Towards Optimal Human-Robot Interface Design Applied to Underwater Robotics Teleoperation

Paulo Padrão[1], Jose Fuentes[1], Tero Kaarlela[2], Alfredo Bayuelo[3], Leonardo Bobadilla[1]

***Abstract*— Efficient and intuitive Human-Robot interfaces are crucial for expanding the user base of operators and enabling new applications in critical areas such as precision agriculture, automated construction, rehabilitation, and environmental monitoring. In this paper, we investigate the design of intuitive human-robot interfaces for the teleoperation of dynamical systems. The proposed framework seeks an optimal interface that complies with key concepts such as user comfort, efficiency, continuity, and consistency. Moreover, we show that optimal interfaces arising from common robot tasks are user-friendly and ensure a certain degree of naturalness within our framework. As a proof-of-concept, we introduce an approach to teleoperating underwater vehicles, allowing the translation between human body movements into vehicle control commands. Field experiments were conducted utilizing a custom head-mounted display coupled with a smartphone to interpret body movements and transmit corresponding commands to operate a remotely operated vehicle. These experiments were performed in a marine robotics research testbed to validate the efficacy and practicality of our proposed interface design.**

***Index Terms*— Human-Robot Interaction, Teleoperation, Remotely Operated Vehicles Marine Robotics**

I. Introduction

The design of intuitive, robust, and efficient human-robot interfaces can extend the user base of operators and enable a new generation of applications in critical areas such as precision agriculture, automated construction, rehabilitation, and environmental monitoring [1], [2], [3]. Due to advances in virtual reality applications in the last few years, user interface design for robotic teleoperation drew increasingly more attention since the human performance of teleoperated systems can be decremented by data bandwidth, time delays, frame rates, and lack of concentration, among other interface-related factors [4]. Furthermore, environments with high spatio-temporal variability, sensing, and communication challenges, such as marine environments, provide additional challenges for teleoperation. The underwater environment is unnatural for humans, and working below the surface requires diving gear or teleoperated underwater robots. This paper presents a formal description of natural human-robot interfaces, showcasing the teleoperation of a remotely operated underwater vehicle (ROV). The proposed concept breaks the barrier between the underwater location and the user, freeing the user from harsh underwater conditions. Our general approach enables the remote operator to control an ROV with their body movements, and this method could be expanded to and tested on other similar robot platforms. Furthermore, a head-mounted display (HMD) enables the teleoperation of the ROV and provides visual feedback of the underwater world.

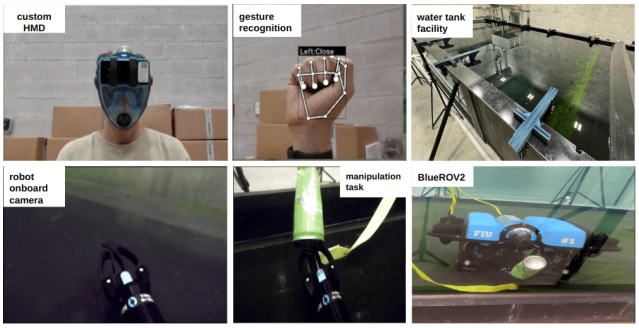


Fig. 1: Field experiment setup. A custom HMD is used to send commands to a BlueROV2 deployed in FIU's marine robotics research testbed located at Florida International University - Biscayne Bay Campus.

The teleoperation method presented provides an intuitive and natural high-level control interface for the human operator of the ROV. Due to its built-in sensors and widespread availability, the presented solution uses a smartphone to translate teleoperator head movements into ROV control commands. Potential applications that may benefit from this solution encompass underwater tasks such as inspection, welding, and rescue. By operating remotely, the teleoperator can ensure an ergonomic and safe working environment, reducing the risk of physical strain or injury. Additionally, this solution can allow for continuous work without frequent breaks or short working periods, ultimately increasing productivity and efficiency in underwater operations. In addition to industrial applications, the development of human-robot interfaces for teleoperation offers educational benefits, including problem-based learning and hands-on opportunities. These educational applications bridge theoretical concepts with real-world scenarios, providing students with practical experience [5], [6].

[1]L. Bobadilla, P. Padrão, and J. Fuentes are with the School of Computing and Information Sciences, Florida International University, Miami, FL 33199, USA. bobadilla@cs.fiu.edu, ppadraol@fiu.edu, jfuen099@fiu.edu

[2]T. Kaarlela is with Centria University of Applied Sciences, 84100 Ylivieska, Finland. tero.kaarlela@centria.fi

[3]A. Bayuelo is with the National University of Colombia, Bogotá, Colombia. ajbayuelos@unal.edu.co.

This research was funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement n° 825196). This work is supported in part by the U.S National Science Foundation (NSF) grants IIS-2034123, IIS-2024733, IIS-2331908, the Office of Naval Research grant N00014-23-1-2789, the U.S. Department of Defense (DoD) grant 78170-RT-REP, and by the U.S. Department of Homeland Security grant 23STSLA00016-01-00.

The contributions of this paper lie in the formal description of an optimization-based framework for natural human-robot interface design that complies with user comfort and efficiency constraints. Furthermore, we define and establish the connection between natural interfaces and the optimal interfaces arising from common teleoperation tasks such as exploration and manipulation, highlighting that intuitive configurations do not only meet design criteria but are also optimal.

The rest of the paper is organized as follows: Section II reviews previous research on the topic. Section III formulates the research problem. Section IV introduces the preliminaries and notations, section V presents the methods used to solve the proposed problem, and section VI-B presents the implementation of the prototype teleoperation as a proof-of-concept. Section VI presents the experimental evaluation of this work, and Section VII concludes this paper.

II. Related Work

Teleoperation has been researched for decades to overcome barriers between the teleoperator and the environment [7], [8], [9]. The barrier can be a physical barrier, such as a wall, or an environmental barrier, such as a toxic or hazardous environment or deep underwater conditions. Research on underwater teleoperation has focused on developing techniques to improve the performance and capabilities of underwater robots [10]. Current examples of underwater teleoperation applications range from developing virtual reality interfaces for underwater missions [11], [12] and the design of underwater humanoid robots [13] to visible light communication systems that can be employed for limited-range teleoperation of underwater vehicles [14]. In [15], authors investigate deep-sea manipulation through human-robot collaboration with a robotic system designed for precise underwater tasks. Additionally, Mixed Reality (MR) technologies, particularly those utilizing HMDs, provide immersive experiences and allow users to interact seamlessly with surrounding objects[16]. In [17], human–robot interfaces combining augmented reality and hand gesture detection for remote operations in hazardous environments are proposed. Our work addresses the problem of *optimal human-robot interface design*. The design of human-robot interfaces relies on the selection of appropriate desired properties, such as consistency, linearity, and continuity, that enable comfort and naturalness in teleoperation [18]. Our approach is closely related to the concepts presented in [19] in the sense of investigating the mappings between human and robot spaces and the mathematical formulation of such a problem. We also share commonalities with recent research on HMD-based immersive teleoperation interfaces [20], human perception-optimized planning [21], human–robot interaction based on gesture and movement recognition [22], and approaches that utilize optimal control for teleoperating robots [23].

III. Problem Formulation

In this paper, we consider the task of visual-based teleoperation of an underwater vehicle. The two agents involved, the person teleoperating the robot and the robot itself, have a workspace, an action space, and a configuration space. Let $\mathcal{W}_o \subset \mathbb{R}^3$ and $\mathcal{W}_r \subset \mathbb{R}^3$ be the workspaces for the operator and the robot, respectively; $\mathcal{C}_o$ and $\mathcal{C}_r$ be their configuration spaces. Let $\mathcal{U}$ denote the set of controls applicable to the robot and $\mathcal{A}$ the set of actions the human operator can perform. Following the notation from [19], we assume that the robot's dynamics are ruled by the relation given by the function $f : \mathcal{C}_r \times \mathcal{U} \longrightarrow \mathcal{C}_r$, $\dot{x} = f(x, u)$.

To establish a teleoperating system, it is necessary to map the user's actions to the robot's actions through a mapping $g : \mathcal{C}_r \times \mathcal{A} \longrightarrow \mathcal{U}$ so that the robot is affected by actions taken by the user as $\dot{x} = f(x, g(x, a))$.

Assuming fixed sets $\mathcal{A}, \mathcal{U}$ and the function f, our problem is building g according to principles such as *Continuity*, *Consistency* and *Reachability* as described in [19] are restated below. However, we may face the challenge wherein the user becomes fully immersed in the teleoperating system, leading to a lack of comprehensive or precise knowledge about the configuration state of the robot. Likewise, the information provided by the teleoperating system may present challenges concerning fast and precise user interpretation for optimal performance. Those reasons suggest the necessity of designing g to be intuitive and potentially independent of the robot's state x.

To achieve a realistic and comfortable teleoperation experience, the mapping g should fulfill key conditions [19], of which we consider five to be the most crucial. Firstly, *Consistency*, which means preserving the attributes that the robot and the operator share. In particular, this implies symmetry, meaning that if the operator's and robot's actions are symmetric with respect to a particular axis, the mapping g should preserve this symmetry as much as possible. Secondly, *Continuity*, which requires mapping actions of the operator to closely related actions of the robot. In this case, the derivative of g may also need to be restricted in order to prevent the robot from moving inconsistently due to sensory-motor aspects. Thirdly, *Linearity*, which provides an intuitive way for the agent to teleoperate the robot. In this case, it is reasonable to expect that if the input is doubled, the operator expects the robot output to be approximately doubled as well. Lastly, *Reachability* and *Completeness* allow the user to operate the robot to a given desired state. Let $\mathcal{F}_o \subseteq \mathcal{C}_o$ and $\mathcal{F}_r \subseteq \mathcal{C}_r$ be subsets of feasible configurations for the operator and robot, respectively. A state $x'_r \in \mathcal{F}_r$ is said to be u-reachable from $x_r \in \mathcal{F}_r$ if there exists a control function $u(t)$ such that x_r is brought to x'_r. In the same way, a state $x'_o \in \mathcal{F}_o$ is said to be a-reachable from $x_o \in \mathcal{F}_o$ if there exists a control function $a(t)$ such that x_o is brought to x'_o. Therefore, the function g is said to be *complete* if all u-reachable pairs $(x_r, x'_r) \in \mathcal{F}_r$ have an equivalent a-reachable pair $(x_o, x'_o) \in \mathcal{F}_o$. In general, this problem is stated as finding a function g that minimizes certain functional $J(g)$ tailored to fulfill the conditions stated before. As a general rule, this function has the form

$$J(g) = \int_0^T L(x, g, Dg)dt, \tag{1}$$

where L is a cost function and Dg indicates the derivative of g. Whenever necessary, the control function a can be computed with g, implying that the optimization problem may involve two variables. Likewise, it is assumed that both variables are required unless otherwise specified and the functions L and J are allowed to take an argument for a, denoted as $L(x, g, Dg, a)$ and $J(g, a)$ respectively. Naturally, this leads to an optimal control problem with $g \in C^1(\mathcal{C}_r \times \mathcal{A}, \mathcal{U})$ being the interface between the human and the robot that translates human actions into robot controls and the sequence of actions performed by the operator $a \in C^1(\mathbb{R}, \mathcal{A})$, both serving as solutions to the constrained problem

$$\begin{aligned} \min_{g,a} \quad & \int_0^T L(x, g, Dg, a)dt \\ \text{s.t.} \quad & \dot{x} = f(x, g(x, a)) \\ & x(0) = x_{initial},\ x(T) = x_{final}, \end{aligned} \tag{2}$$

where $C^1(X, Y)$ denotes the set of functions with continuous derivative from X to Y.

We emphasize that the robot's state $x = x(t)$ and the action taken by the user $a = a(t)$ have temporal dependencies (one state and action per timestamp t) and we dropped them in (2) to ease the notation. In any case, the solution to this problem involves solving a Jacobi-Bellman equation [19], which requires solving a partial differential equation to obtain optimal functions. This procedure is, in general, challenging. Furthermore, the resulting solutions may not fulfill the principles described previously, or the translation scheme provided for them may be difficult for an average person to execute. Therefore, we will study particular cases that have an intuitive design and are easy to manage for an average person.

Problem definition: Translating configuration spaces

Given the configuration spaces of the operator and the robot $\mathcal{C}_o$, $\mathcal{C}_r$ respectively, and the set of actions $\mathcal{A}$ the operator can perform, compute an optimal mapping g that translates the operator's action space into robot's action space $\mathcal{U}$.

IV. Preliminaries

Fig. 2 depicts a mapping between the actions of the operators and the actions of the robot. This mapping exhibits key properties such as continuity and consistency, meaning that small operator movements result in corresponding small robot movements and that these actions are reversible and consistent. As a result, the robot is highly intuitive to control even though the robot's reachability is somewhat limited, as not all operator actions can be translated into robot actions.

Our purpose in this work is to show that natural mappings can be obtained as solutions for common tasks, demonstrating that they are also optimal interfaces. To this end, theoretical arguments were developed to show and quantify how far an interface g is from being "natural". In this context, we define a *natural* interface as one that assigns a single user action to a corresponding robot action, without depending on the robot's state, i.e. $g : \mathcal{A} \longrightarrow \mathcal{U}$. Introducing such a dependency would make the operation process more cumbersome and less intuitive; instead, the interface solely relies on the user's action. If g were a linear transformation, one might expect it to exhibit behavior similar to that of a permutation matrix P with scaled rows. Therefore, it becomes necessary to exercise tight control over the first derivative of g, Dg. This necessity gives rise to Definition 1.

Definition 1: Let $\mathcal{A} \subseteq \mathbb{R}^n$, $p(a)$ a probability density function distribution defined on $\mathcal{A}$, $g \in C^1(\mathcal{A}, \mathcal{U})$, σ a permutation of the numbers $1, \ldots, n$, $P_\sigma \in \mathcal{M}_{n\times n}(\mathbb{R})$ the permutation matrix whose rows are permuted according to σ, Λ a diagonal matrix whose diagonal entries come from a vector $\lambda \in \mathbb{R}^n$ and $||\cdot||$ a norm defined on the space of the matrices. We say that g is (ϵ,p)-*unnatural* interface if

$$\int_{\mathcal{A}} \min_{\sigma,\lambda} \{||Dg(a) - \Lambda P_\sigma||\} p(a)da \leq \epsilon. \tag{3}$$

Equation (3) quantifies the extent to which a potential natural interface, denoted as g, deviates from being considered natural on average. If an interface is $(0, p)$-unnatural for any probability density function p, it is considered completely natural. Furthermore, when g is an affine transformation, this measure becomes independent of p. In that case, g is said to be ϵ-unnatural.

We turn our attention to the interfaces g that behave as linear transformations, as the natural-unnatural concepts rely on the first derivative of g. From this point forward, we assume that g can be described using a matrix $G = Dg$. The functional $J(g)$ described in (2) can be rewritten as $J(G, a)$, adding a as a new variable and G as a description of g. The regularized problems considered are of the form $\mathcal{L}(G, a) = J(G, a) + \alpha R(G)$, where $\alpha > 0$ and $R(G)$ is a regularizer function. The regularizer $R(G)$ enforces G to have certain characteristics while minimizing $J(G, a)$. This formulation is more numerically tractable than constrained problems. Importantly, the functional $\mathcal{L}(G, a)$ should include a regularization term $R(G)$ related to (3), allowing the enforcement and estimation of the naturalness of g by calculating $R(G)$ for a given function g.

The following analysis focuses on bounding the value of $R(G)$ when it is used to regularize optimization problems. Since $R(G)$ is related to (3), it will estimate the naturalness of an interface G (i.e. g). Decoupling the min operator computed for the pair (x, y) into two consecutive min operators applied one after the other will be beneficial throughout the subsequent analysis. Specifically, it is well known that the join min operator can be decoupled by minimizing individually over the arguments of a function, as provided by Proposition 1.

Proposition 1: Let $\mathcal{X}$, $\mathcal{Y}$ two Banach spaces, $F(x, y)$ a functional defined on $\mathcal{X} \times \mathcal{Y}$. Suppose that the function $G(y) = \min_x F(x, y)$ is well-defined for each $y \in \mathcal{Y}$, then

$$\min_{x,y} F(x, y) = \min_y G(y) = \min_y \{\min_x F(x, y)\}. \tag{4}$$

We apply Proposition 1 to $\mathcal{L}(G, a)$ in order to define $F(G) = \min_a J(G, a)$ and ensure that

$$\min_G F(G) + \alpha R(G) = \min_{G,a} \mathcal{L}(G, a). \tag{5}$$

This separation allows for the study of the properties of the optimal interface G given optimal user actions a. On the other hand, it is important to examine the relationship between the optimization problem in Eq. (5) and

$$\begin{aligned} &\min_G F(G) \\ &\text{s.t. } R(G) \leq r \end{aligned} \tag{6}$$

for $r \geq 0$. This consideration allows us to connect the regularized optimization problem with the constrained problem. The regularization constant α provides a bound on the value of the regularization function evaluated at the optimum, i.e., it allows the estimation of r. This bound will be particularly valuable in Section VI. By solving Eq. (5) to find a minimizer $G^\star$ and restricting the search to the super level set $L^-_{R(G^\star)}(R)$, we ensure that there is no $G \in L^-_{R(G^\star)}(R)$ such that $F(G) < F(G^\star)$. This contradicts the fact that $G^\star$ serves as a minimizer. This understanding, condensed in Proposition 2, is relevant for our subsequent analysis.

Proposition 2: Let $G^\star$ be a minimizer of the problem (5), then $G^\star$ is a minimizer of the problem (6) with $r = R(G^\star)$. Moreover, the inequality constraint $R(G) \leq r$ can be replaced by $R(G) = r$ in Eq. (6).

Lastly, the values of r and α can be related by first-order conditions arising from (5), where $G^\star$ and α satisfy $F(G^\star) + \alpha R(G^\star) = 0$. If the Hessian matrix $D^2F(G) + \alpha D^2R(G)$ has an inverse when $G = G^\star$, then, by the inverse function theorem, $G^\star$ can be isolated by a mapping $G^\star = \omega(\alpha)$ such that $F(\omega(\alpha)) + \alpha R(\omega(\alpha)) = 0$ in some neighborhood of $(G^\star, \alpha)$. Therefore, $r = R(\omega(\alpha))$. Theoretically, this suggests that r can be defined before the optimization process by choosing α appropriately.

As a proof of concept, we address the problem of translating operator spaces into robot spaces, by examining the scenario wherein the human operator transmits commands to the robot using head movements. The range of motion for the human head depends on several factors, such as age, sex, health, and individual anatomical differences. On average, an adult human can rotate their head up to 90 degrees to either side and tilt them up and down about 45 degrees, giving a total range of motion of about 180 degrees. The movements mentioned above are captured by a smartphone's built-in sensors, including the inertial measurement unit (IMU) and barometer, which are affixed to a diving mask worn by the operator. The operator's movements are subsequently translated into commands to enable the teleoperation of the underwater robot. The complete architecture is shown in Fig. 4. The visual feedback is provided to the operator through the robot camera, which captures images of the robot's environment and displays them on the smartphone screen. This allows the operator to see where the robot is going and adjust its movements accordingly. Due to the restrictions of user input commands, the robot is initially treated as a rigid body that moves at a constant speed in $\mathbb{R}^2$. Let a_θ and a_ψ be the head pitch and head yaw commands, respectively, and u_θ and u_ψ be the robot camera tilt command and yaw command of the robot base. Let v_o and v_r be the linear forward velocities of the operator and the robot in $[m/s]$, respectively. Let a_{close} and a_{open} describe the user action of closing and opening hands, respectively, and let u_{close} and u_{open} describe the robot's gripper discrete state (closed and open). We define the action space of the operator as $\mathcal{A} = (a_{\theta\,\min}, a_{\theta\,\max}) \times (a_{\psi\,\min}, a_{\psi\,\max}) \times (v_{o\,\min}, v_{o\,\max}) \times \{a_{\text{close}}, a_{\text{open}}\}$ and the action space of the robot as $\mathcal{U} = (u_{\theta\,\min}, u_{\theta\,\max}) \times (u_{\psi\,\min}, u_{\psi\,\max}) \times (v_{r\,\min}, v_{r\,\max}) \times \{a_{\text{close}}, a_{\text{open}}\}$.

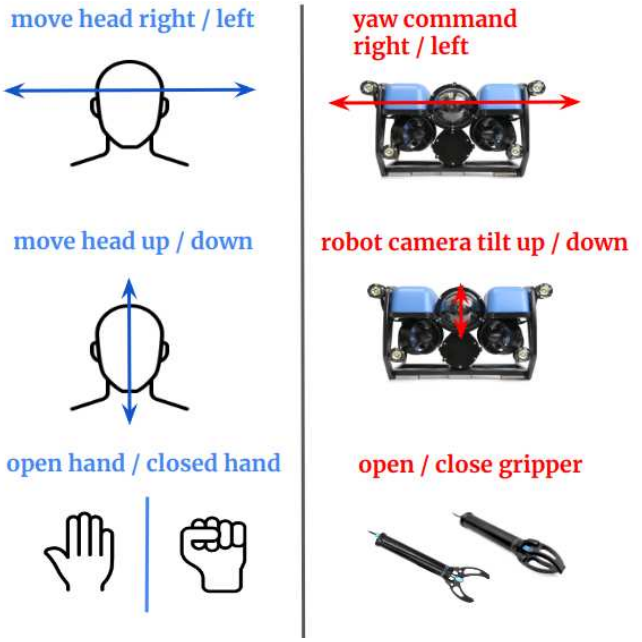


Fig. 2: An example of human-robot action space translation

To evaluate the mappings, we modified functionals defined in [19] to reflect specific factors that are relevant to underwater teleoperation. An important formulation is the shortest-distance problem where

$$\begin{aligned} &\min_{g,a} \int_0^T L(x, g, Dg)dt = \min_{g,a} \int_0^T ||\dot{x}||dt \\ &\text{s.t } \dot{x} = f(x, g(x, a)) \\ &\quad x(0) = x_{initial},\ x(T) = x_{final}. \end{aligned} \tag{7}$$

Several works have reported the need to maintain a comfortable environment for the user during teleoperation tasks [21], [24]. In this context, the goal is to reduce the number of movements made by the person to increase their comfort level when operating the robot. For example, [21] focuses on minimizing the number of head movements to improve user comfort. This approach can be seen as simplifying the curvature of a path, where intuitively, paths with fewer turns are considered more comfortable. In the subsequent section, additional terms will be added to the formulation to enforce the desired properties.

V. METHODS

There could be numerous potential interface designs. This work aims to show that the natural interface (Fig. 2) is optimal in some sense. To this end, we consider the aforementioned action spaces $\mathcal{A}$ and $\mathcal{U}$ of the operator and the robot, respectively. Since the camera angle does not affect the robot's movement, we consider an operator who can perform two actions: moving their head and body (e.g., by walking or moving their hands). These actions are denoted as $a_{head}(t)$ and $a_{body}(t)$, respectively, and the operator's action space is represented as $a(t) = [a_{head}(t), a_{body}(t)]^\top$, where $\mathcal{A} \subseteq \mathbb{R}^2$ and $\mathcal{C}_0 \subseteq \mathbb{R}^2$. The kinematic model of the robot is given by

$$\begin{aligned} \dot{x}_{pos}(t) &= v(t)\cos(\theta(t)) \\ \dot{y}_{pos}(t) &= v(t)\sin(\theta(t)) \\ \dot{\theta}(t) &= w(t) \end{aligned} \tag{8}$$

where $v(t)$ is the forward speed and $\omega(t)$ is the angular speed. The robot's action and state spaces are represented as $u(t) = [v(t), w(t)]^\top$ and $\dot{x}(t) = [\dot{x}_{pos}(t), \dot{y}_{pos}(t), \dot{\theta}(t)]^\top$, respectively, where $\mathcal{U} \subseteq \mathbb{R}^2$ and $\mathcal{C}_r \subseteq \mathbb{R}^2 \times S^1$.

As described in Problem 1, the objective is to find the mapping g. In this case, g is assumed to be a linear transformation given by $u(t) = Ga(t)$, which exhibits several properties including continuity, linearity, and consistency under certain conditions. We consider the task of moving the robot from the initial point $x_{initial}$ to the final point, x_{final}, as this is a common task during environment exploration. The goal is to find the optimal interface g and the control policy $a(t)$ that should be applied by the operator. To achieve this, the following optimization problem is defined, inspired by (7)

$$\begin{aligned} \min_{g,a} \quad & \alpha||x_{final} - x(T)||^2 + \beta \int_0^T a(t)^\top M a(t) dt \\ & + \gamma \int_0^T ||\dot{x}(t)|| dt + \delta \, dist(G, O(2)) \\ \text{S.t.} \quad & \dot{x}(t) = \begin{bmatrix} \cos(\theta(t)) & 0 \\ \sin(\theta(t)) & 0 \\ 0 & 1 \end{bmatrix} u(t), \quad x(0) = x_{initial}. \end{aligned} \tag{9}$$

Here, M is a positive-definite matrix, $O(2)$ is the set of orthogonal matrices of size 2×2, and the coefficients α, β, γ, and δ are non-negative regularization coefficients that determine the relative importance of each term. The first term ensures that the desired point is reached given the control policy of the operator, after being transformed by the interface. The second term measures the effort made by the user, with a higher cost assigned to head movements compared to body movements, to maintain a comfortable interface for the user. The third term considers the distance the robot traverses, encouraging it to take the optimal path. The fourth term $R(G) = dist(G, O(2)) = \min_Q ||G - Q||_F$, $Q \in O(2)$ encourages G to preserve angles. Hence, G should be an orthogonal matrix; this requirement aims to better fulfill the consistency criterion. This term is expressed as the distance between G and the set of orthogonal matrices in the Frobenius norm, which is $R(G) = ||UV^\top - G||_F$, where $G = U\Sigma V^\top$ is the singular value decomposition of the matrix G.

Equation (3), the definition of an unnatural interface, and the regularization functional $R(G)$ are connected by the following proposition

Proposition 3: Let $R(G) = \min_Q ||G - Q||_F$, $Q \in O(n)$, λ, Λ, σ and P_σ as in (3). Then,

$$\min_{\lambda,\sigma} ||G - \Lambda P_\sigma|| \leq k(R(G) + \sqrt{n}). \tag{10}$$

Where k is a constant that depends only on the norms $||\cdot||$ and $||\cdot||_F$.

Proof. Let $Q \in O(n)$ be an orthogonal matrix. We start with the triangle inequality

$$\begin{aligned} ||G - \Lambda P_\sigma|| &\leq ||G - Q|| + ||Q - \Lambda P_\sigma|| \\ &\leq k(||G - Q||_F + ||Q - \Lambda P_\sigma||_F). \end{aligned} \tag{11}$$

The second inequality arises from the fact that the space is finite-dimensional, and every norm is equivalent. Specifically, there exists a constant k that depends only on the norms $||\cdot||$ and $||\cdot||_F$ such that $||A|| \leq k||A||_F$ for each fixed-size matrix A.

Proposition 1 is heavily relied upon to apply the min operator by each variable individually, rather than applying the complete join operator to obtain

$$\begin{aligned} \min_\lambda\{||G - \Lambda P_\sigma||\} &\leq k \min_\lambda\{(||G - Q||_F + ||Q - \Lambda P_\sigma||_F)\} \\ &= k(||G - Q||_F + \min_\lambda ||Q - \Lambda P_\sigma||_F). \end{aligned} \tag{12}$$

It can be shown by using Lagrange multipliers and the fact that the rows of an orthogonal matrix form an orthonormal basis of $\mathbb{R}^n$ that

$$\min_\lambda ||Q - \Lambda P_\sigma||_F = \sqrt{\sum_i \sum_{j \neq \sigma(i)} q_{ij}^2} \leq \sqrt{n} \tag{13}$$

so, (12) is updated to

$$\min_\lambda\{||G - \Lambda P_\sigma||\} \leq k(||G - Q||_F + \sqrt{n}). \tag{14}$$

Moreover, the right-hand side of inequality 14 does not depend on σ and the left-hand side does not depend on Q. Thus, the min operator can be applied on both sides with respect to σ and Q leading to

$$\begin{aligned} &\min_{\lambda,\sigma}\{||G - \Lambda P_\sigma||\} \leq k(||G - Q||_F + \sqrt{n}) \\ &\min_{\lambda,\sigma}\{||G - \Lambda P_\sigma||\} \leq k(\min_Q\{||G - Q||_F\} + \sqrt{n}) \\ &\qquad = k(R(G) + \sqrt{n}), \end{aligned} \tag{15}$$

which is our desired bound. □

Leveraging the definition of the naturalness of an interface g (Definition 1), we have tailored a functional following the average human operational demands outlined in Eq. (9). Through this process, we demonstrate that optimal interfaces derived from this functional must meet a specific level of naturalness (Propositions 2 and 3). This combination encapsulates a key aspect of our contribution, establishing that interfaces characterized by naturalness also align with the optimality criterion outlined in Eq. (9).

VI. Experimental Evaluation

A. Finding optimal human-robot interfaces

Simulation experiments were conducted to determine the optimal actions for the operator. An example of a navigation task is illustrated in Fig. 3. Additionally, we utilized functional (9) to derive the optimal set of actions $a(t) = [a_{head}(t), a_{body}(t)]^\top$ (Fig. 3, top). Our findings indicate that the operator can accomplish the assigned task more efficiently by prioritizing body movements over excessive head movements. Consequently, we found the control applied to the robot, computed as $u(t) = Ga(t)$ (Fig. 3, middle), and the resulting path taken by the robot $x(t)$ (Fig. 3, bottom). More important is the mapping g, which turned out to be

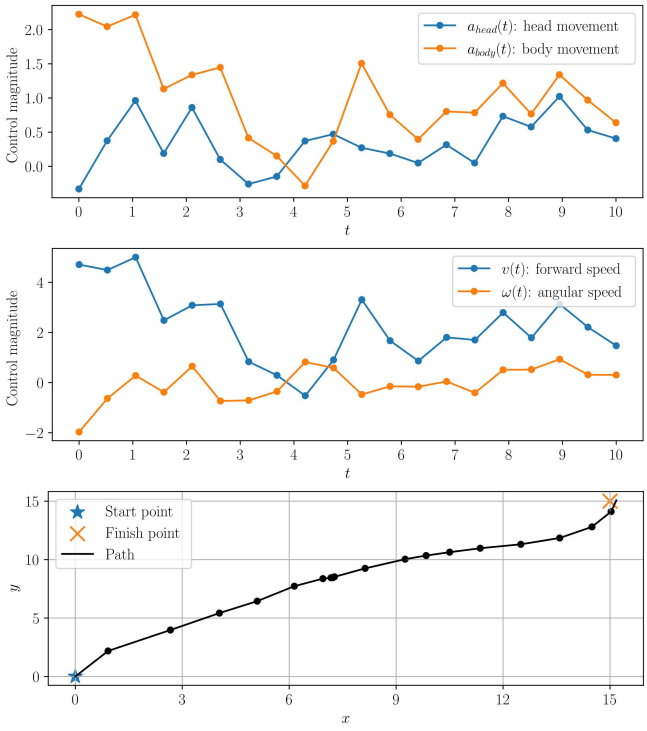


Fig. 3: Experiment results solving problem (9). $x_{initial} = [0, 0, \pi/2]^\top$, $x_{final} = [15, 15, -\pi/2]^\top$, $M = \text{diag}(10, 1/2)$, $\alpha = 250$, $\beta = 10, \gamma = 5$ and $\delta = 10$. Control $a(t)$ performed by the operator (top); control $u(t)$ mapped to the robot (middle); and simulated path task (bottom).

$$G^\star = \begin{bmatrix} 0.24 & 2.15 \\ 1.73 & -0.62 \end{bmatrix} \approx \begin{bmatrix} 0 & 2.15 \\ 1.73 & 0 \end{bmatrix}. \tag{16}$$

Assuming the coarse approximation expressed in (16), it can be observed that there exists a relationship between the human control $a(t)$ and the robot control $u(t)$ described as

$$\begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} = u(t) = G^\star a(t) \approx \begin{bmatrix} 0 & 2.15 \\ 1.73 & 0 \end{bmatrix} \begin{bmatrix} a_{head}(t) \\ a_{body}(t) \end{bmatrix}$$
$$\begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} \approx \begin{bmatrix} 2.15 \cdot a_{body}(t) \\ 1.73 \cdot a_{head}(t) \end{bmatrix}. \tag{17}$$

It can be inferred that the natural mapping obtained from optimizing (9), in which the user operates the robot's yaw movement by their head orientation, is not only intuitive and user-friendly but also optimal. Regarding our bounds computed previously to estimate how natural g is, we found that at $dist(G^\star, O(2)) \approx 1.45$. Besides, according to (10), $G^\star$ is at most 2.89-unnatural. When we compute (3) $G^\star$ is 0.66-unnatural. Also, the closest natural interface ΛP_σ (see Definition 1) parametrized by λ and $P\sigma$ is

$$\lambda = (1.73, 2.15),\ P_\sigma = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{18}$$

which is precisely the coarse approximation in Eq. (16).

B. Software-in-the-Loop Experiments

The Software-in-the-Loop (SIL) configuration employed the BlueSim hardware simulator [25] instead of real hardware for software component development and configuration. BlueSim simulates the BlueROV2 hardware [26], providing a virtual camera unit for testing and refining the system. The teleoperating control device consists of a diving mask, casing, and smartphone. To access the smartphone's inertial measurement unit (IMU) data and stream it to the teleoperator workstation, we utilized the SensorServer application [27]. A custom extension was developed to receive the sensor data stream and translate the orientation data into directional commands for the ROV, as well as up and down commands for the ROV camera. Also, an OpenCV-based application was developed to process the video stream and transmit only black-and-white images to conserve bandwidth for the teleoperator while keeping meaningful information. Fig. 4 presents the architecture of the initial prototype and connections of the SIL and Hardware-in-the-Loop (HIL).

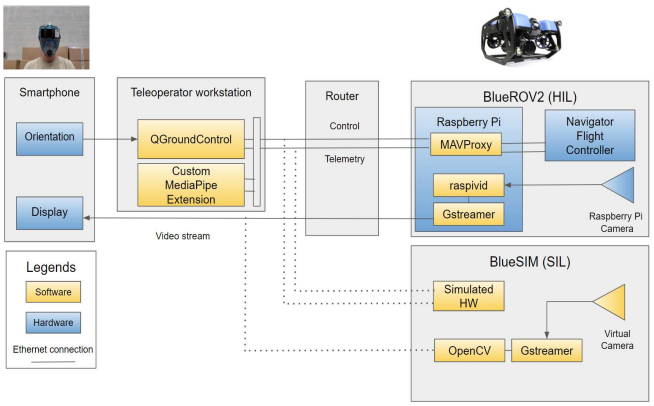
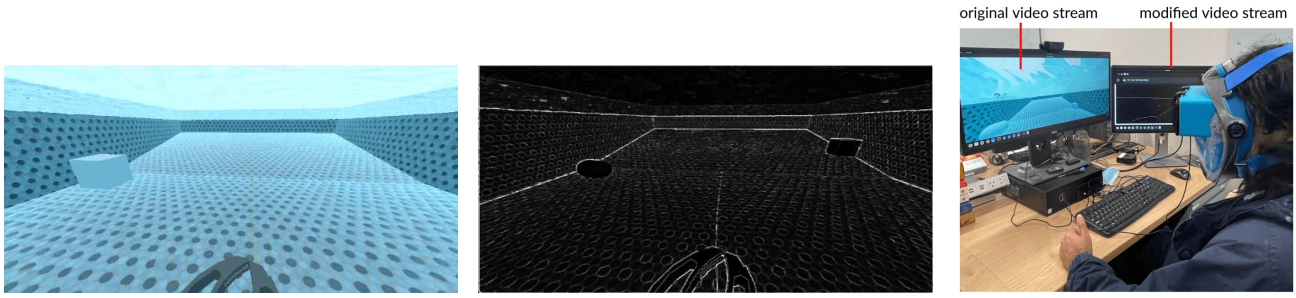


Fig. 4: Architecture of the initial prototype.

To gather quantifiable data on the effectiveness of the presented solution, the framework is tested by a group of volunteers based on the guidelines of [21]. The experimental procedure is described in three tasks. In the first task, each user is provided with a virtual empty pool scenario and has 3 minutes to familiarize themselves with the headset and simulator. In the second task, the user is provided with an RGB video stream from the virtual robot's front camera and is asked to identify a cubic shape in the pool by pointing the robot camera at the respective shape. Only one cubic shape



Simulation scenario: task 2 | Simulation scenario: task 3 | Experimental setup

Fig. 5: Experimental setup for underwater teleoperation simulation. (left) Task 2: find the cubic shape provided RGB video stream; (middle) Task 3: find the oval shape provided edge-detected, black-and-white video stream; (right) the original video stream is displayed to the user's HMD for tasks 1 and 2, and the modified video stream is provided for task 3.

is located in one corner of the pool (Fig. 5, left). To conserve bandwidth, in the third task, the user is provided with a black-and-white video stream of the pool and is asked to identify an oval shape by pointing the robot camera at the respective shape (Fig. 5, middle).In this step, there is an oval shape in one of the pool's corners and a cubic shape in another.

Regarding the implementation of the user commands strategy, we used a fixed-size buffer to collect enough samples at the beginning of the simulation and avoid unnecessary rotations of the simulated ROV. To move the ROV, we compute the moving average of the buffer, where each sample represents the difference between two consecutive measurements of head orientations. The resulting moving average at each time step is then checked against a predefined threshold to decide which way (left or right) the ROV should turn. The time required to complete tasks 2 and 3 is recorded during the piloting.

The group of users exhibited a balanced distribution with respect to gender, comprising three females and three males, with a mean age of 24.5 years (Fig. 5, right). Regarding the HMD, the experiments indicate that optimal teleoperation comfort can be achieved by increasing the distance between the eyes and the smartphone. Additionally, we noted that laggy communication significantly increases task completion times, as users must wait for image updates on their phone screens. As expected, users could detect shapes faster when presented with RGB streams, while black-and-white streams resulted in comparatively longer detection times. The average completion time and standard deviation for tasks 2 and 3 were $\approx 14.77 \pm 5.18$ seconds and $\approx 32.44 \pm 5.62$ seconds, respectively.

C. Field Experiments

Field experiments are carried out at FIU's marine robotics research testbed, a water tank of size 7.6m $\times$ 4.5m $\times$ 1.8m, with an approximate water capacity of $45,400$L located in Florida International University's Biscayne Bay Campus, as illustrated in Figures 1 and 6.

We conducted three different sets of experiments. The first set focuses on controlling the ROV camera gimbal using the information provided by the smartphone's IMU associated with up and down head movements. The second set of experiments is designed to explore left and right head movements to command robot yaw actions accordingly. Finally, the last set of experiments encompasses simple manipulation tasks, such as grabbing and releasing a soda can placed near the bottom of the tank. This is accomplished by translating appropriate hand gestures into open and close commands for the robot gripper. Fig. 6 shows a snapshot of the experimental procedure, and the complete video is provided at `https://youtu.be/8QjVyL8I0GU`. The physical BlueROV2 establishes a tethered connection with an above-surface workstation, enabling the transmission of RGB video stream utilizing GStreamer [28]. All robot commands are sent through QGroundControl (QGC) via Pymavlink, a Python implementation of the MAVLink protocol. Google Mediapipe is utilized to translate recognized hand gestures into meaningful BlueROV2 gripper commands [29]. For gesture recognition, the neural network architecture consisted of a 6-layer sequential model. The input layer is a one-dimensional array of length 21×2, covering landmarks for both hands [29]. Following the input layer, a 20% dropout layer was used for regularization and to prevent overfitting. Subsequently, a fully connected layer with 20 units and a ReLU activation function was added. Then, another dropout layer with a dropout rate of 40% was introduced, followed by a dense layer with ten units, also using the ReLU activation function. Finally, the output layer used the softmax activation function to produce output probabilities for each class (open or closed hand). The final model comprises 1,092 parameters. For model compilation, we utilized the sparse categorical cross-entropy loss function and employed a stochastic gradient descent method based on Adam optimization. Each epoch involved training the model for 1,000 iterations, with a batch size of 128. This training process resulted in an accuracy of 98%.

VII. Conclusions and Future Work

In this paper, we investigated the design of human-robot interfaces for robotics teleoperation based on key concepts such as linearity, consistency, continuity, and user comfort. As a proof-of-concept, the proposed solution was applied

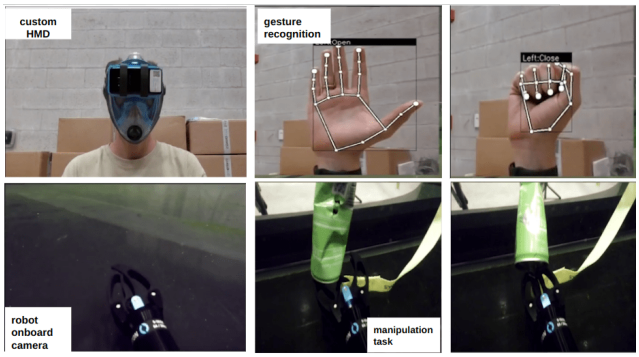


Fig. 6: Field experiment: Manipulation task. The gripper is actuated (opened/closed) by the user's hand movements

to perform underwater tasks from a safe and ergonomic location. Additionally, we established and developed a theoretical framework emphasizing the necessity of natural and user-friendly interfaces. Consequently, we demonstrated that optimal interfaces arising from common tasks possess a certain level of naturalness. The experimental simulation results involved a group of volunteers to collect quantifiable data to assess the effectiveness of the presented solution. Additionally, field experiments conducted in a marine robotics testbed validated the real-world applicability of our approach. Our future work will enhance the overall system's functionality by expanding the proposed action space and configuration space mappings. This will include incorporating the vehicle's depth control, translational motion, and additional hand gestures for manipulation.

REFERENCES

[1] K. Hauser, "Recognition, prediction, and planning for assisted teleoperation of freeform tasks," *Autonomous Robots*, vol. 35, 11 2013.

[2] A. Birk, T. Doernbach, C. Mueller, T. Łuczynski, A. Gomez Chavez, D. Koehntopp, A. Kupcsik, S. Calinon, A. K. Tanwani, G. Antonelli, P. Di Lillo, E. Simetti, G. Casalino, G. Indiveri, L. Ostuni, A. Turetta, A. Caffaz, P. Weiss, T. Gobert, B. Chemisky, J. Gancet, T. Siedel, S. Govindaraj, X. Martinez, and P. Letier, "Dexterous underwater manipulation from onshore locations: Streamlining efficiencies for remotely operated underwater vehicles," *IEEE Robotics & Automation Magazine*, vol. 25, no. 4, pp. 24–33, 2018.

[3] J. A. Cardenas, Z. Samadikhoshkho, A. U. Rehman, A. U. Valle-Pérez, E. H.-P. de León, C. A. Hauser, E. M. Feron, and R. Ahmad, "A systematic review of robotic efficacy in coral reef monitoring techniques," *Marine Pollution Bulletin*, vol. 202, p. 116273, 2024.

[4] J. Y. C. Chen, E. C. Haas, and M. J. Barnes, "Human performance issues and user interface design for teleoperated robots," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1231–1245, 2007.

[5] T. Kaarlela, P. Padrao, T. Pitkääho, S. Pieskä, and L. Bobadilla, "Digital twins utilizing XR-technology as robotic training tools," *Machines*, vol. 11, no. 1, 2023.

[6] T. Kaarlela, T. Pitkäaho, S. Pieskä, P. Padrão, L. Bobadilla, M. Tikanmäki, T. Haavisto, V. Blanco Bataller, N. Laivuori, and M. Luimula, "Towards metaverse: Utilizing extended reality and digital twins to control robotic systems," *Actuators*, vol. 12, no. 6, 2023.

[7] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Berlin, Heidelberg: Springer-Verlag, 2007.

[8] D. Lee and M. Spong, "Passive bilateral teleoperation with constant time delay," *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 269–281, 2006.

[9] J. Kofman, X. Wu, T. Luu, and S. Verma, "Teleoperation of a robot manipulator using a vision-based human-robot interface," *IEEE Transactions on Industrial Electronics*, vol. 52, no. 5, pp. 1206–1219, 2005.

[10] M. Moniruzzaman, A. Rassau, D. Chai, and S. M. S. Islam, "Teleoperation methods and enhancement techniques for mobile robots: A comprehensive survey," *Robotics and Autonomous Systems*, vol. 150, p. 103973, 2022.

[11] M. De la Cruz, G. Casañ, P. Sanz, and R. Marín Prades, "A new virtual reality interface for underwater intervention missions," *IFAC-PapersOnLine*, vol. 53, pp. 14600–14607, 01 2020.

[12] J. J. Roldán, E. Peña-Tapia, P. Garcia-Aunon, J. Del Cerro, and A. Barrientos, "Bringing adaptive and immersive interfaces to real-world multi-robot scenarios: Application to surveillance and intervention in infrastructures," *IEEE Access*, vol. 7, pp. 86319–86335, 2019.

[13] W. Wu, C. Yang, Z. Xu, X. Wu, Y. Zhu, and Q. Wei, "Development and control of a humanoid underwater robot," in *2020 6th International Conference on Mechatronics and Robotics Engineering (ICMRE)*, pp. 6–11, 2020.

[14] R. Codd-Downey and M. Jenkin, "Wireless teleoperation of an underwater robot using Li-Fi," in *2018 IEEE International Conference on Information and Automation (ICIA)*, pp. 859–864, 2018.

[15] G. Brantner and O. Khatib, "Controlling ocean one: Human–robot collaboration for deep-sea manipulation," *Journal of Field Robotics*, vol. 38, no. 1, pp. 28–51, 2021.

[16] C. Domingues, M. Essabbah, N. Cheaib, S. Otmane, and A. Dinis, "Human-robot-interfaces based on mixed reality for underwater robot teleoperation," *IFAC Proceedings Volumes*, vol. 45, no. 27, pp. 212–215, 2012. 9th IFAC Conference on Manoeuvring and Control of Marine Craft.

[17] K. A. Szczurek, R. M. Prades, E. Matheson, J. Rodriguez-Nogueira, and M. D. Castro, "Multimodal multi-user mixed reality human–robot interface for remote operations in hazardous environments," *IEEE Access*, vol. 11, pp. 17305–17333, 2023.

[18] K. J. Mimnaugh, M. Suomalainen, I. Becerra, E. Lozano, R. Murrieta-Cid, and S. M. LaValle, "Analysis of user preferences for robot motions in immersive telepresence," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4252–4259, 2021.

[19] K. Hauser, "Design of optimal robot user interfaces," in *Workshop on Progress and Open Problems in Motion Planning, 2011 International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[20] J. Halkola, M. Suomalainen, B. Sakcak, K. J. Mimnaugh, J. Kalliokoski, A. P. Chambers, T. Ojala, and S. M. LaValle, "Leaning-based control of an immersive-telepresence robot," in *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 576–583, IEEE, 2022.

[21] M. Suomalainen, A. Q. Nilles, and S. M. LaValle, "Virtual reality for robots," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11458–11465, 2020.

[22] X. Li, "Human–robot interaction based on gesture and movement recognition," *Signal Processing: Image Communication*, vol. 81, p. 115686, 2020.

[23] I. Havoutis and S. Calinon, "Supervisory teleoperation with online learning and optimal control," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1534–1540, 2017.

[24] I. Becerra, M. Suomalainen, E. Lozano, K. J. Mimnaugh, R. Murrieta-Cid, and S. M. LaValle, "Human perception-optimized planning for comfortable vr-based telepresence," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6489–6496, 2020.

[25] W. Galvani and P. Pereira, "Bluesim." `https://github.com/bluerobotics/bluesim`, 2023. Accessed 23 February 2023.

[26] Bluerobotics Inc., "Bluerov2." `https://bluerobotics.com/store/rov/bluerov2/`, 2023. Accessed 2 February 2023.

[27] U. Farooq, "SensorServer." `https://github.com/umer0586/SensorServer`, 2021. Accessed: March 25, 2024.

[28] Gstreamer team, "Gstreamer open source multimedia framework." `https://gstreamer.freedesktop.org/`, 2023. Accessed 8 February 2023.

[29] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "Mediapipe: A framework for perceiving and processing reality," in *Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR) 2019*, 2019.