (DVRK) [7, 22]. This environment supports multiple telemanipulation hardware and software components in a mix-and-match fashion. For our current research, we use da Vinci DVRK "master" manipulators and slave manipulators choosing from either DVRK patient-side manipulators (at JHU) or a custom cartesian robot (at Vanderbilt, 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, see Fig. 2. The results reported in Section 5 were obtained with the Vanderbilt slave hardware. 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.

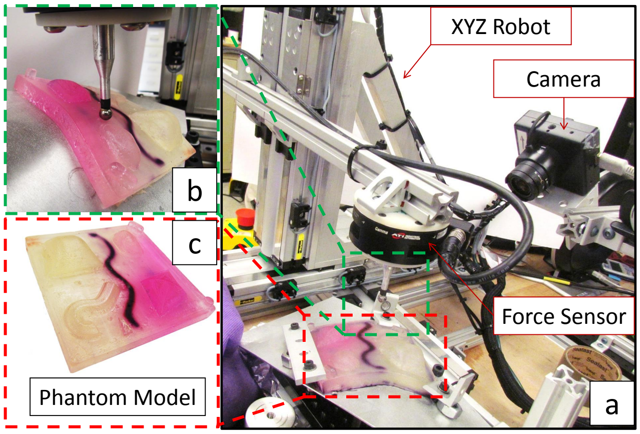


FIGURE 2. Vanderbilt Cartesian Slave Robot System: a) Experiment setup; b) Ball Probe Finger ATI Force Torque Sensor; c) A phantom model used in experiment

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 regulated 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 [14] we demonstrated force-controlled estimation of flexible environment constraints and impedances. In [15,23] 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 Admittance Controller

As a slight variation of the *Slave Controller* process above, the Vanderbilt Cartesian slave robot system is different in that: 1) its MLC and LLC are integrated in the same processor and developed in Matlab Simulink® Real-Time environment; 2) while accepting admittance commands and virtual fixture specifications it is also compatible with position and desired force commands from *TeleOp*. The *slave MLC* is implemented as an admittance type controller, and the *slave LLC* is in hybrid motion/force control type, shown in Fig. 3. This hybrid motion/force with dynamic compensation controller structure is motivated by the works of Khatib [24] and Featherstone [25].

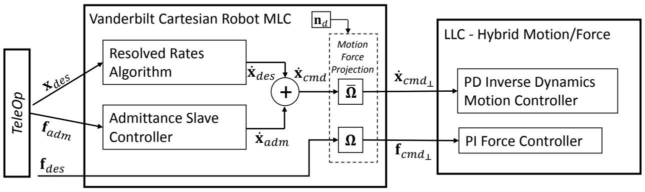


FIGURE 3. Vanderbilt Slave MLC-LLC Controller

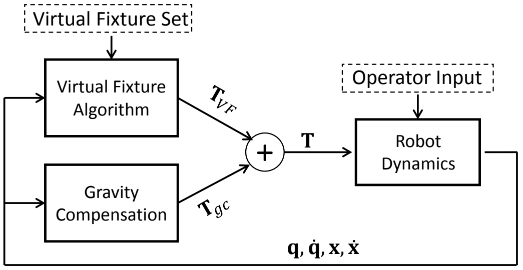


FIGURE 4. Master MLC Impedance Type Controller: $\mathbf{q}$ - joint position; $\dot{\mathbf{q}}$ - joint velocity; $\mathbf{x}$ - cartesian position; $\dot{\mathbf{x}}$ - cartesian velocity; T - total joint torque applied to robot; $\mathrm{T_{VF}}$ - joint torque from virtual fixture controller $\mathrm{T_{gc}}$ - joint torque from gravity compensation

The MLC will accept 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 [26], given the desired position $\mathbf{x}_{des}$ and the current slave position. Another velocity command $\dot{\mathbf{x}}_{adm}$ will be generated given an admittance force command and admittance gains. The added velocity command $\mathbf{x}_{cmd}$ and desired force will be decomposed by the *Motion Force Projection* block. The projected velocity and force command $\mathbf{x}_{cmd_\perp}$ and $\mathbf{f}_{cmd_\perp}$ are then sent to the low level controller. The projection matrices in Fig. 3 are given as:

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

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. 4, 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.

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 $(e_i \leq 0)$ 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
```

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 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 [27], 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. 5, 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.

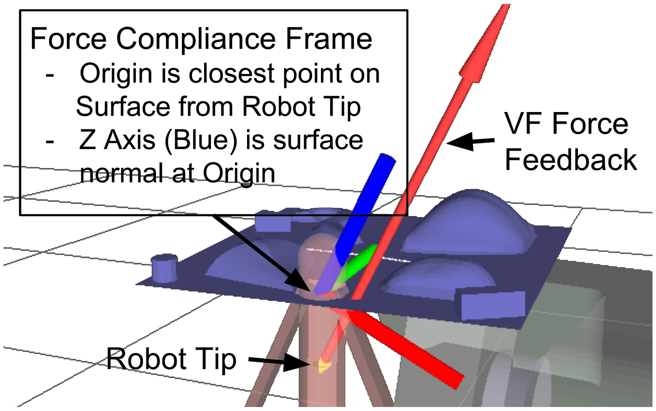

Force Compliance Frame
- Origin is closest point on Surface from Robot Tip
- Z Axis (Blue) is surface normal at Origin
VF Force Feedback
Robot Tip

FIGURE 5. Surface following frame with master robot tip and force feedback

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. 6.

3 FORCE CONTROLLED EXPLORATION

Using hybrid force/motion controller of the cartesian slave robot described in section 2 we can achieve a force-controlled exploration of the environment, i.e. a *surface following* function. Our previous work [14] used a similar *surface following* function where the robot was controlled given a constant predefined force desired direction (i.e. $\mathbf{n}_d$ in Fig. 3 is constant and specified). In this work, the force desired direction is updated based on current estimation of the environment. 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. 7. The hybrid force/motion slave controller accepts the

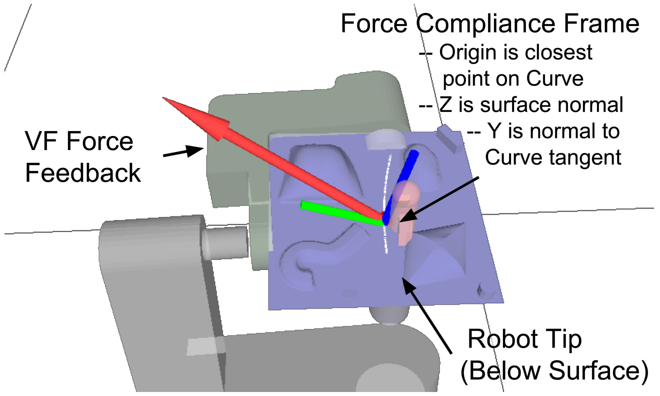

Force Compliance Frame
-- Origin is closest point on Curve
-- Z is surface normal
-- Y is normal to Curve tangent
VF Force Feedback
Robot Tip (Below Surface)

FIGURE 6. Curve following frame with master robot tip and force feedback

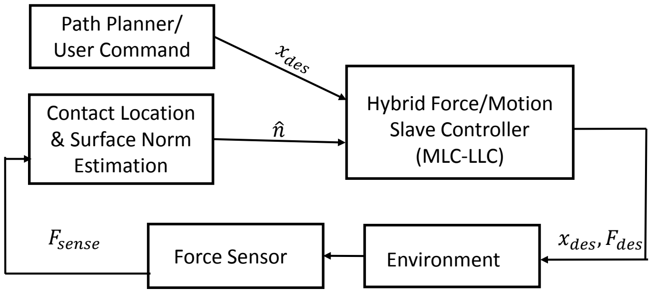

Path Planner/ User Command
x_{des}
Contact Location & Surface Norm Estimation
$\hat{n}$
Hybrid Force/Motion Slave Controller (MLC-LLC)
F_{sense}
Force Sensor
Environment
x_{des}, F_{des}

FIGURE 7. Force-controlled Exploration Strategy

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. 8. 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.

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. 8, 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*.

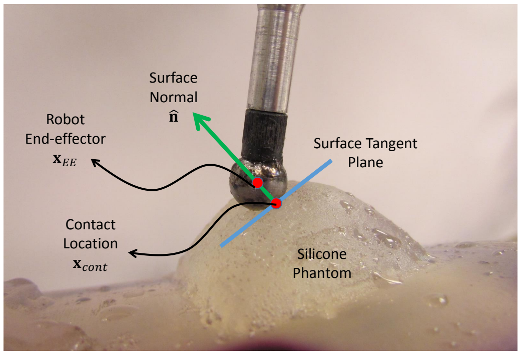


FIGURE 8. Contact Location and Surface Norm Estimation

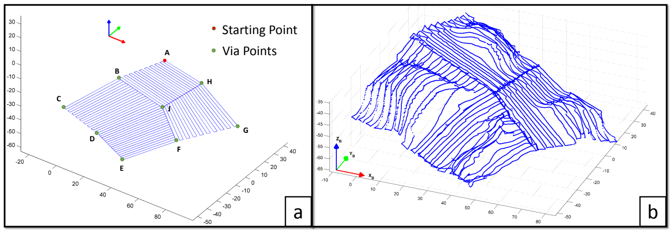


FIGURE 9. Force-controlled Robot Exploration

a) is the planned scan pattern; b) is the actual scanned point cloud

3.2 Surface Exploration

Fig. 9a 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. 9a. 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. 9a $N_p = 148$. And $\mathbf{P}_{XY}$ combined with the current Z coordinate of the robot in real-time were sent as command positions to the slave hybrid motion/force MLC.

The 2D scan pattern $\mathbf{P}_{XY}$ was executed at a constant velocity of 1 mm/sec. Currently, 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 can be used to calibrate the loading mass parameters. Hence the compensation of the dynamic effect from the loading mass can be provided in real-time, enabling a much faster scanning capability. Fig. 9b shows the actual estimated contact locations during the force-controlled exploration.

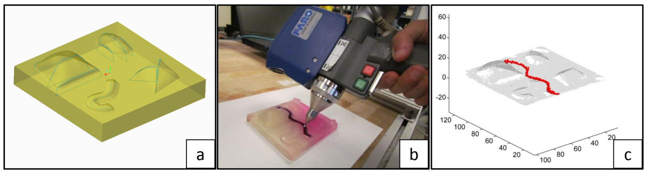


FIGURE 10. Creating an *a-priori* model of the silicone phantom a) *a-priori* STL model (P_A); b) digitizing the target curve (C_{dig}) using Faro Arm; c) laser scan (P_{ls}) and the digitized curve (C_{dig}) in red

4 DEFORMABLE MODEL REGISTRATION

Having explained our method to obtain the environment geometry data using a robot, this section focuses on the use of that data in updating a pre-planned virtual fixture descriptor based on force-controlled exploration data. In this paper, a 3D target curve is used to describe a virtual fixture 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 virtual fixture target curve was incorporated to an *a-priori* model. An STL file representing the geometry of a non-deformed silicone phantom model was obtained from a Creo CAD model. We will henceforth refer to this STL file as 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 digitized target curve (C_{dig}) denoting a mockup pre-operative plan was also marked on the non-deformed silicone model and obtained using the Faro Arm. The laser scanned point cloud (P_{ls}) of the non-deformed silicone model and the the digitized target curve points (C_{dig}) are shown in Fig. 10.

A deformable registration method (denoted by *DReg*) based on Coherent Point Drift [19] 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* STL 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 thin spline was fitted to (C_{dig_a}) to result in a smooth target curve (C_a). The root mean square (RMS) error of target curve fitting process was 0.73 mm.

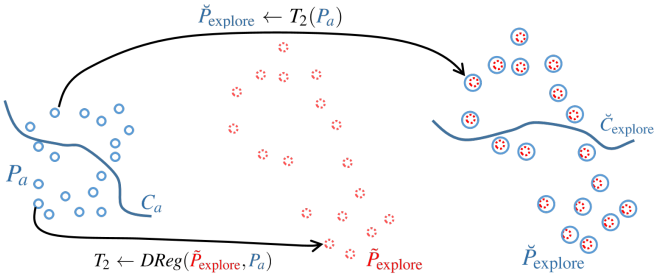

$\check{P}_{\text{explore}} \leftarrow T_2(P_a)$
$\check{C}_{\text{explore}}$
P_a
C_a
$T_2 \leftarrow DReg(\tilde{P}_{\text{explore}}, P_a)$
$\tilde{P}_{\text{explore}}$
$\check{P}_{\text{explore}}$

FIGURE 11. Illustration of the process for updating the virtual fixture curve

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 Target Curve

The approach to registering and updating the surgical plan (target curve) virtual fixture curve is depicted graphically in Fig. 11. The deformed silicone model was explored using the robot and a point cloud $\left(\tilde{P}_{\text{explore}}\right)$ was obtained[1]. The same *DReg* (deformable registration) method was used between the *a-priori* model point cloud (P_a) and the robot exploration data of the deformed phantom model ($\tilde{P}_{\text{explore}}$) as in Eq. 2. This registration resulted in a transformation T_2. The transformation T_2 was then used to deformably transform the *a-priori* model (P_a) to match $\left(\tilde{P}_{\text{explore}}\right)$. This resulted in ($\check{P}_{\text{explore}}$) which denotes the *a-priori* model point data represented in the deformed model frame. Similarly, the updated virtual fixture curve was transformed using T_2 to obtain ($\check{C}_{\text{explore}}$).

$$T_2 \leftarrow DReg(\tilde{P}_{\text{explore}}, P_a) \tag{2}$$

$$\check{P}_{\text{explore}} \leftarrow T_2(P_a) \tag{3}$$

$$\check{C}_{\text{explore}} \leftarrow T_2(C_a) \tag{4}$$

Both $\check{P}_{\text{explore}}$ and $\check{C}_{\text{explore}}$ were computed using the robot exploration data. The registration position error was calculated by comparing the transformed point cloud ($\check{P}_{\text{explore}}$) and the robot exploration point cloud ($\tilde{P}_a$), i.e. $\|\check{P}_{\text{explore}} - \tilde{P}_a\|$. The RMS of this registration error was approximately 0.4255 mm.

4.3 Laser vs. Force-Controlled Exploration

We designed another experiment to determine whether the force-controlled exploration process affected the registration due

[1] The wave accent (˜) will be used to denote data obtained from the deformed model

to local deformation caused by the exploration probe of the robot. During this experiment, a laser scanner mounted on the Faro-Arm was used to generate a contact-less exploration point cloud ($\tilde{P}_{\text{explore}}$). We then used this point cloud data and repeated the steps described above in equations (2)-(4). The registration error of the same method using laser scanned data of the deformed model was captured between the registered point cloud ($\breve{P}_{\text{explore}}$) and the laser scanned data ($\tilde{P}_{\text{explore}}$). The RMS registration error was 0.3543 mm which is close to the value of 0.4255 mm obtained using the force-controlled exploration data following the scan path in Fig. 9. This confirms that the effect of the shape scanning using robot force-controlled exploration on the registration was negligible. Both the laser-based registration and the force-controlled scan registration errors were under 0.5 mm, which is well above the required registration accuracy for most clinical applications [28–32].

To evaluate the accuracy of the updated virtual fixture, we compared $\breve{C}_{\text{explore}}$ resulting from the laser-scanner registration with $\breve{C}_{\text{explore}}$ resulting from the force-controlled exploration. This comparison resulted in approximately 2.8210 mm RMS error.

5 EXPERIMENTAL VALIDATION

The experimental validation was carried out using the robot of Fig. 2 with a DVRK master. Three users were given 10 minutes each to warm up and get used to telemanipulate the system with and without virtual fixture 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. 12. 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.

The results shown in Fig. 13 and Fig. 14 show that all users benefited from reduced time for each trial and increased tracking accuracy when the curve-following virtual fixture 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.

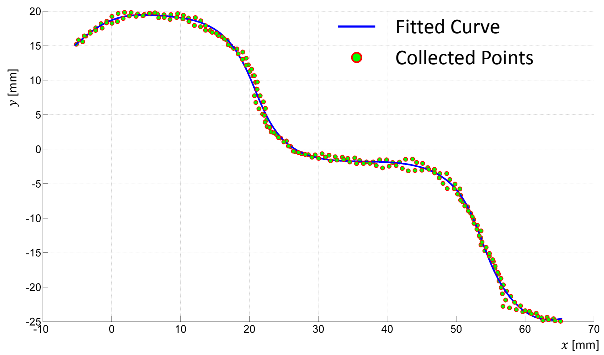
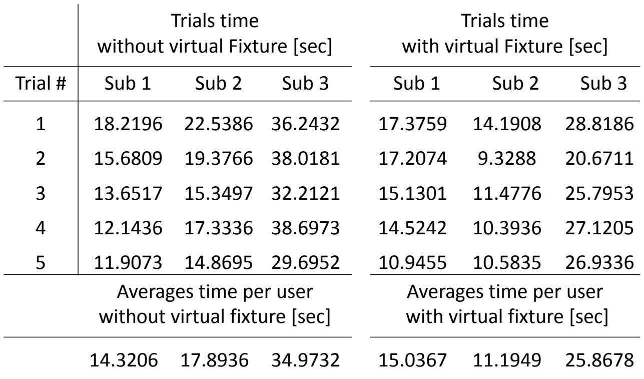
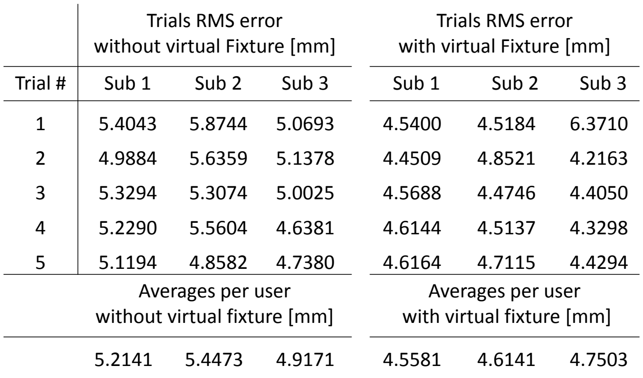


FIGURE 12. Ground truth digitization data points and the smooth curve fit



	Trials time without virtual Fixture [sec]			Trials time with virtual Fixture [sec]		
Trial #	Sub 1	Sub 2	Sub 3	Sub 1	Sub 2	Sub 3
1	18.2196	22.5386	36.2432	17.3759	14.1908	28.8186
2	15.6809	19.3766	38.0181	17.2074	9.3288	20.6711
3	13.6517	15.3497	32.2121	15.1301	11.4776	25.7953
4	12.1436	17.3336	38.6973	14.5242	10.3936	27.1205
5	11.9073	14.8695	29.6952	10.9455	10.5835	26.9336
	Averages time per user without virtual fixture [sec]			Averages time per user with virtual fixture [sec]		
	14.3206	17.8936	34.9732	15.0367	11.1949	25.8678

FIGURE 13. RMS trial time for each user with and without virtual fixture assistance. Each column represents the data per a user



	Trials RMS error without virtual Fixture [mm]			Trials RMS error with virtual Fixture [mm]		
Trial #	Sub 1	Sub 2	Sub 3	Sub 1	Sub 2	Sub 3
1	5.4043	5.8744	5.0693	4.5400	4.5184	6.3710
2	4.9884	5.6359	5.1378	4.4509	4.8521	4.2163
3	5.3294	5.3074	5.0025	4.5688	4.4746	4.4050
4	5.2290	5.5604	4.6381	4.6144	4.5137	4.3298
5	5.1194	4.8582	4.7380	4.6164	4.7115	4.4294
	Averages per user without virtual fixture [mm]			Averages per user with virtual fixture [mm]		
	5.2141	5.4473	4.9171	4.5581	4.6141	4.7503

FIGURE 14. RMS target curve tracking errors for each user with and without virtual fixture assistance. Each column represents the data per a user

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 Cartesian slave robot and a da Vinci research kit master device. 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 by NSF grants #IIS-1327566, #IIS-1327657 and #IIS-1208540 through the National Robotics Initiative.

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

[4] 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.

[5] 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.

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

[7] 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).

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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).

[17] 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).

[18] 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).

[19] 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.

[20] 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).

[21] 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.

[22] 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.

[23] Tully, S., Bajo, A., Kantor, G., Choset, H., and Simaan, N., 2012. "Constrained filtering with contact detection data for the localization and registration of continuum robots in flexible environments". In 2012 IEEE International Conference on Robotics and Automation, IEEE, pp. 3388–3394.

[24] 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.

[25] 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.

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

[27] 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.

[28] Poon, R. T.-P., Fan, S.-T., Ng, I. O.-L., and Wong, J., 2000. "Significance of resection margin in hepatectomy for hepatocellular carcinoma: a critical reappraisal". *Annals of surgery,* **231**(4), pp. 544–551.

[29] Vandeweyer, D., Neo, E. L., Chen, J. W., Maddern, G. J., Wilson, T. G., and Padbury, R. T., 2009. "Influence of resection margin on survival in hepatic resections for colorectal liver metastases". *HPB,* **11**(6), pp. 499–504.

[30] Wiles, A. D., Moore, J., Linte, C. A., Wedlake, C., Ahmad, A., and Peters, T. M., 2008. "Object identification accuracy under ultrasound enhanced virtual reality for minimally invasive cardiac surgery". In Medical Imaging, International Society for Optics and Photonics, pp. 69180E–69180E.

[31] 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.

[32] Horiguchi, Y., Sekoguchi, B., Imai, H., Suzuki, T., Kubo, H., Itoh, H., and Itoh, M., 1994. "Treatment of choice for unresectable small liver cancer: Percutaneous ethanol injection therapy or transarterial chemoembolization therapy". *Cancer chemotherapy and pharmacology,* **33**(1), pp. S111–S114.