Advanced Telerobotic Underwater Manipulation Using Virtual Fixtures and Haptic Rendering

Fredrik Rydén, Electrical Engineering Dept., University of Washington
Seattle, WA 98195, Email: ryden@uw.edu
Andrew Stewart, Applied Physics Laboratory, University of Washington
Seattle, WA 98105, Email: andy@apl.uw.edu
Howard Jay Chizeck, Electrical Engineering Dept., University of Washington
Seattle, WA 98195, Email: chizeck@uw.edu

***Abstract*—As the variety and complexity of tasks carried about by telerobotic systems underwater increases, so does the demand for technology that increases efficiency and safety of operations. We present a novel approach to subsea manipulation that leverages the natural perceptive capability of human pilots. We present a method and framework for generation and rendering of six degree-of-freedom haptic virtual fixtures based on streaming point cloud data captured by an RGB-D sensor. Sensor data is captured at 30 Hz and virtual fixtures are constructed in real-time in parallel on a GPU. The method is applied to two typical underwater telerobotic valve operations by implementing a forbidden-region virtual fixture and a guidance virtual fixture. During operation, real-time haptic feedback is sent to the operator at 1000 Hz. This haptic feedback allows pilots of remotely-operated vehicles to receive sense of touch feedback through the use of non-contact sensors.**

I. Introduction

When tasks must be carried out in the deep ocean or hazardous marine environments, teleoperated robotic systems are necessary. In most applications a pilot resides at the surface while a remotely-operated vehicle (ROV) carries out functions below. Munitions remediation, oil and gas operations, studies of hydrothermal vent colonies, and construction of ocean observatories are a subset of underwater applications that require teleoperation. In particular, manipulation is often necessary for operating tools, opening valves, mating cables, or simply moving objects from one place to another. Subsea manipulation is a challenging and expensive endeavor that relies on expert pilots with substantial experience. The cost of operating ships and robotic systems often drives crews to operate 24 hours/day and underscores the importance of avoiding mistakes. Advanced tools that lower the burden on pilots, increasing safety, reliability and efficiency of operations, has potential to dramatically reduce cost and increase feasibility of complex tasks performed underwater.

Perception is paramount to efficiently and safely carrying out complex tasks remotely. We make use of *virtual fixtures* to provide haptic feedback to a remote operator of a robotic device underwater. Haptic virtual fixtures are computer-generated constraints that limit the motion of a robot in space. Violation of these constraints is opposed by a resistive force, communicated to the pilot by a haptic device (input device that conveys reaction forces to the user). *Forbidden Regions* act as *force fields* that push back on the end effector to avoid constraints, allowing for sensitive objects to be protected. If the pilot commands the end effector into a forbidden region, a resistive force is rendered – preventing violation of the constraint. *Guidance Virtual Fixtures* help the pilot move toward a desired object to be manipulated.

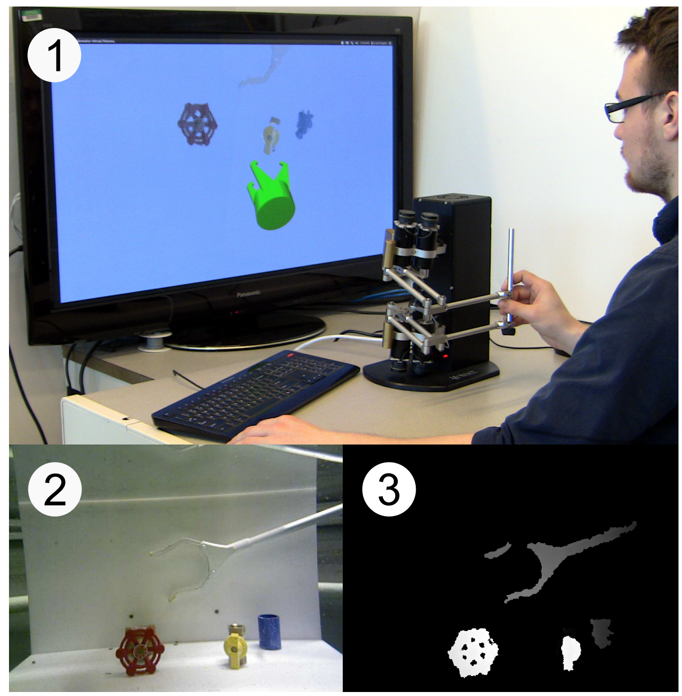

Fig. 1. Interaction with haptic virtual fixtures generated from underwater sensor data. The operator interacts with the virtual environment using a W5D haptic device (Entact Robotics) (1). RGB data of the underwater scene (2). Depth data of the same underwater scene (3).

In this work we demonstrate our ability to generate virtual fixtures in an underwater environment from non-contact (camera) sensors to provide haptic feedback to a remote pilot of a subsea manipulator. We focus on the application of remotely opening and closing valves underwater. This task is representative of operations that would be carried out by ROVs on oil rigs or in subsea infrastructure maintenance. It may be necessary to service one valve while not disturbing an adjacent valve. When valves are close in proximity this presents a challenge to the pilot. By using both guidance fixtures and

978-0-933957-40-4 ©2013 MTS

forbidden regions we are able to assist the pilot in servicing a valve while avoiding undesired contact or collisions with other objects in the environment.

Our approach makes use of a 3D camera (or sonar) to create a model of the environment. The haptic rendering algorithm runs in the virtual environment to compute reaction forces which are fed to the pilot. This leverages the natural perceptive capability of pilots and simplifies remote manipulation tasks.

We note that many ROVs have been equipped with tools for haptic feedback, but suffer from limitations. The standard approach to providing haptic feedback requires contact forces to be measured *after* contact. For subsea applications, latency and a subsequent instability arise driving many ROV pilots to switch off the haptic feedback function. Even when functioning properly, a significant burden is placed on pilots as they must predict points of contact by inferring three-dimensional structure through a series of two-dimensional displays. With our approach, the pilot receives feedback before contact is made. Further, virtual fixtures can be used to assist a pilot in and around tight corners, or toward targets (i.e. handles, knobs, interfaces), as well as away from sensitive objects.

The contributions of this paper include:

- A method for rendering of six degree-of-freedom (6-DOF) forbidden-region haptic and visual virtual fixtures from streaming point cloud data.
- A method for rendering of 6-DOF guidance virtual fixtures generated from streaming point cloud data, in combination with forbidden region virtual fixtures.
- A demonstration of virtual fixture rendering for a relevant underwater tool positioning task.

II. Related work

This paper builds upon efforts in virtual fixture-, marine robotics- as well as haptics literature. Related work in each area is presented below.

A. Virtual Fixtures

The term virtual fixtures was first defined in [1] as "abstract sensory information overlaid on top of reflected sensory feedback from a remote environment." The reflected sensory information can for instance be visual, auditory, haptic (as implemented in this paper) or any combination of these. Haptic virtual fixtures have been investigated by many groups, e.g., [2], [3], [4], [5]. Although these efforts successfully implemented simpler virtual fixtures (such as *virtual rulers* etc.), virtual fixtures in a un-modeled and dynamic environment remain a challenge. A method for generating adaptive haptic virtual fixtures, which this paper builds upon, was presented in [6] where 3-DOF haptic feedback was rendered to the user upon violation of virtual fixture constraints. In that work, an RGB-D camera was used in real-time to capture the three-dimensional structure of the task space. Color segmentation was then used to define the virtual fixtures and the algorithm allowed for haptic forces to be rendered.

B. Underwater Robotics

The concept of virtual fixtures and assisted teleoperation is well-studied in many areas of robotics, yet underwater, remotely-operated vehicles (ROVs) lack this capability. As a result, and to the best of the authors' knowledge, there exists little work on rendering of dynamic virtual fixtures for underwater applications. However, sonars and subsequent three-dimensional imagery are often used for navigation and collision avoidance. Thus, there exists a set of relevant efforts that will be useful towards achieving our goal of implementing haptic rendering for underwater applications.

Work in underwater survey and mapping, where *macro* imagery is collected with underwater cameras or sonar, often focuses on simultaneous location and mapping (SLAM) [7], [8]. Eustice et. al. [9], [10], [11] have used ROVs for ship hull inspection, developing three-dimensional models of structures underwater with both visual and sonar data. The authors of [12] used a sonar-equipped ROV to map ancient cisterns on the islands of Malta and Gozo where it is important to avoid potentially damaging collisions. Collision avoidance for AUVs (autonomous underwater vehicles) was evaluated in [13] using a 3D sonar and path planner that was updated at each time step. In [14], robust state estimation was achieved using a 3D sonar.

In this work we are focused on high-resolution, small-scale imagery to support dextrous manipulation of objects underwater. In [15], a static point cloud was constructed using a laser scanner and then used for grasp planning. Similar in spirit to rendering fixtures in a virtual environment, in [16] an offline simulator for ROV/AUV intervention was presented. In the UNION project [17], dynamic disturbance rejection as well as robust underwater color segmentation was achieved. In [18] and [19], a method for underwater target localization using sonar and vision was presented. This information was then used to position an ROV or AUV relative to the target.

C. Haptic Rendering of Streaming Sensor Data

The 'virtual coupling'-method [20] is the most common theoretical framework for haptic rendering methods today. The 'god-object' method [21] and the proxy method [22] for 3-DOF rendering are based upon this idea that a virtual object can be used to simulate the ideal outcome of haptic interaction in a virtual environment. That is, the virtual object never violates any collision constraints (such as forbidden region virtual fixtures) in the virtual environment. Forces are then calculated as translational (and torsional for the 6-DOF case) springs between this ideal tool configuration and the configuration of the haptic device, which behaves non-ideally. A method for 3-DOF haptic interaction with streaming sensor data was presented in [23]. This idea was then used in [6] where forbidden-region virtual fixtures (captured by a depth sensor) were haptically rendered by superimposing spherical constraint around delicate objects. There exist several methods for 6-DOF haptic interaction with voxels [24], polygons [25], implicit surfaces [26], static point clouds [27] and streaming point cloud data [28] (which this paper builds upon).

III. Six Degree-of-Freedom Haptic Virtual Fixtures in Underwater Environments

In this paper, we present a novel method for rendering of 6-DOF haptic forbidden-region virtual fixtures and guidance virtual fixtures based on real-time sensor data. Essentially this is done by running a real-time simulation of robot motions in the actual environment to calculate the effect of haptic virtual fixtures. This is then presented to the operator. This work extends the method of rendering 3-DOF forbidden-region virtual fixtures of [6], to the 6-DOF case. In [6], the haptic feedback was based solely on the desired position of the robotic end effector, not taking rotations or torques into account. This was done by approximating the robotic end effector as a simple bounding sphere, which is not always a good approximation. The forbidden-regions were further defined by superimposing spherical regions where each sphere corresponded to a point in the point cloud captured by the a depth sensor overlooking the robot task space.

In this work, the three-dimensional structure of the robot's task space is captured in real time. The forbidden-regions are defined in a similar fashion by finding the points that correspond to a virtual fixture and then produce a forbidden region sphere around each of those points. Rendering of haptic virtual fixtures then simply corresponds to *conventional* haptic rendering of spheres (typically a large number of time-varying spheres) using a virtual-coupling method. The guidance virtual fixture can then simply be implemented as a spring to a desired target configuration.

There are several real-world challenges in rendering of haptic virtual fixtures that depend upon properties of the robot task space. These include:

1) **Registration:** The environment around the robot needs to be captured using some type of sensor with depth sensing capabilities. This sensor further has to be registered to the robot base frame. When the sensor is mounted on the robot itself, this is trivial (as the displacement can be easily measured). When, instead, the sensor is at another location, registration markers can be used.
2) **Processing of sensor data:** The sensor data needs to be processed (transformation to base frame and filtering etc.) and the virtual fixtures must be defined in real-time. Parallel processing is typically necessary to achieve this.
3) **Defining virtual fixtures:** The virtual fixtures have to be defined implicitly by the sensor data. Therefore, a method for choosing the data that corresponds to a guidance virtual fixture or a forbidden-region virtual fixture is needed. For instance, this can be a simple color segmentation algorithm.
4) **Speed of haptic rendering:** In order to meet the $1000Hz$ recommended haptic rendering rate, an efficient rendering method is needed. As haptic rendering can be seen as a simplified physics simulation, a fast collision detection method is required. A parallel collision detection method can be used in combination with a sphere tree data structure [29].

To generate virtual fixtures for an underwater task space, sensor data was captured underwater using an RGB-D sensor (RGB + depth). The data was transformed to the robot base frame and spatially low pass filtered. A color segmentation algorithm was used to define the haptic virtual fixtures and package them in a sphere tree data structure that could efficiently be used for haptic rendering. The haptic rendering then uses the desired configuration of the robot end effector in combination with the virtual fixtures to present the force to the operator.

As this work focuses on rendering of virtual fixtures rather than system integration on a robot system, the results here are presented in simulation. However, the objects involved are obtained from streaming point clouds from the camera. Our justification for this is that the virtual fixture rendering essentially runs a real-time simulation of the robot plus its surroundings. In the next phase of this work we will be incorporating a five degree of freedom submersible manipulator. If the registration and the simulation is correct we can assume that the robot is very close to where the operator commands it to be using the haptic device.

IV. Capturing and Processing Underwater Data

Before virtual fixtures can be constructed and haptic feedback can be given to the operator, the RGB-D sensor data has to be processed. This is done in two steps by first transforming and filtering the data and then defining the virtual fixtures.

A. Processing

Since image processing (both depth and RGB) is highly parallel work, it is efficient to utilize modern graphics processors (GPUs). Our method is to first transform the depth image into a point cloud using the camera parameters and then spatially low-pass filter using a bilateral filtering approach [30] such that the depth values are weighted using the Cartesian distance to neighboring points. That is, for a given depth value v_0 corresponding to a point in space $\mathbf{p_0}$, the filtered depth value v_0' is calculated as:

$$v_0' = \frac{\sum_{k=0}^{N-1} v_k \phi(k)}{\sum_{k=0}^{N-1} \phi(k)} \tag{1}$$

where N is the number of neighboring points and $\phi(k)$ are the weights such that

$$\phi(k) = e^{\frac{-(\mathbf{p_k}-\mathbf{p_0})^T(\mathbf{p_k}-\mathbf{p_0})}{2\sigma^2}} \tag{2}$$

where σ is a design parameter. The spatial bilateral filtering approach is preferable as it does not induce any time delay (unlike temporal filtering).

B. Defining virtual fixtures

With the data filtered and transformed into a point cloud, virtual fixtures can be assigned to either protect certain regions, or guide the operator towards certain regions. A color segmentation method is used to find the point that should be assigned to either a forbidden-region virtual fixture or a guidance virtual fixture. This is done by converting the RGB (red, green and blue) values to corresponding HSL (hue, saturation and lightness) values and then selecting points based on their hue value (which is orthogonal to the lightness value). This technique is very useful in environments with varying lighting conditions. As a result, a forbidden-region can for instance be defined around all *red* points and a guidance virtual fixture can for instance be generated based on all *yellow* points. Due to local memory constraints on typical GPUs, the data is processed in patches of 16x16 pixel which due to required overlapping results in some redundant work. This approach is chosen as it is one to two orders of magnitude faster to read from local memory than from global memory.

1) Defining the Forbidden-Region Virtual Fixture: For each point (corresponding to some physical object) that is assigned as a forbidden-region virtual fixture, a spherical forbidden region (with a radius chosen by the application) is constructed. Thus, a distance constraint can be maintained in all directions. The final forbidden-region virtual fixture is then constructed by superimposing multiple spherical regions, as was done in [6], see Fig 2. For purposes of rapidly being able to query this virtual fixture using a tree structure, a bounding sphere is generated around each 16x16 patch. Rather then generating a perfectly fitting sphere, a simple bounding sphere is generated such that the center is at the mean location $\tilde{\mathbf{s}}$ of the forbidden-regions and the radius such that:

$$\tilde{r} = \max_k \|\mathbf{s_k} - \tilde{\mathbf{s}}\| + r_k \tag{3}$$

where $\mathbf{s_k}$ and r_k is the location and radius of the k-th spherical virtual fixture respectively.

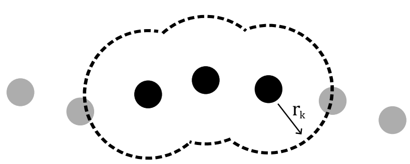

Fig. 2. Illustration of a forbidden-region virtual fixture obtained by superimposing three spherical fixtures. [6]

2) Defining the Guidance Virtual Fixture: The guidance virtual fixture is implemented as a desired 6-DOF configuration (position and orientation) in space. In terms of processing, all points with a certain hue value are selected and the mean location is chosen as the location of the guidance virtual fixture. If the selected points correspond to a plane, a normal vector can be estimated and the orientation of the virtual fixture can be based on this. Otherwise, the orientation can simply be fixed relative to the robot base frame as done in this work.

V. Rendering of Haptic Virtual Fixtures

Since the forbidden-region virtual fixture constraints are defined by superimposing spheres and since the virtual model of the robot arm can be stored in a sphere-tree (for purposes of collision detection), the most straightforward method for haptic rendering is a sphere-tree to sphere-tree haptic rendering method. The optimized sphere-tree containing the virtual tool is generated prior to operation using the Sphere Tree Toolkit [29]. The sphere-tree containing the forbidden regions is generated on-the-fly every time a frame arrives from the sensor system.

Since the haptic feedback is based on the difference (in configuration space) between the configuration of the haptic device and the configuration of an ideal virtual tool that does not penetrate any forbidden-regions, a method for moving the virtual tool is needed. Therefore, at each haptic time step ($1ms$), the virtual tool is moved iteratively towards the configuration of the haptic device. If this configuration is inside a forbidden region, the virtual tool stays at the boundary of the forbidden region and locally minimizes the *distance* (in configuration space) between the virtual tool and the configuration of the haptic device. This is done using a 'quasi-static' simulation as suggested for haptic rendering by [25]. The force and torque on the haptic device is then defined by two virtual springs (one translational and one torsional) between these two configurations. When new sensor data arrives, the virtual tool might be at a configuration inside a forbidden region, in that case the virtual tool will move away from the forbidden region. This is similar to the approach for moving a voxelized tool out of point cloud constraints in [28].

The guidance virtual fixture is used to guide the operator, using force feedback, towards a desired configuration in the robot's task space. An efficient method of implementing this is to simply augment the force contributed by the forbidden-regions with the guidance force such that the operator is *pulled* towards the guidance configuration by a virtual spring.

A. Collision detection and forming constraints

When the operator controls the robot (using the haptic device) and no forbidden-regions are violated, the virtual tool matches the configuration of the haptic device. But when the operator guides the robot into a forbidden region, this effectively forms a set of linear constraints on the virtual tool. These constraints can be found by recursively checking for collisions between the virtual tool (a sphere tree) and the forbidden regions (another sphere tree); denote these trees T_{tool} and T_{FR} respectively. This is done by first using a depth-first search of T_{tool}, querying against the highest level of T_{FR}, until the leaf node is reached. Then any occurrence of collision in T_{FR} is expanded in a similar depth-first fashion by querying against the leaf nodes of T_{tool}. If a collision has occurred between a leaf node in T_{tool} and a leaf node in T_{FR}, this effectively forms a linear motion constraint on the tool, of the following form:

$$\mathbf{n}^T\mathbf{a} + (\mathbf{r} \times \mathbf{n})^T\alpha \geq d \tag{4}$$

where $\mathbf{a}$ and α are the tool's unconstrained translational and rotational acceleration, respectively. $\mathbf{n}$ is the normal to the constraint, $\mathbf{r}$ is the vector from the tool's center of rotation to the point of collision and d is the penetration depth (see Fig. 3). No constraints are formed on the velocities as the velocity is considered zero in a quasi-static simulation.

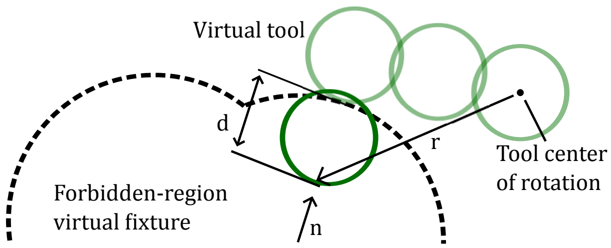

Fig. 3. Illustration of a collision constraint between a forbidden-region virtual fixture (the dashed line) and a virtual tool constructed by a sphere tree (where solid circles represent the leaf nodes). Here $\mathbf{n}$ is the normal to the constraint, $\mathbf{r}$ is the vector from the tool's center of rotation to the point of collision and d is the penetration depth. The penetration is exaggerated for clarity as the penetration typically is very small.

B. Moving the virtual tool with respect to virtual fixture constraints

Given a haptic device configuration, a virtual tool configuration and a set of linear collision/contact constraints, the goal is to minimize the distance in configuration space between the configuration of the haptic device and the configuration of the virtual tool with respect to the virtual fixture constraints. Such a minimization process can be solved as a nearest-point problem and results in a vector pointing in a feasible direction (in configuration space). Unfortunately, since the constraints are linearized, this feasible direction is only valid in a small neighborhood around the tool. Therefore, this process must be repeated iteratively such that a new minimization process is performed every time a new constraint/collision occurs. The full derivation of this minimization process is outside the scope of this paper, but can be found in [31].

C. Calculating force feedback

The resulting force feedback is calculated as a function of the difference between the haptic device and virtual tool configuration. That is, the force and torque are chosen as the springs that would bring the configuration of the haptic device to the configuration of the virtual tool.

VI. Results

The proposed virtual fixture rendering method was implemented on a Core i7-3930K CPU (Intel Corp.) with Radeon HD 6990 GPU (Advanced Micro Devices, Inc.). The force was rendered at 1000 Hz and sent to a W5D haptic device (Entact Robotics) with 6-DOF sensing and 5-DOF actuation (no actuation on the roll joint). The streaming point cloud data was captured in a water tank at the University of Washington Applied Physics Lab. The GPU processed the point cloud in real-time at $30Hz$ using the OpenCL API (Khronos Group); this includes transformation and filtering of depth and RGB data, color segmentation, as well as generation of sphere trees corresponding to the virtual fixtures. The collision detection is performed in parallel on four cores on the CPU. Depth and RGB data was captured using a Xtion Pro depth + RGB sensor (Asus) and recorded using the Robot Operating System (ROS).

We evaluate the proposed method by evaluating two different scenarios. First, two forbidden-region virtual fixtures and secondly a guidance virtual fixture in combination with a forbidden-region virtual fixture.

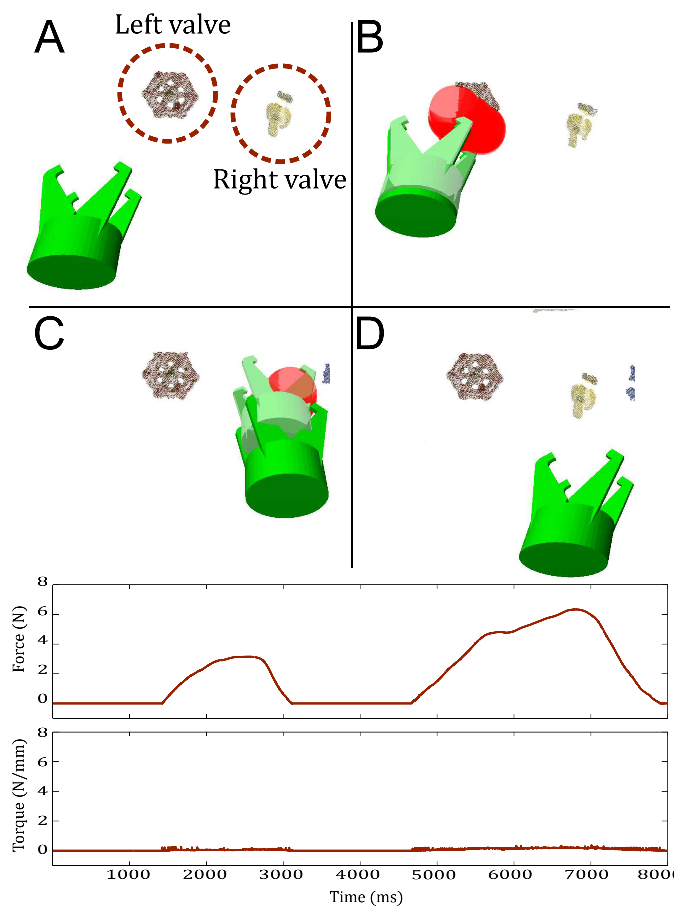

Fig. 4. The top and bottom traces show the norm of the force and torque respectively while the operator performs the first task (A-D). A) The operator starts at the initial position. B) The operator pushes the haptic device towards the left forbidden-region virtual fixture and experiences a repelling force. C) The operator pushes the haptic device towards the right virtual fixture and experiences a similar force. D) The operator moves away from the virtual fixture.

A. Evaluation of Guidance Virtual Fixtures

A forbidden-region virtual fixture in combination with a guidance virtual fixture is evaluated in a similar fashion such that the left valve is protected by a forbidden-region virtual fixture. A guidance virtual fixture is placed automatically, using color segmentation, in front of the right valve. It is activated by a key press from the operator. Fig. 5 illustrates a sequence where the operator first pushes the haptic device towards the forbidden-region and then, around $3000ms$, activates the guidance virtual fixture. As can be seen, the torque never reaches zero even though the user moves the haptic device towards the configuration of the guidance virtual fixture. This occurs partly because the operator somewhat resisted the guidance force and mainly because the W5D haptic device can not exert force on the roll joint, resulting in a residual error that only can be corrected visually.

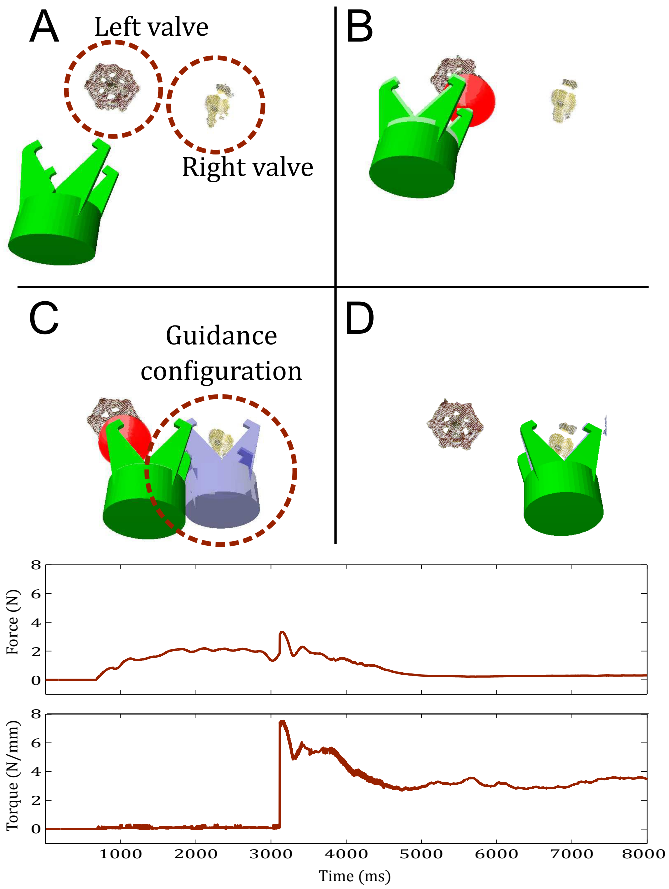

Fig. 5. The top and bottom traces illustrate the force and torque during the second task (A-D). A) The operator is at the initial position. B) The operator moves towards the left valve which is protected by a forbidden-region virtual fixture, a repelling force is therefore sent to the haptic device. C) The guidance virtual fixture is activated with a key press and the operator experiences forces and torques pushing towards the right valve. When activated, the guidance virtual fixture is visualized in blue. D) The haptic device at the configuration of the guidance virtual fixture.

B. Evaluation of Forbidden-Region Virtual Fixtures

The implementation of forbidden-region virtual fixtures is evaluated by considering an application of ROV operations that involves the opening and closing of valves. In this example task, we have a series of distinct valves in a water tank that are close to one another.

The depth and RGB data was then segmented such that only points sensor data corresponding to the two valves were selected. Two $4cm$ virtual fixtures were then defined around the two valves. As a result, the operator feels a repelling force when pushing the haptic device into the forbidden regions. Every time depth sensor data is captured, the virtual fixtures are updated to match the most recent data, thus allowing for adaptive virtual fixtures to be generated. A sequence when the operator, using a virtual representation of a tool, first tries to push the haptic device towards the left valve and then the right valve is shown in Fig. 4.

VII. Work in progress: Submersible test bed implementation

To test these ideas and algorithms for actual underwater use, we have begun system integration with a *5E Mini* submersible manipulator by ECA Robotics. This is a five degree-of-freedom submersible, all-electric (no hydraulics) manipulator developed by ECA Robotics. As a first step of this system integration we have successfully established communication between the manipulator and a PC running Linux over serial RS-485 using our own drivers. In addition, we have created a simulation of the manipulator with its full inverse kinematics running in ROS (Robot Operating System). In this simulation, we have also imported streaming RGB-D data, see Fig. 6.

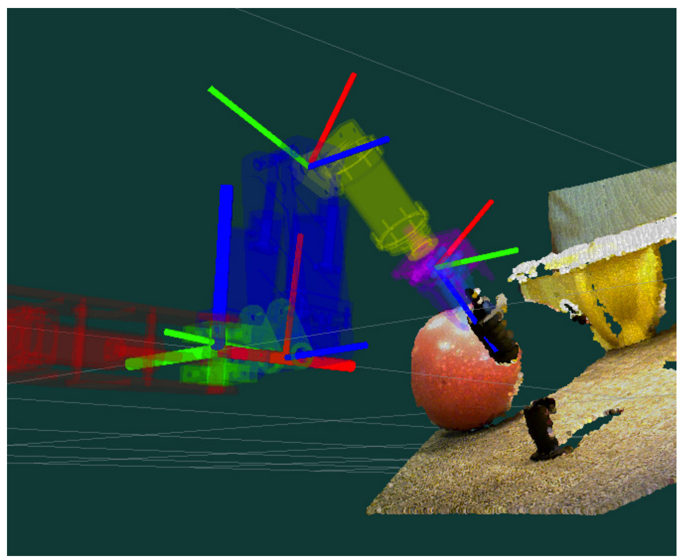

Fig. 6. Submersible manipulator simulation augmented with RGB-D data of a hybrid, wet-mate connector. Note: data captured in an indoor laboratory environment.

Since the encoders on the manipulator are relative, an initialization sequence (manipulator homing) will be needed to complete the system integration. This is necessary for registration in order to make the modeled manipulator (in the virtual space) match the physical manipulator (in the task space). The modeled manipulator is used within the virtual space to implement haptic rendering so that interactions between the manipulator and the environment (not just end effector) are captured by the algorithm. It will also be important to implement low-level safety features such as motor current limits in this implementation.

This work represents the first steps required to adapt our haptic rendering algorithm to function with this particular manipulator. To achieve this, the virtual model of the 5E Mini manipulator is imported into the haptic rendering module. Special considerations are made to account for the fact that the arm (in effect) has only four degrees-of-freedom plus grasp. The full 6-DOF rendering algorithm presented in this paper will still be useful in this case since force feedback can be used to constrain the operator to movements that are feasible with respect to the limited kinematics.

The system integration will be tested by implementing a forbidden-region virtual fixture and then a guidance virtual fixture in an indoor setting. This demo will then be submerged in APL/UW test tank facilities to demonstrate this technology's value for underwater use.

VIII. Conclusion

We have presented a method for rendering of six-degree of freedom forbidden-region and guidance virtual fixtures defined by streaming point cloud data from an RGB-D sensor in an underwater environment. The method is applicable to many underwater robotic applications such as opening valves, positioning tools or mating cables. Forces and torques are successfully rendered at $1000Hz$ with sensor data captured at $30Hz$. We demonstrate that an operator can be guided, using haptic feedback, towards a desired tool configuration while avoiding unintended collisions with the environment. Future work includes integration with the subsea manipulator and sonar system as well as an improved camera design (with HD overlay on depth data) and customization.

Acknowledgments

The authors would like to thank Entact Robotics for providing us with the W5D haptic device as well as Advanced Micro Devices for providing the GPU hardware, and OceanGate Inc. for providing use of their manipulator. We would also like to thank Professor Weichih Wang, Professor Payman Arabshahi, Chi Leung Tsui, Fong-Ru Hsieh and Ben Estroff for their assistance with the water tank tests. Ben Rush, Scott Daley, and Ryan Cox have provided much-needed assistance with developing drivers and the kinematic model of the ECA manipulator. This work was funded by SERDP 13 MRSEED01-006/MR-2323.

References

[1] L. Rosenberg, "Virtual fixtures: Perceptual tools for telerobotic manipulation," in *Virtual Reality Annual International Symposium, 1993., 1993 IEEE*, sep 1993, pp. 76 –82.

[2] J. Abbott, P. Marayong, and A. Okamura, "Haptic virtual fixtures for robot-assisted manipulation," *Robotics Research*, pp. 49–64, 2007.

[3] A. Bettini, P. Marayong, S. Lang, A. Okamura, and G. Hager, "Vision-assisted control for manipulation using virtual fixtures," *Robotics, IEEE Transactions on*, vol. 20, no. 6, pp. 953–966, 2004.

[4] J. Ren, R. Patel, K. McIsaac, G. Guiraudon, and T. Peters, "Dynamic 3-d virtual fixtures for minimally invasive beating heart procedures," *Medical Imaging, IEEE Transactions on*, vol. 27, no. 8, pp. 1061–1070, 2008.

[5] T. Gibo, L. Verner, D. Yuh, and A. Okamura, "Design considerations and human-machine performance of moving virtual fixtures," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 671–676.

[6] F. Ryden and H. Chizeck, "Forbidden-region virtual fixtures from streaming point clouds: Remotely touching and protecting a beating heart," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 3308–3313.

[7] D. Ribas, P. Ridao, J. Tardos, and J. Neira, "Underwater SLAM in a marina environment," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, 29 2007-Nov. 2, pp. 1455–1460.

[8] S. Williams, P. Newman, G. Dissanayake, and H. Durrant-Whyte, "Autonomous underwater simultaneous localisation and map building," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 2, pp. 1793–1798 vol.2.

[9] A. Kim and R. M. Eustice, "Real-time visual SLAM for autonomous underwater hull inspection using visual saliency," *IEEE Transactions on Robotics*, 2013, in Print.

[10] F. S. Hover, R. M. Eustice, A. Kim, B. Englot, H. Johannsson, M. Kaess, and J. J. Leonard, "Advanced perception, navigation and planning for autonomous in-water ship hull inspection," *International Journal of Robotics Research, Special Issue on 3D Exploration, Mapping, and Surveillance*, vol. 31, no. 12, pp. 1445–1464, October 2012.

[11] N. Carlevaris-Bianco and R. M. Eustice, "Multi-view registration for feature-poor underwater imagery," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Shanghai, China, May 2011, pp. 423–430.

[12] C. White, D. Hiranandani, C. S. Olstad, K. Buhagiar, T. Gambin, and C. M. Clark, "The malta cistern mapping project: Underwater robot mapping and localization within ancient tunnel systems," *Journal of Field Robotics*, vol. 27, no. 4, pp. 399–411, 2010. [Online]. Available: http://dx.doi.org/10.1002/rob.20339

[13] K. Teo, K. W. Ong, and H. C. Lai, "Obstacle detection, avoidance and anti collision for MEREDITH AUV," in *OCEANS 2009, MTS/IEEE Biloxi-Marine Technology for Our Future: Global and Local Challenges*. IEEE, 2009, pp. 1–10.

[14] S.-C. Yu, T.-W. Kim, G. Marani, and S. K. Choi, "Real-time 3D sonar image recognition for underwater vehicles," in *Underwater technology and workshop on scientific use of submarine cables and related technologies, 2007. Symposium on*. IEEE, 2007, pp. 142–146.

[15] M. Prats, J. J. Fernández, and P. J. Sanz, "Combining template tracking and laser peak detection for 3d reconstruction and grasping in underwater environments," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 106–112.

[16] M. Prats, J. Pérez, J. Fernández, and P. Sanz, "An open source tool for simulation and supervision of underwater intervention missions," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 2577–2582.

[17] V. Rigaud, E. Coste-Maniere, M. Aldon, P. Probert, M. Perrier, P. Rives, D. Simon, D. Lang, J. Kiener, A. Casal, J. Amar, P. Dauchez, and M. Chantler, "Union: underwater intelligent operation and navigation," *Robotics Automation Magazine, IEEE*, vol. 5, no. 1, pp. 25 –35, mar 1998.

[18] G. Marani, S. K. Choi, and J. Yuh, "Underwater autonomous manipulation for intervention missions AUVs," *Ocean Engineering*, vol. 36, no. 1, pp. 15–23, 2009.

[19] G. Marani and S. Choi, "Underwater target localization," *Robotics & Automation Magazine, IEEE*, vol. 17, no. 1, pp. 64–70, 2010.

[20] J. Colgate, M. Stanley, and J. Brown, "Issues in the haptic display of tool use," in *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, vol. 3. IEEE, 1995, pp. 140–145.

[21] C. Zilles and J. Salisbury, "A constraint-based god-object method for haptic display," in *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, vol. 3. IEEE, 1995, pp. 146–151.

[22] D. Ruspini, K. Kolarov, and O. Khatib, "The haptic display of complex graphical environments," in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1997, pp. 345–352.

[23] F. Rydén and H. J. Chizeck, "A proxy method for real-time 3-dof haptic rendering of streaming point cloud data," *IEEE Transactions on Haptics*, p. 1, 2013.

[24] W. McNeely, K. Puterbaugh, and J. Troy, "Six degree-of-freedom haptic rendering using voxel sampling," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1999, pp. 401–408.

[25] M. Ortega, S. Redon, and S. Coquillart, "A six degree-of-freedom god-object method for haptic display of rigid bodies with surface properties," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 3, pp. 458–469, 2007.

[26] S. Chan, F. Conti, N. Blevins, and K. Salisbury, "Constraint-based six degree-of-freedom haptic rendering of volume-embedded isosurfaces," in *Proc. Of the 2011 IEEE International World Haptics Conference*, 2011.

[27] J. Lee and Y. Kim, "Haptic rendering of point set surfaces," in *EuroHaptics Conference, 2007 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2007. Second Joint*. IEEE, 2007, pp. 513–518.

[28] F. Rydén and H. Chizeck, "A method for constraint-based six degree-of-freedom haptic interaction with streaming point clouds," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2013.

[29] G. Bradshaw and C. O'Sullivan, "Adaptive medial-axis approximation for sphere-tree construction," *ACM Transactions on Graphics (TOG)*, vol. 23, no. 1, pp. 1–26, 2004.

[30] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Computer Vision, 1998. Sixth International Conference on*. IEEE, 1998, pp. 839–846.

[31] S. Redon, A. Kheddar, and S. Coquillart, "Gauss' least constraints principle and rigid body simulations," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 1. IEEE, 2002, pp. 517–522.