An assisted re-synchronization method for robotic teleoperated tasks

Alexander Pérez and Jan Rosell

***Abstract*— Teleoperation tasks are performed at cartesian level when the robot and the haptic device have dissimilar kinematics. If the size of the workspaces is also dissimilar, as it is usually the case, the mapping between workspaces must be handled with care in order to let the user teleoperate the robot in a natural and precise way. This paper formulates the mapping of workspaces based on the re-synchronization method and proposes an assisted system that lightens the user from the tedious part of the method, by guiding him/her towards the best re-synchronization position, thus minimizing the number of jumps. The proposal is part of a teleoperated guiding system being developed by the authors.**

***Index Terms*— Teleoperation, haptic devices, workspace mapping.**

I. INTRODUCTION

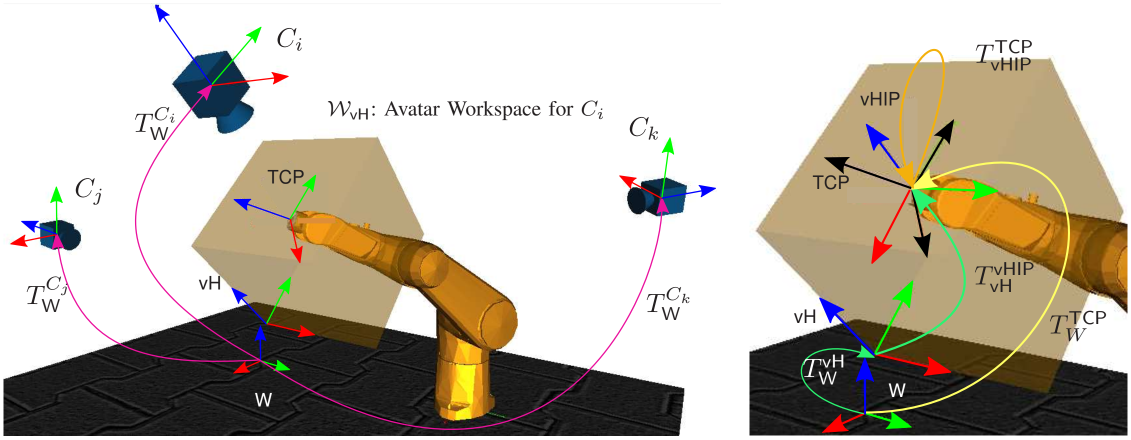
Robotic teleoperated tasks are performed by a mechanical manipulator (a robot) controlled remotely by a human operator equipped with a force reflecting interface (a haptic device). The teleoperation should be transparent and intuitive, which requires the robot, the haptic device and the human arm to have similar kinematic structures. In this sense, some approaches work on the design of haptic devices that do not restrict the arm mobility by interfacing the human limb in a tight and natural way [1], while others work on the design of telemanipulators able to replicate the human motion abilities [2]. However, in many teleoperation scenarios the robot and the haptic device have very dissimilar kinematics, like when an industrial robot is teleoperated with a ground-based (desktop) haptic device. This fact requires the teleoperation be performed in cartesian space, using the inverse kinematics of the robot. One of the problems associated with this approach is the mapping between the workspace of the haptic device and that of the robot, because usually the haptic device workspace is much smaller than that of the robot. There are several methods to perform this mapping [3]:

- **Re-syncronization** (also called indexing or clutching): This method is equivalent to the classical 2D mouse jump. It relies on a position control mode where the displacements (position and orientation) of the end-effector of the haptic device are directly mapped to displacements of the end-effector of the robot, with a given scaling factor that can be adjusted by the user. When reaching the limits of the haptic workspace, the user releases the synchronization (using an additional input) and repositions the haptic device (without moving the robot), and synchronizes again the robot with the haptic device at its new position [4].
- **Ballistic tracking:** This method adjusts the scaling between workspaces as a function of the velocity of the haptic device, i.e. when the haptic device is moved quickly the scaling factor is big and a coarse position control is performed, and when it is moved slowly the scaling factor is small and a fine position control results.
- **Rate control:** This method does not provide a direct physical mapping between workspaces but instead the velocity of the robot is proportional to the position of the haptic device, like joysticks do [5].
- **Drift control:** This method consists in a continuous and imperceptible repositioning of the haptic workspace while the user moves the device. This drift is proportional to the velocity of the user hand and to the distance of the haptic end-effector from the center of the workspace, which makes the user to move away from the limits of the workspace [3].

Each of this methods has its advantages and disadvantages. The use of re-synchronization needs of an additional input (a switch, button or pedal) to uncouple the haptic device, and the re-synchronization action may interfere the teleoperation of the task, which can be a problem if some part of the task is to be done in a continuous way. On the other hand, the control is done at position level, which is useful when teleoperating precision tasks. The advantage of the ballistic tracking is supposed to be the automatic scaling, but this can produce an offset when the user moves rapidly in one direction and slowly in the opposite one, thus loosing the control over the exploration zone of the workspace. Rate control is intuitive and useful for coarse motions but it is not adequate for precision tasks, nor for fast (accelerated) motions. Drift control has the advantages of the position control, without the disadvantages of the re-synchronization method, but some distortion may be felt by the user depending on the gain drift, i.e. the forces that produce the drift may interfere the other forces produced by virtual contacts or bilateral teleoperation control. To profit the advantages and minimize the disadvantages, combination of methods are also found, like the "buble" method [6] that combines position control and rate control, or the method proposed in [7] that combines drift control with ballistic tracking.

The teleoperation of precision tasks can be difficult and tiring for the user. Different aids can be provided to cope with this problem, like augmented reality, relational positioning or haptic guiding [8], [9]. This latter aid consists in the

The authors are with the Institute of Industrial and Control Engineering (IOC), Technical University of Catalonia (UPC), Barcelona, Spain (jan.rosell@upc.edu). A. Pérez is also with the Escuela Colombiana de Ingeniería "Julio Garavito", Bogotá D.C. This work was partially supported by the Spanish Government through the projects PI09/90088, DPI2008-02448 and DPI2010-15446.



(a) Avatar workspace

(b) Virtual coupling.

Fig. 1. Model of a remote robotic cell for the execution of teleoperated tasks equipped with a Stäubli TX90 robot and three cameras: a) Camera i is the active camera and the avatar workspace is aligned with it; b) Virtual coupling is attained by making the origins of frames vHIP and TCP coincident.

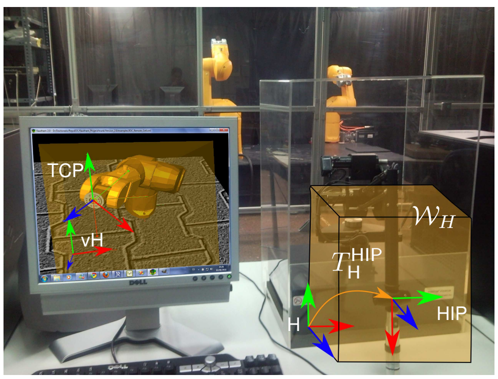


Fig. 2. User tools for the teleoperation of robotic tasks: (left) image fed back by the active camera; (right) the haptic device and its workspace.

generation of haptic guiding forces to help the operator in the execution of the task, i.e. the user has full control of the task being teleoperated, but senses some forces that suggest him/her the best motion direction to complete the task. These guiding forces are obtained using path planning techniques, which is possible when the model of the task is available. The teleoperation of precision tasks using a guiding aid requires that the mapping between the haptic device workspace and the robot workspace be performed using the re-synchronization method, because position control allows the transparent mapping needed for precision tasks and no drift force may interfere with the guiding forces. Within an assisted teleoperation framework being developed at our lab, that includes a haptic guiding aid, this paper first presents a novel mapping model between workspaces where the camera plays a relevant role. Then, based on this mapping, an automatic re-synchronization method is proposed that leverages the difficulties of re-synchronizations, taking full advantage of its benefits. The paper is structured as follows. The mapping problem is tackled in Section IV, that presents the expressions of the transformations that are to be set each time a re-synchronization is required. The re-synchronization aid is presented in Section V, Section VI evaluates the proposal and Section VII concludes the work.

II. Problem Statement and Overview

A. Nomenclature

Let consider the teleoperation framework shown in Fig. 1 and Fig. 2, composed of a robot, a haptic device and several cameras. The following workspaces and frames are shown:

- $\mathcal{W}_H$: workspace of the haptic device.
  - H: reference frame of $\mathcal{W}_H$.
  - HIP: reference frame attached to the end-effector of the haptic device.
  - $T_{\mathsf{H}}^{\mathsf{HIP}}$: transformation between H and HIP. It is read from the haptic device.
- $\mathcal{W}_W$: workspace of the world where the robot is located.
  - W: reference frame of $\mathcal{W}_W$.
  - TCP: reference frame attached to the end-effector of the robot.
  - T_{W}^{TCP}: transformation between W and TCP. It is periodically recomputed during teleoperation, and can be read from the robot controller.
- $\mathcal{W}_{\mathsf{vH}}$: virtual workspace of the haptic device into the scene, i.e. the embedding of $\mathcal{W}_H$ into $\mathcal{W}_W$. It is also called the **avatar workspace**.
  - vH: reference frame of $\mathcal{W}_{\mathsf{vH}}$.
  - vHIP: reference frame of the avatar of the end-effector of the haptic device.
  - $T_{\mathsf{vH}}^{\mathsf{vHIP}}$: transformation between vH and vHIP. It is computed from $T_{\mathsf{H}}^{\mathsf{HIP}}$ and scale factors.
  - $T_{\mathsf{W}}^{\mathsf{vH}}$: transformation between W and vH. It is set during the synchronization procedure.
  - $T_{\mathsf{vHIP}}^{\mathsf{TCP}}$: transformation between vHIP and TCP. It is set during the synchronization procedure (it is a pure rotation).
- $\mathcal{W}_{C_i}$: workspace of camera i (several cameras may be available, and the one chosen to feed back the images is called the active camera).
  - C_i: reference frame of camera i.
  - $T_{\mathsf{W}}^{C_i}$: transformation between W and C_i. It locates camera i with respect to the world reference frame.

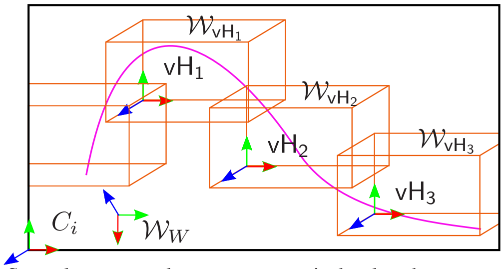


Fig. 3. Several avatar workspaces consecutively placed to cover the whole path.

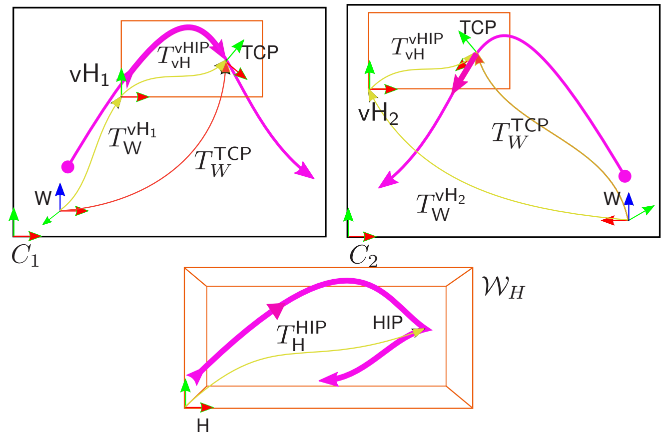


Fig. 4. Change between two opposite cameras: Top figures show in magenta the TCP path as seen from the cameras (to the right when seen from C_1 and to the left when seen from C_2). The region covered by the virtual haptic workspace is shown as an orange square. The bottom figure shows in magenta the path followed by the HIP within the haptic workspace (it moves to the right when camera C_1 is used and to the left when camera C_2 is used) .

B. Problem Statement

Given a teleoperation task to be performed, the following problems are to be solved:

- **Teleoperation**: Find the expression of T_W^{TCP} with respect to $T_{\text{H}}^{\text{HIP}}$. The user manipulates the haptic device and hence changes $T_{\text{H}}^{\text{HIP}}$, i.e. the position and orientation of HIP with respect to its base H. These motions must be translated to the robot, i.e. to changes of the TCP with respect to the reference frame W.
- **Mapping**: Find the mapping between the haptic device workspace and the robot workspace. This includes:
  - *Scaling*: Find the relation between the physical and the virtual workspaces of the haptic device, i.e. find the expression of $T_{\text{vH}}^{\text{vHIP}}$ from $T_{\text{H}}^{\text{HIP}}$ using rotational and translational scale factors.
  - *Synchronization*: Find the relationship between the avatar workspace and the robot workspace, defined by the transformation T_{W}^{vH} that relates the corresponding reference frames, and the transformation $T_{\text{vHIP}}^{\text{TCP}}$ that relates the corresponding end-effector frames. This synchronization uses the camera frame C_i as a common point between the haptic and the scene frames.
- **Re-synchronization aid**: At user requirement, update the synchronization transformations. This includes:
  - *Optimization*: Find a new pose of the haptic device, $T_{\text{H}}^{\text{HIP}}$, that allows resuming the teleoperation task and executing it for as long as possible, i.e. minimizing the need of future re-synchronizations (Fig. 3).
  - *Guiding*: Generate the forces to be exerted to the user in order to guide him/her towards the new pose of the haptic device.

C. Overview

The teleoperation problem is tackled in Section III. The proposed method allows a natural correlation between the motions of the user manipulating the end-effector of the haptic device and the resulting motions of the end-effector of the robot shown by the active camera. Therefore the active camera provides the coupling between the scene and the haptic space. This is achieved with the following two guidelines:

- *Position guideline:* Make the origins of the frames TCP and vHIP coincident, i.e. consider $T_{\text{vHIP}}^{\text{TCP}}$ a pure rotational transformation (Fig. 1b).
- *Orientation guideline:* Make the orientations of vH and of the active camera coincident, i.e. consider[1] $ori(T_{\text{W}}^{\text{vH}}) = ori(T_{\text{W}}^{C_i})$ (Fig. 1a).

III. Teleoperation

The main objective of the whole framework is to provide the user with the capability to move the robot TCP by moving the HIP of the haptic device in concordance with the image that comes from the active camera. Following Fig. 1b, T_W^{TCP} is computed as:

$$T_W^{\text{TCP}} = T_{\text{W}}^{\text{vH}} \cdot T_{\text{vH}}^{\text{vHIP}} \cdot T_{\text{vHIP}}^{\text{TCP}} \tag{1}$$

Where:

- $T_{\text{vH}}^{\text{vHIP}}$ is recomputed from $T_{\text{H}}^{\text{HIP}}$, as detailed in subsection IV-A, each time $T_{\text{H}}^{\text{HIP}}$ is updated from the haptic device (the period of the haptic loop is approximately 1 ms).
- T_{W}^{vH} and $T_{\text{vHIP}}^{\text{TCP}}$ are calculated, as detailed in subsection IV-B, each time a new synchronization is required.

While the teleoperation is performed, it is governed by Eq. (1), i.e. the changes in $T_{\text{vH}}^{\text{vHIP}}$ are mapped to T_W^{TCP}; and each time a new synchronization is required, both T_{W}^{vH} and $T_{\text{vHIP}}^{\text{TCP}}$ are updated. This requirement follows three possible strategies to improve the teloperation:

- An increase of visibility, that can be achieved by changing the active camera.
- An increase of reachability, that can be achieved by moving the avatar workspace.
- An increase/decrease of precision, that can be achieved by changing the mapping scale factors.

[1] If T is an homogeneous transformation, $pos(T)$ and $ori(T)$ are used to represent, respectively, the corresponding position and orientation.

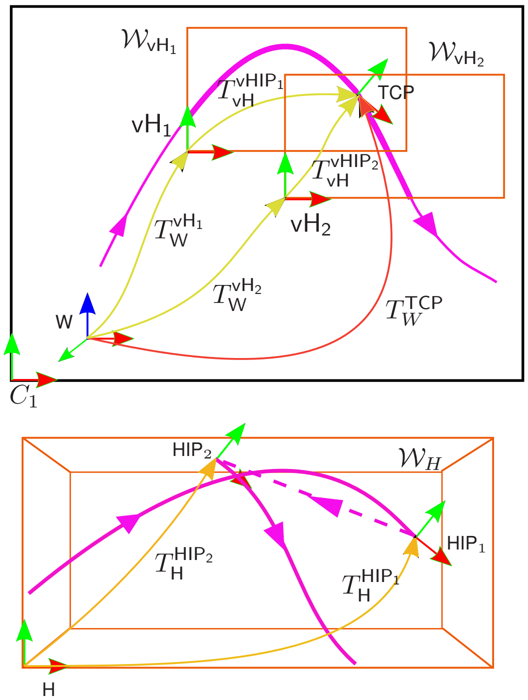


Fig. 5. Change of the avatar workspace: Top figure shows in magenta the TCP path and the virtual haptic workspace located at two different positions (orange squares). Bottom figure shows the HIP path divided into three parts: the first and the last correspond to the teleoperation, and the rectilinear middle part (dashed) corresponds to the motion of the HIP while the robot is detached.

As an example of the first case, Fig. 4 shows the motion of the HIP when the user switches between two cameras, being one in front of the other and covering almost the same volume of the scene (note the position and orientation of W). As an example of the second case, Fig. 5 shows the motion of the HIP when the user re-locates the avatar workspace.

IV. Mapping

The mapping deals with the relationship between the haptic device workspace and that of the robot. Following the guidelines in subsection II-C, all mathematical expressions of the mapping are aimed to match the position of vHIP with the TCP of the robot while keeping the avatar workspace aligned with the active camera frame.

A. Scaling

$T_{\mathsf{vH}}^{\mathsf{vHIP}}$ is computed from $T_{\mathsf{H}}^{\mathsf{HIP}}$. Once $T_{\mathsf{H}}^{\mathsf{HIP}}$ is read from the haptic device, the translational and the rotational parts are scaled separately in order to allow $\mathcal{W}_{\mathsf{vH}}$ to cover a certain volume of $\mathcal{W}_W$. The rotational scaling factor is applied directly to the rotation angle, keeping the rotation axis unchanged.

Let $AA(v_x, v_y, v_z, \alpha)$ be the axis-angle representation of a rotation of an angle α around an axis (v_x, v_y, v_z), and $\mathcal{S}_t$ and $\mathcal{S}_r$ be the translational and the rotational scales respectively. Then:

$$\begin{aligned} AA(v_x, v_y, v_z, \alpha) &= ori(T_{\mathsf{H}}^{\mathsf{HIP}}) \\ ori(T_{\mathsf{vH}}^{\mathsf{vHIP}}) &= AA(v_x, v_y, v_z, \mathcal{S}_r \cdot \alpha) \\ pos(T_{\mathsf{vH}}^{\mathsf{vHIP}}) &= pos(T_{\mathsf{H}}^{\mathsf{HIP}}) \cdot \mathcal{S}_t \end{aligned} \tag{2}$$

The scaled angle of the avatar is forced to the range $[0 - 2\pi)$, to allow the user to feel and control the angular movements more easily, i.e. even if the user does a large movement of the gimball angles in the haptic device or if he/she chooses an unsuitable rotational scale (i.e. too large), the rotation angle never exceeds 2π in a single turn.

B. Synchronization procedure

At the initialization of the teleoperation or when the user wants to change either the active camera, the mapping scales or the position of the avatar workspace, the system needs to be synchronized to create a new virtual connection between the robot and the haptic device. Besides allowing to move the robot with the haptic device, this connection must provide a simple and direct correlation between the motions of the haptic device and those of the robot as seen from the images fed back by the active camera. This synchronization can be achieved with the guidelines of Section II-C following two steps:

1) Determine $T_{\mathsf{vHIP}}^{\mathsf{TCP}}$: As stated by the *position guideline* this transformation is pure rotational. Using the *orientation guideline* ($ori(T_{\mathsf{W}}^{\mathsf{vH}}) = ori(T_{\mathsf{W}}^{C_i})$), $T_{\mathsf{vHIP}}^{\mathsf{TCP}}$ is calculated as follows[2]:

$$\begin{aligned} T_{\mathsf{vHIP}}^{\mathsf{TCP}} &= \left(\begin{array}{c|c} \mathcal{R}_{\mathsf{vHIP}}^{\mathsf{TCP}} & \mathbf{0} \\ \hline 0 & 1 \end{array}\right) \\ \mathcal{R}_{\mathsf{vHIP}}^{\mathsf{TCP}} &= (\mathcal{R}_{\mathsf{vH}}^{\mathsf{vHIP}})^{-1} \cdot \mathcal{R}_{\mathsf{vH}}^{\mathsf{TCP}} \\ &= (\mathcal{R}_{\mathsf{vH}}^{\mathsf{vHIP}})^{-1} \cdot (\mathcal{R}_{\mathsf{W}}^{\mathsf{vH}})^{-1} \cdot \mathcal{R}_{\mathsf{W}}^{\mathsf{TCP}} \\ &= (\mathcal{R}_{\mathsf{vH}}^{\mathsf{vHIP}})^{-1} \cdot (\mathcal{R}_{\mathsf{W}}^{\mathsf{Ci}})^{-1} \cdot \mathcal{R}_{\mathsf{W}}^{\mathsf{TCP}} \end{aligned} \tag{3}$$

   $\mathcal{R}_{\mathsf{vH}}^{\mathsf{vHIP}}$ is the value of the rotation of the vHIP read after the repositioning of the haptic device (when the robot is attached again to the haptic device); $\mathcal{R}_{\mathsf{W}}^{\mathsf{Ci}}$ is the rotation of the camera chosen as the current active camera; and $\mathcal{R}_{\mathsf{W}}^{\mathsf{TCP}}$ is the value of the rotation of the robot TCP at the synchronization instant (when the robot was detached from the haptic device).

2) Determine $T_{\mathsf{W}}^{\mathsf{vH}}$: Once $T_{\mathsf{vHIP}}^{\mathsf{TCP}}$ is computed using Eq. (3), it is used to update $T_{\mathsf{W}}^{\mathsf{vH}}$ as follows.

$$T_{\mathsf{W}}^{\mathsf{vH}} = T_W^{\mathsf{TCP}} \cdot (T_{\mathsf{vHIP}}^{\mathsf{TCP}})^{-1} \cdot (T_{\mathsf{vH}}^{\mathsf{vHIP}})^{-1} \tag{4}$$

V. The re-synchronization aid

Consider a teleoperation framework with a haptic guiding aid that, by providing slight forces to the operator, suggests him/her the motions to be performed (this kind of aid alleviates the task burden and allows the teleoperation to be performed more quickly). This guiding help can be calculated based on a planned path since many teleoperated task are performed in known environments with available models (that can be updated with sensor information during the execution to recalculate the paths).

When the whole robot path planned to be teleoperated does not fit within the avatar workspace (due to the size of the workspaces of the robot and of the haptic device and the mapping scales used), a sequence of re-synchronizations are

[2] $\mathcal{R}$ represents the rotation matrix of an homogeneous transformation T, i.e. $\mathcal{R} = ori(T)$.

required. As an example Fig. 3 showed a possible solution of successive locations of the avatar workspace that allowed to cover the robot path.

This re-synchronization process can be viewed as an optimization problem, either global or local. Seen from a global scope, the problem to be tackled is to find the minimum number of successive locations of the avatar workspace that cover the robot path. The position and orientation of each of these locations depend on the haptic device workspace, the mapping scales, the proposed robot path and the position and orientation of the active camera used to execute the task. The use of a global optimization, however, is not useful in teleoperation since it should be completely redone whenever the operator decides not to follow the suggested motions.

In the local scope, on the other hand, the problem is reduced to find the next location of $\mathcal{W}_{\mathsf{vH}}$ that cover the proposed robot path as much as possible, thus locally minimizing the need of a new synchronization. It is assumed that the re-synchronization is done at user requirement, i.e. although the path could still be followed in the current position of $\mathcal{W}_{\mathsf{vH}}$ the operator may decide to re-synchronize because he/she may not feel comfortable at the present posture. At the instant when the re-synchronization is required, the user is allowed to change the active camera or select new translational and rotational scales.

A proposal to solve this local optimization problem is presented in this paper that both computes the best new location of $\mathcal{W}_{\mathsf{vH}}$ (Section V-A), and computes the forces that guide the user to the associated new HIP position (Section V-B). The current approach is restricted to the translational part, since many haptic devices don't provide torque feedback and, as tested experimentally, rotational motions are very difficult to suggest from torque feedback (apart from the rotation along the stylus axis). The paths of the robot TCP to be guided are considered piecewise linear paths in $SE(3)$, represented as ordered sequences of reference frames. The points of a path are defined as the origins of these reference frames, and are assumed to satisfy that the distance between two consecutive points is small compared with the translational size of the avatar workspace.

A. Optimization

The optimization procedure first computes the largest bounding box B that satisfies two conditions:

1) Contains the current position of the vHIP and as much points of the path as possible.
2) Fits within the avatar workspace.

Then places $\mathcal{W}_{\mathsf{vH}}$ such that B is centered in it and computes the corresponding new vHIP position. Algorithm 1 formalizes the procedure, illustrated in Fig. 6, and uses the following functions and nomenclature:

- P_d: Ordered set of points of the robot path.
- $P_d(i)$: Point number i of set P_d.
- P: Set of points.
- Box(P): Function that computes the bounding box of the points in the set P.

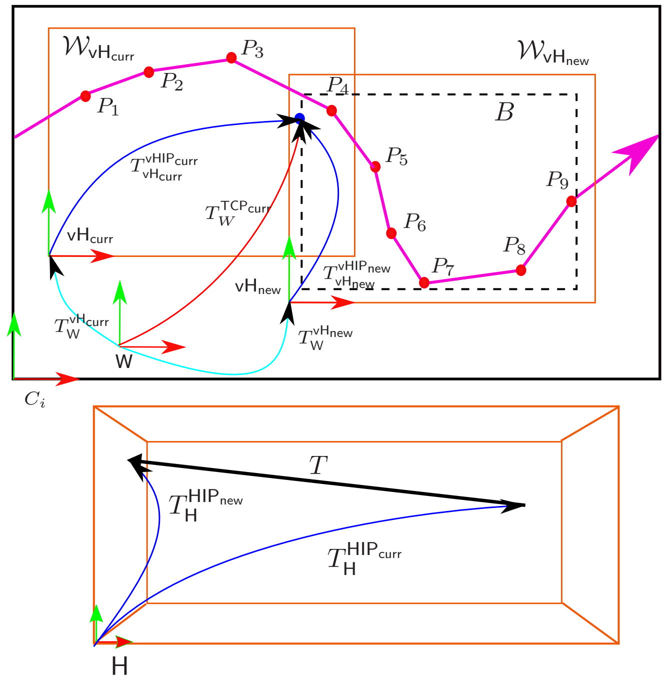


Fig. 6. Obtention of HIP_{new} using Algorithm 1.

- Dist2Nearest(P_d,x): Function that returns the index of the point in P_d that is nearest to x.
- Fit(X,Y): Function that returns true if volume X fits within volume Y, and false otherwise.
- Find-Workspace(B): Function that returns the transformation $T_{\mathsf{W}}^{\mathsf{vH}}$ that locates vH such that box B is centered in $\mathcal{W}_{\mathsf{vH}}$.

Algorithm 1 New $T_{\mathsf{vH}}^{\mathsf{vHIP}}$.

Require: $T_W^{\mathsf{TCP_{curr}}}$, transformation of the current TCP.
Ensure: $T_{\mathsf{vH_{new}}}^{\mathsf{vHIP_{new}}}$, suggested new transformation of vHIP.

i=Dist2Nearest$(P_d, pos(T_W^{\mathsf{TCP_{curr}}}))$
$P = \{pos(T_W^{\mathsf{TCP_{curr}}})\}$
B=Box$(\emptyset)$
repeat
  $B_{prev} = B$
  $P = P \cup P_d(i)$
  B=Box(P)
  $i = i + 1$
until Fit(B,$\mathcal{W}_{\mathsf{vH}}$)=FALSE
$T_{\mathsf{W}}^{\mathsf{vH_{new}}}$ = Find-Workspace(B_{prev})
return $T_{\mathsf{vH_{new}}}^{\mathsf{vHIP_{new}}} = \left[T_{\mathsf{W}}^{\mathsf{vH_{new}}}\right]^{-1} T_W^{\mathsf{TCP_{curr}}}$

B. Guiding

Once Algorithm 1 computes $T_{\mathsf{vH_{new}}}^{\mathsf{vHIP_{new}}}$, the scaling process detailed in Section IV-A can be reversed to obtain $T_{\mathsf{H}}^{\mathsf{HIP_{new}}}$. The motion from the current position of the haptic device, HIP_{curr}, to the new position HIP_{new} is described by the following transformation (Fig. 6 Bottom):

$$T = \left[T_{\mathsf{H}}^{\mathsf{HIP_{curr}}}\right]^{-1} T_{\mathsf{H}}^{\mathsf{HIP_{new}}} \tag{5}$$

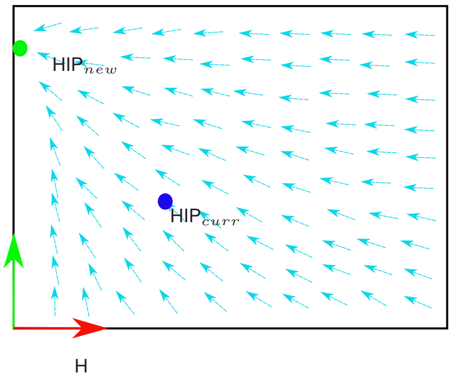


Fig. 7. Force field able to guide the user towards HIP$_{new}$.

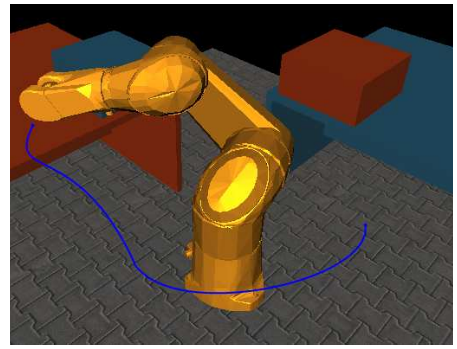


Fig. 8. Pick-and-place teleoperated task used as a test for the assisted re-synchronization aid.

The haptic device must exert forces in the direction defined by $pos(T)$ in order to guide the motion. As the user moves the haptic device, $T_{\mathsf{H}}^{\mathsf{HIP_{curr}}}$ is continually updated and transformation T recomputed, therefore resulting like a magnetic field that attracts HIP$_{curr}$ towards HIP$_{new}$, as illustrated in Fig. 7 for a simplified 2D case.

The user has control of the moment where he wants the teleoperation to resume, i.e. he/she can wait till the proposed HIP$_{new}$ is reached or not. The guiding force towards HIP$_{new}$ disappears as soon as teleoperation resumes.

VI. Evaluation

The validation of the proposed approach has been carried out in a virtual environment with simulated elements. A robot simulation toolkit for motion planning and teleoperation guiding has been developed in our lab, and has served to generate and validate the paths and to simulate the teleoperated task without delays. Several experiments have been designed to evaluate the usefulness of the re-synchronization aid. As an example Fig. 8 shows a simple pick-and-place task for a TX90 robot that has to be teleoperated with a Phantom haptic device (Fig. 2). The task consists in moving a block from the top of one desk to the floor under another desk.

The user is free to change the camera, to re-synchronize the haptic device and the robot at any time, and he/she is never forced to follow the path proposed during the test. Several users have done the test twice, with and without the re-synchronization aid. Half of them have first done the test with the aid and afterwards without, and the other half the other way round. The execution time and the number of re-synchronizations done have been recorded. The use of the re-synchronization aid both reduced the execution time ($20-35\%$ of reduction) and the number of re-synchronizations needed (more than 25% of reduction). Similar results were obtained for the other tasks tested.

VII. Conclusions

A teleoperation framework with a novel workspaces mapping model and an assisted re-synchronization tool has been proposed to ease the teleoperation of robotic tasks. It has the following main features: a) the teleoperation framework provides a simple and intuitive correlation between the motions applied to the haptic device and the actual motions of the robot as seen from visual information fed back by the active camera using the camera frame instead of the world frame, as usually done; b) during the teleoperation the user can change the active camera to increase the visibility, or change the scales that relates the workspaces to adjust the precision; c) at user requirement the system computes the new position of the virtual haptic workspace from where the teleoperation can resume and proceed for as long as possible, and guides the user to that position using force feedback.

The proposal has been tested in the teleoperation of virtual tasks, and results show that an increase in the teleoperation performance is obtained, both as a decrease in the execution time and in the number of re-synchronizations required. Experiments with the real robot are currently being implemented.

References

[1] M. Folgheraiter, B. Bongardt, J. Albiez, and F. Kirchner, "A bio-inspired haptic interface for tele-robotics applications," in *Proc. of the IEEE Int. Conf. on Robotics and Biomimetics*, pp. 560–565, 2009.

[2] L. Dominjon, A. Lécuyer, J.-M. Burkhardt, and S. Richir, "Development of a telerobotic system for exploration of hazardous environments," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2532–2537, 2004.

[3] F. Conti and O. Khatib, "Spanning large workspaces using small haptic devices," in *Proc. of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2005.

[4] J. M. Azorin, J. M. Sabater, L. Paya, and N. Garcia, "Kinematics correspondence & scaling issues in virtual telerobotics systems," in *Proc. of the World Automation Congress*, pp. 383 – 388, 2004.

[5] B. Horan and S. Nahavandi, "Intuitive haptic control surface for mobile robot motion control," in *Proc. of the IEEE Int. Workshop on Safety, Security and Rescue Robotics*, pp. 121–127, 2008.

[6] L. Dominjon, A. Lécuyer, J.-M. Burkhardt, G. Andrade-Barroso, and S. Richir, "The "bubble" technique: Interacting with large virtual environments using haptic devices with limited workspace," in *Proc. of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2005.

[7] P. Chotiprayanakul and D. Liu, "Workspace mapping and force control for small haptic device based robot teleoperation," in *Proc. of the IEEE Int. Conf. on Information and Automation*, pp. 1613–1618, 2009.

[8] E. Nuño, A. Rodríguez, L. Palomo, and L. Basañez, "A framework for teleoperators control," *Research in Computing Science. Special Issue in Electronics and Biomedical Engineering, Computer Science and Informatics*, vol. 35, pp. 21–30, 2008.

[9] J. Rosell, C. Vázquez, A. Pérez, and P. Iñiguez, "Motion planning for haptic guidance," *Journal of Intelligent and Robotic Systems*, vol. 53, no. 3, pp. 223–245, 2008.