applied sciences

MDPI

Article

Haptic Teleoperation of Impact Hammers in Underground Mining

Mauricio Correa, Daniel Cárdenas, Diego Carvajal and Javier Ruiz-del-Solar *

Advanced Mining Technology Center, Department of Electrical Engineering, Universidad de Chile, Santiago 8370451, Chile; mauricio.correa@amtc.cl (M.C.); daniel.cardenas@amtc.cl (D.C.); diego.carvajal@amtc.cl (D.C.)

* Correspondence: jruizd@ing.uchile.cl

Abstract: Impact hammers are used to reduce the size of blasted ore in mining operations. In underground mines, tele-operated impact hammers are used to reduce the size of boulders placed on the orepass's grizzlies. An impact hammer consists of a hydraulic arm with 4 degrees of freedom, powered with a hydraulic impact hammer as an end-effector. The tele-operation of impact hammers is difficult due to the latency of communications, the poor visibility of the environment, and the used 2D interfaces. This may result in a collision with the hammer and the infrastructure, idle strokes, and non-optimal operation. To address these issues, this paper proposes the haptic tele-operation of impact hammers. The proposed haptic tele-operation system is based on a 3D model of the environment, which is used to estimate repulsion forces that are transferred to the operator via a haptic device, so that the hammer does not collide with the structures of the mine. The system also allows identifying the oversized boulders deposited on the grizzly and notifying the operator every time the orepass is blocked, as well as providing different 3D views of the environment. A proof of concept is presented using a scaled setup, where it is validated that the use of the proposed system allows for providing a better and more efficient tele-operation experience.

Keywords: impact hammers; industrial robotics; haptic teleoperation; underground mining

check for updates

Citation: Correa, M.; Cárdenas, D.; Carvajal, D.; Ruiz-del-Solar, J. Haptic Teleoperation of Impact Hammers in Underground Mining. *Appl. Sci.* **2022**, *12*, 1428. https://doi.org/10.3390/app12031428

Academic Editors: António Paulo Moreira, Pedro Neto and Félix Vilariño

Received: 31 December 2021
Accepted: 24 January 2022
Published: 28 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Underground mines commonly use orepass systems as a safe and economic method to transport blasted ore, i.e., broken rocks, between production levels. In each orepass, a steel grate, called a grizzly, is used to control the size of the material entering the orepass. Material that is too large cannot pass through the grizzly and has to be broken using impact hammers. Impact hammers, also known as rock breakers, rock-breaking manipulators, rock-breaking hammers, or pedestal-mounted breaker booms, consist of an hydraulic arm with 4 degrees of freedom (rotation, lift, tilt, and breaker joints) powered with a hydraulic impact hammer as an end-effector. Figure 1 shows a diagram of the operating environment of an impact hammer. It shows a possible arrangement of sensors to obtain data from the material over the grizzly.

In modern mining facilities, impact hammers are usually tele-operated from a control room located in a safe place, away from the operation area. Under this schema, the operator actuates the electro-valves that control the hydraulic system using joysticks and buttons via a data network. To compensate for this lack of direct vision of the environment, cameras and video transmission equipment are often installed in the operation area. However, the data network and video stream encoding and decoding introduce a delay, or latency, between what the operator sees, and what is actually happening, which adds difficulties to tele-operation [1,2]. Due to the mentioned latency, the poor visibility of the environment, and the used 2D visualization interfaces, hammer tele-operation requires highly skilled operators that are trained to deal with these two phenomena. A typical operating environment can be observed in Figure 2. This figure shows a frontal loader dumping material onto the grizzly, and a tele-operated impact hammer fragmenting rocks.

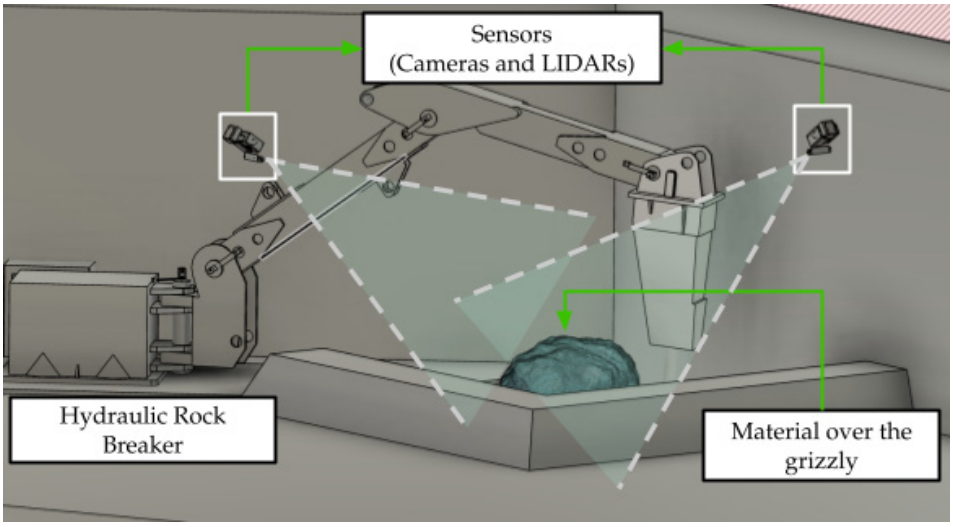


Figure 1. A rock breaking hammer environment with sensors, which includes: A hydraulic rock breaker, a grizzly with material, and sensors pointing to the working area.



Figure 2. Real impact hammer in operation, while material is dumped onto the grizzly.

Normally an operator handles more than one impact hammer, and due to the intermittent nature of this operation, sometimes the operator does not notice immediately that an orepass is being obstructed. This often causes the fragmentation process to take longer than it should because it is more difficult to clear a stoked orepass than a few rocks over the ore pass's grizzly, generating a delay in the mine's production chain. In addition, the described, standard tele-operation of impact hammers does not allow the operator to properly perceive the actions of the impact hammer in the environment because its sensors are limited to visual cameras that provide a 2D representation of the environment. This issue may cause damage to the impact hammer, for example, when the tool is activated without being in contact with the rock, which causes idle strokes in the air, or when the hammer collides with the environment (e.g., with the grizzly). In order to address these drawbacks, assistive tele-operation technology can be used for: (i) Enhancing the operator's perception by adding other sensing modalities (e.g., range sensors), (ii) notifying the operator every time that an orepass is blocked to have a shorter reaction time, and (iii) giving haptic feedback to the operator to avoid collisions of the hammer's end-effector

with the surrounding environment. Using these improvements, the operator could make better decisions that result in a more precise operation, extended lifetime of the equipment, and higher throughput of the mineral transport system.

There is very little literature related to the automation or tele-operation of impact hammers. According to [3], previous works concerning the automation or modernization of impact hammers are few. First attempts to automate impact hammers date back to 1998 [4], and the first tele-operated impact hammer was reported in 2000 [5].

Most of the reported work addressing tele-operation of impact hammers uses 2D cameras, and in some cases time-of-flight cameras (TOF) or stereo cameras. According to [3], TOF's low resolution is insufficient for this task, and stereo cameras obtain a single view of the scene. To the best of our knowledge, there are no previous works using 3D LIDAR (Laser Imaging Detection and Ranging) in this application, nor works reporting the use of haptic devices for the tele-operation of impact hammers.

Regarding the autonomous operation of impact hammers, the reported results are very poor. In [3], an average success rate of 34% is obtained in the task of breaking rocks, which does not make it possible for the application of this technology in a real mining environment. For this reason, we believe that a good intermediate step to full-automation, which improves the current technology used for the tele-operation of impact hammers, is haptic tele-operation.

In this context, this paper proposes the haptic tele-operation of impact hammers for improving the efficiency of the fragmentation process. The proposed haptic tele-operation system is based on a 3D model of the environment, which is used to estimate repulsion forces that are transferred to the operator via a haptic control device, so that the hammer does not collide with the structures of the mine. The repulsion forces are a function of the distances between the hammer's end-effector and the environment. The 3D model of the environment is built using point clouds acquired using range sensors, and updated continuously.

The proposed system also allows identifying the oversized boulders deposited on the grizzly and notifying the operator every time the ore pass is blocked. Moreover, thanks to the use of 3D sensors and cameras placed on opposite sides of the hammer (see Figure 1), it is possible to show the operator the environment from different viewpoints and a 3D model of the rocks to be crushed. With this information, the operator can make better decisions.

We do believe that the use of point clouds for improving tele-operation in mining applications will increase in the following years, thanks to the popularization of 3D scanning sensors, and the availability of libraries for processing this data (e.g., [6,7]). Field applications, as the one described here, will use this haptic technology.

In other industries, the use of point clouds (obtained using range sensors) for obtaining haptic feedback in tele-operation applications is not new, although most reported papers describe systems that operate in controlled or semi-controlled environments, which is not the case for underground mining. In ref. [8], a method of haptic feedback in real time, using streaming point clouds from RGB-D cameras, without using contact sensors and without preprocessing the data, is proposed. The concept of virtual world is used, in which the simulated robot can move freely. In this virtual world, the tip of the virtual effector or HIP (haptic interface point) is related to the movement desired by the user. This effector, in turn, is wrapped in a zone of influence called proxy, which is forbidden to interact with the point cloud. Therefore, if the effector tries to enter the point cloud (or be in a position very close to it), a difference between the position of the virtual effector or HIP with the center of the proxy will be produced. This difference in positions will result in a feedback force vector, which will cause the effector's motion to change direction. The work described in ref. [9] is based on the described feedback system, and introduces the concept of delay in tele-operation, where the motion is first performed in virtual space and then executed in the real world. This implementation and testing of the system were performed by teleoperation of a KUKA industrial robot, where the virtual world was created using point clouds obtained with an RGB-D camera. This work confirms that haptic feedbacks

increase the operator's accuracy by reducing the errors in the desired trajectory of the end effector, decreasing the collisions with the environment, and reducing the operation time. In ref. [10], a haptic system based on distances measured using a stereo camera, is evaluated on a robotic arm used to transport an object without colliding with obstacles. In ref. [11], a 3D virtual environment is recreated by a Kinect sensor used to calculate haptic forces based on potential fields. The paper evaluates the path generated when operating a robot arm with 7 degrees of freedom.

This paper is organized as follows. In Section 2, the proposed haptic tele-operation of impact hammers is described. Section 3 presents results, both in simulation and in reality, which are discussed in Section 4. Finally, in Section 5, conclusions of this work are drawn.

2. Haptic Teleoperation of Impact Hammers

2.1. Overall System Architecture

The proposed haptic tele-operation system is composed of eight sub-modules (see Figure 3). The *Point Cloud Server* acquires and merges the data from the two LIDARs, and segments it in three different point clouds, each one corresponding to measurements associated to reflections on the infrastructure, hammer, and material. These point clouds, and the acquired images, are sent to the *User Interface* where the different visualization are generated for the user. The *Collision Point Cloud Generation* module receives the point clouds and generates a so-called *collision point cloud*, which represents the points in the space to be avoided during tele-operation. The *Haptic Feedback Calculation* module estimates possible future collisions using the collision point cloud, the current commands on the end-effector (u_{eef}), and the state of the hammer's joints (x), and it generates haptic feedback information (f) for the haptic controller and haptic control interface. The *Haptic Controller* receives this feedback and generates the haptic force to be applied on the *Haptic Device* to alert the user of the limitation of the commands. At the same time, the *Haptic Control Interface* receives the control command on the end effector, which is the result of the operator force on the *Haptic Device* and the haptic force on the joystick, and the haptic feedback to limit the commands in case of any risky situation, for example, when the operator in spite of the haptic feedback decides to perform a risky task. Finally, these commands are converted from the Cartesian space (u_{eef}) to the joint space (u_{joints}) using the *Inverse Kinematics* of the machine to handle the articulation actuators directly.

2.2. Sensors

Sensors are located above the grizzly, on opposite sides, to generate complementary points of view of the working space (see Figure 1), and to provide a better representation of the material over the grizzly when both views are combined. In each side, a 3D LIDAR and camera are used. These sensors provide a streaming of point clouds and images. Point clouds are used to build a 3D model of the environment, and images are used for visualization purposes.

2.3. Point Cloud Server

This module first merges the point clouds acquired by the different 3D LIDARs into a single cloud. This point cloud is down sampled to generate a voxel grid representation of the space. Then, all the voxel grid points are separated in three different categories (see Figure 4):

1. Environment points that include all points that do not change in the operation of the system (e.g., point of the floor, points of the grizzly, and point over other parts of the structure of the mine). These points are used to build the 3D model of the environment;
2. Hammer points that correspond to the points generated by the LIDAR's reflections on the impact hammer, which are generated when certain parts of the hammer are positioned in the scanning area of the sensors;

3. Material points that correspond to any other point detected, which should correspond to the material over the grizzly.

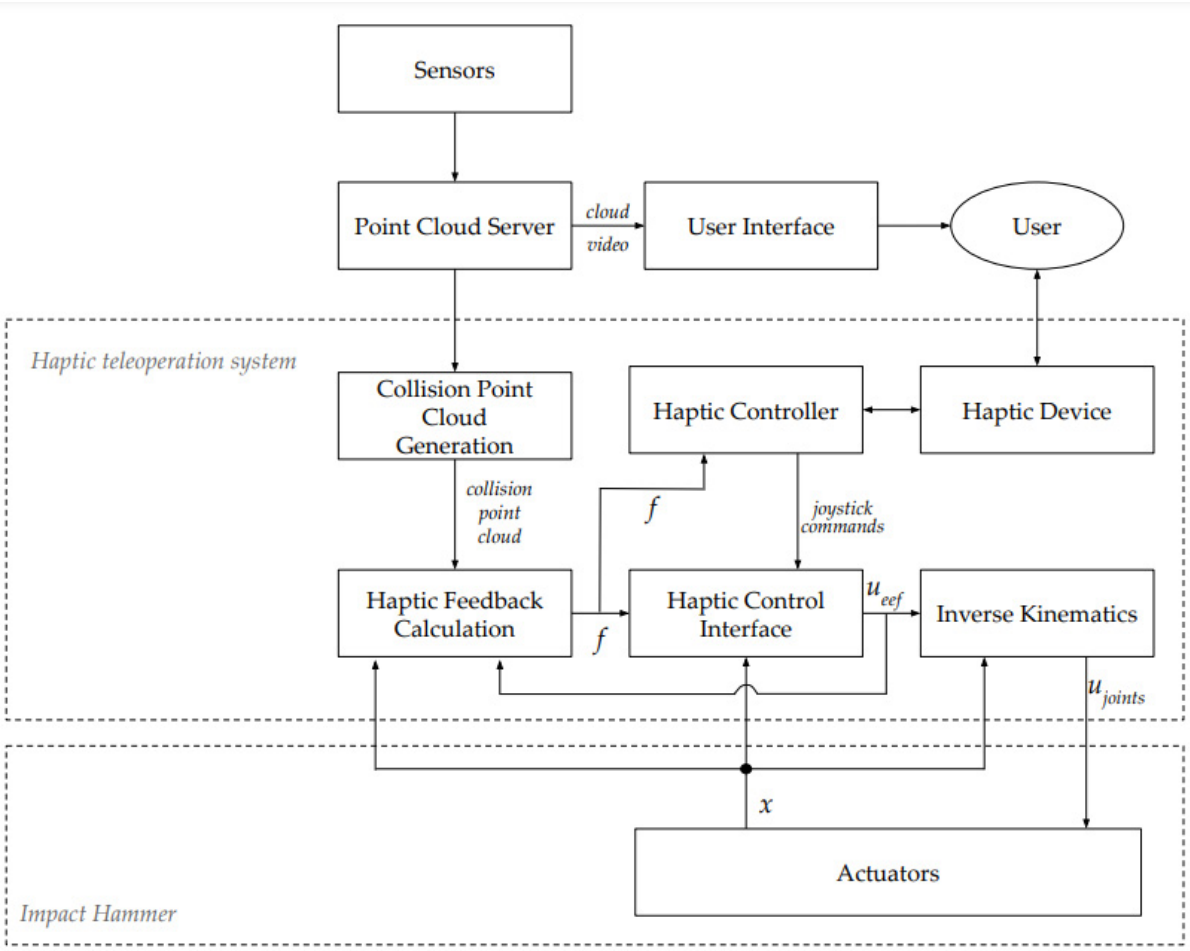


Figure 3. Overall view of the haptic tele-operation framework.

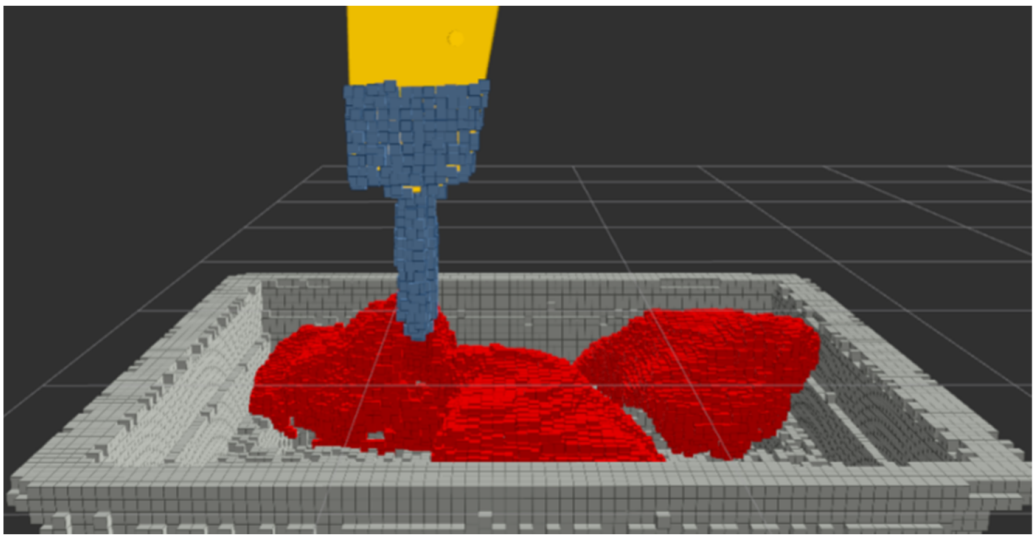


Figure 4. Point cloud categorization. Environment points (in gray) and material points (in red). Hammer points in blue, showed over the end-effector model (in yellow).

During the system's initialization, the environment model is generated by successively integrating point clouds from the environment, without considering the ones corresponding to material or the impact hammer. This results in a very dense point cloud that is first filtered for deleting isolated points that could be generated by noise on the sensors. Once the noise has been eliminated, the cloud is voxelized in 3D square voxels, reducing the number of point and maintaining the occupancy information of the environment in the space. Finally, this cloud is saved as the environment model. Figure 5 shows the main modules of the described process. Sensors 1 and 2 correspond to the 3D LIDARs used to capture the model of the environment through point clouds. The *Point Cloud Integration* module corresponds to the process of joining the point clouds of both sensors and performing the task of integrating these successive point clouds over time. The *Noise Filtering* module is responsible for removing noise. The *Voxelization & Occupancy Grid* module is in charge of voxelizing the point cloud in order to eliminate repeated points and to show the occupancy of the space. And finally, the *Save Model in Memory* module stores the resulting point cloud in memory for later use.

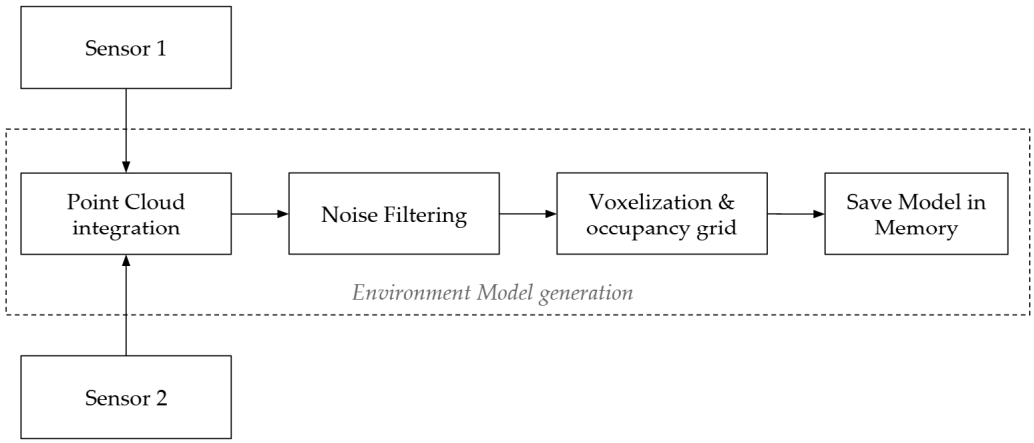


Figure 5. Processes to obtain the point cloud of the environment.

During operation, the environment model is used for determining the points that belong to the material model and hammer model. In each iteration, each new point cloud is voxelized and contrasted with the point cloud of the environment model. This comparison allows to subtract the points belonging to the environment. Once subtracted from the environment points, the resulting point cloud contain the hammer-points and material-points. Hence, to obtain the material-points, a new filtering process should be performed to subtract the hammer-points. This is done by first estimating the spatial region where the hammer is located, which is obtained using the current state of the hammer's joints and the kinematic model of the hammer. This way, all those points within this space will be classified as belonging to the hammer and therefore eliminated. Figure 6 shows the process used to obtain the point cloud of the material. Sensors 1 and 2 correspond to the 3D LIDARs used to capture the working environment. The *Point Cloud Concatenation* module is in charge of joining the point clouds coming from both sensors. The *Voxelization* module is in charge of voxelizing the point cloud resulting from the union of both sensors, thus eliminating repeated points. The *Environment Model Points* corresponds to the point cloud stored in memory, which only contains points corresponding to the grizzly. The *Subtraction of Environment Points* module compares the current point cloud with the point cloud stored in memory, and removes the points that are found in both models. Finally, the *Material Points* module is in charge of publishing the points resulting from the previous module, which correspond to the material on the grizzly.

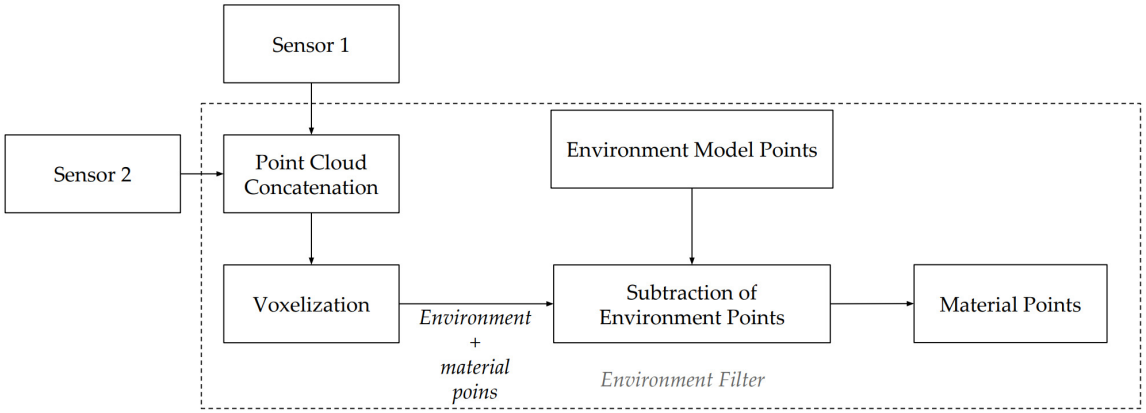


Figure 6. Processes to obtain the Point Cloud of the material.

2.4. Collision Point Cloud Generation

The generation of a collision point cloud has the goal of using it for avoiding collisions between the impact hammer and surrounding environment.

The collision point cloud always includes the environment point cloud, and the material cloud, depending on if the user wants to avoid the material or not. For instance, when the hammer's end-effector needs to be placed in the rock to be broken, the user chooses not to include the material cloud. This is done by just pressing a button on the haptic device.

The generation of collision points starts with a dilation operation over the environment point cloud, and applies over the material point cloud, which is implemented in each case using as structuring element cube of $3 \times 3 \times 3$ points. The dilation of the input point cloud by the cube of $3 \times 3 \times 3$ (which corresponds to a cube of 27 points), can be understood as the locus of the points covered by the cube of $3 \times 3 \times 3$ point when the center of it, moves within the input point cloud. The dilation operation is defined as follows:

$$A \bigoplus B = \cup_{b \in B} A_b \tag{1}$$

where A is the initial point cloud, and B is the structuring element (cube $3 \times 3 \times 3$).

This means that for each initial point, 26 extra points are created, which will surround the initial point (27 points in total). Figure 7 shows the dilation operation, and the result of this operation on a single voxelized point.

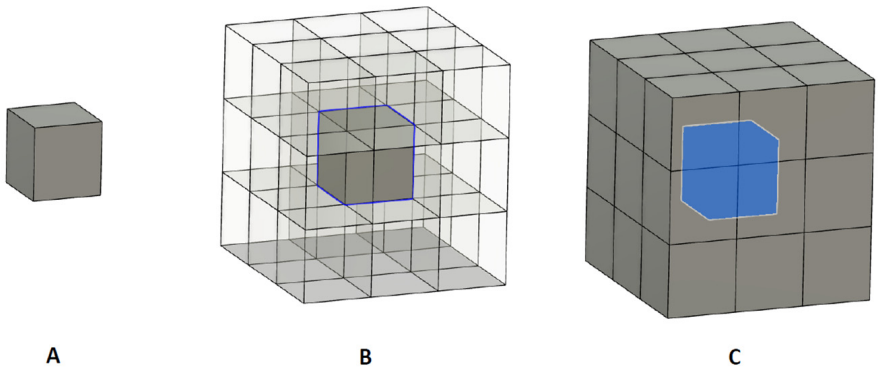


Figure 7. (**A**): Single voxelized point; (**B**): $3 \times 3 \times 3$ structuring element with origin in central point; and (**C**): result of dilatation operation.

At the end of the morphological dilation, a new point cloud is obtained, with multiple repeated points because the implemented dilation algorithm does not check if there is already a point in the space where the new points are generated. To eliminate this redundant

data and to increase the algorithm's performance, the resulting cloud is voxelized again. Figure 8 shows the process to obtain the collision points.

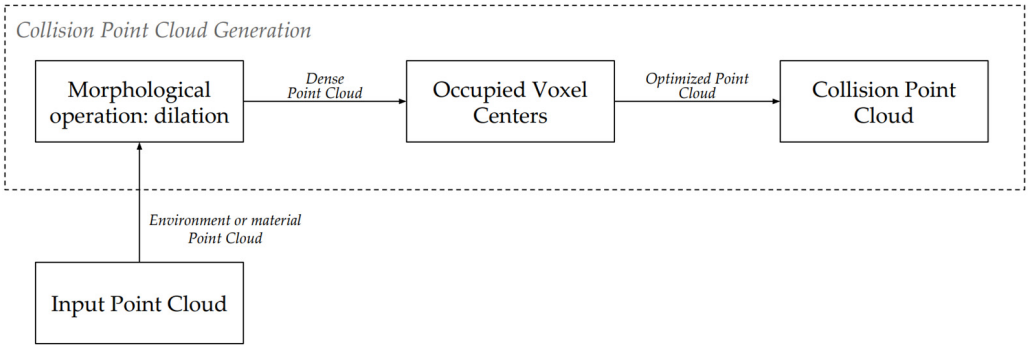
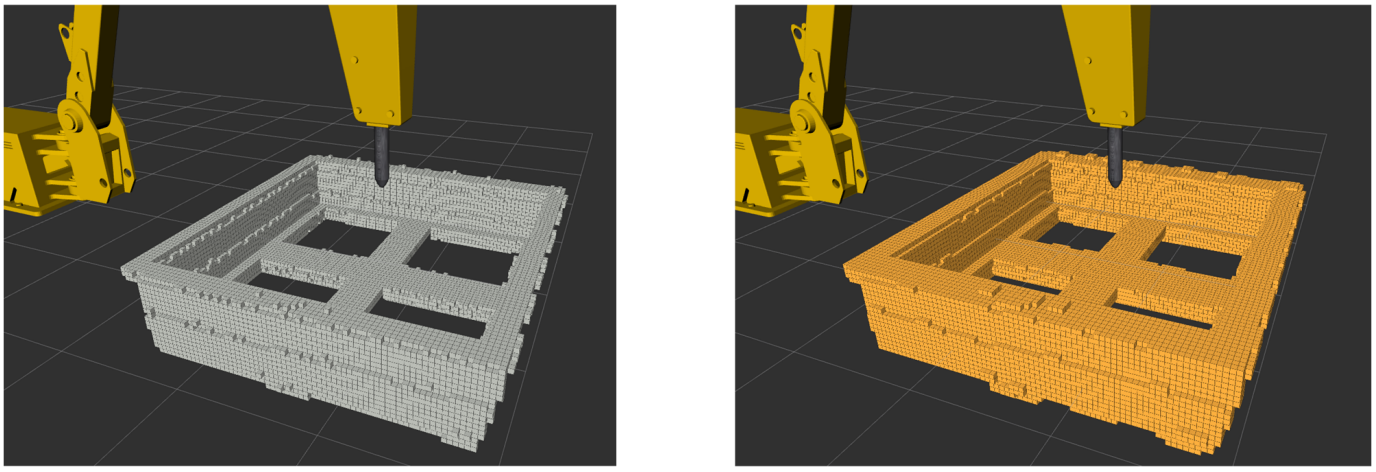


Figure 8. Processes to obtain the collision point cloud.

As mentioned, the collision point cloud is composed of two parts. The first part will come from the processing of the point cloud belonging to the environment model. As this model is static and was previously calculated in a previous module, it is only processed once to generate the cloud of collision points belonging to the environment. The second part will come from the processing of the point cloud belonging to the material. As this point cloud is dynamic (it is expected to change frequently), the process of obtaining the collision point cloud is performed in each execution cycle. In this case, for performance reasons, the dilatation operation is performed once per cycle.

In the case of requiring to generate a thicker collision point cloud, several dilation iterations should be performed in the same cycle. This would result in a larger safety distance between the real object and end effector. Figure 9 shows a comparison between the point cloud of the generated grizzly model and the point cloud resulting from the dilation process.



(**a**) (**b**)

Figure 9. (**a**) The original collision cloud generated with the environment points. (**b**) The resulting collision point cloud after applying the dilation process.

Therefore, the resulting collision cloud will be the concatenation of points from the static processing (performed only once at the beginning of the program) of the environment model, with that of the dynamic processing of the material (performed in each cycle). This point cloud will result in a more homogeneous cloud, filling the empty spaces due to the imperfection of the sensors.

It must be emphasized that at any moment, the user can choose using just the environment model for the generation of the collision point cloud, not including the point clouds associated to the material. This is normally made just before placing the hammer on the rock to be broken. The user makes the selection of this operation mode by just pressing a button in the haptic device.

2.5. Haptic Feedback Calculation

This module generates the haptic feedback that will restrict the movement of the joystick axes and the movement of the impact hammer. For this purpose, a so-called *virtual collision region* is generated around the end effector, which represents the space of the possible future positions of the end-effector, considering the direction in which it moves. The intersection of the virtual collision region with the collision point cloud will indicate possible collisions of the end-effector with the environment.

For simplicity, in this work two different geometries are used for representing the virtual collision region: A rectangular cuboid and a semi-sphere, which are positioned at the end-effector point that is located on the tip of the chisel. Figure 10 shows these geometries, which are oriented with respect to the direction of movement of the end-effector. In this simulation, the yellow arrow corresponds to the direction of motion of the end-effector.

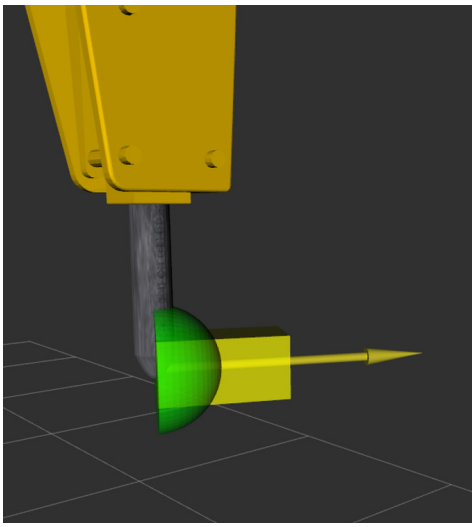


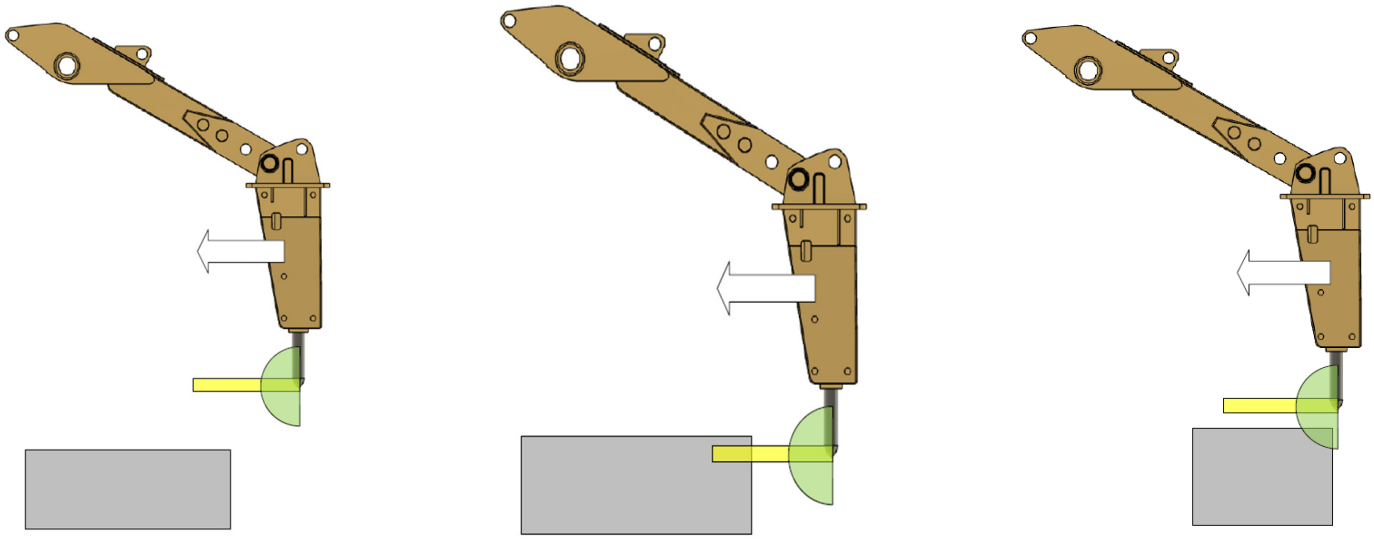
Figure 10. The yellow arrow shows the direction that the end-effector is moving. The yellow cuboid and green half-sphere are oriented according to the direction of motion of the effector.

As mentioned, the intersection of the virtual collision region with the collision point cloud will indicate how close is the effector to collide if the current direction and velocity of movement are maintained. The orientation of the virtual collision region has a direct relation to the direction of displacement of the end-effector, and the region it represents indicates the possible future positions in which the effector will find itself in future time steps.

The length of the cuboid is selected as the maximum distance that the effector will move before stopping (inertia of the system). The height and width of the cuboid are slightly larger than the height and width of the end-effector. These values make it possible to distinguish when the end-effector is in an apparent collision state with the collision point clouds, and therefore perform subsequent actions on the effector to avoid the collision. However, it could happen that a sudden movement in the actuation of the controls would produce an abrupt change on the movement direction of the end-effector, resulting in a collision that cannot be detected by the cuboid. To avoid this situation, the semi-sphere region will act as a safety region, detecting these particular cases, where the effector is not in a collision direction, but very close to the objects.

Thus, depending on the positions of the virtual collision region with respect to the collision point cloud, three different cases can be identified:

1. Rectangular cuboid and semi-sphere regions without intersection with the collision point cloud (see Figure 11a). In this case the impact hammer can continue its trajectory;
2. Rectangular cuboid intersects the collision point cloud (see Figure 11b). In this case, the velocity will be reduced according to the distance to the obstacle (see explanation in next sub section);
3. Semi-sphere region intersects the collision point cloud and the cuboid does not (see Figure 11c). In this case the velocity will be reduced to 50% (see explanation in next sub section).



(**a**) (**b**) (**c**)

Figure 11. (**a**) Rectangular cuboid and semi-sphere regions without intersection with a collision point cloud. (**b**) Rectangular cuboid intersects the collision point cloud. (**c**) Semi-sphere region intersects the collision point cloud and the cuboid does not. The yellow rectangle represents the cuboid. The green semi-circle represents the semi-sphere. The white arrow corresponds to the hammer direction.

The output of haptic feedback (f) has three values: A Boolean indicating if the cuboid intersects the collision point cloud or not, a Boolean value indicating if the semi-sphere intersects the collision point cloud or not, and a normalized value that indicates a level of safety of executing a command.

The level of safety parameter (S) is calculated using the distance from the end-effector to the collision point cloud ($Dist_{EP}$), as the percentage of the cuboid region lying outside the intersection with the collision point cloud. To calculate this percentage, a max distance ($Dist_{MAX}$) is predefined. This distance is considered as a deceleration distance by the end effector when it moves with the maximal allowable speed and it stops. Thus, S is calculated as:

$$S = min\left\{\frac{Dist_{EP}}{Dist_{Max}}, 1\right\}. \quad (2)$$

2.6. Haptic Control Interface

This module limits the end-effector's actuation orders. Depending on the distances between the virtual collision region and collision point cloud, the maximum allowed speed will change, so that the effector does not collide and is not damaged. According to the cases defined in the former sub section:

1. Rectangular cuboid and semi-sphere regions without intersection with the collision point cloud. In this case no constraint action is performed, the end effector can reach maximum velocity;
2. Rectangular cuboid intersects the collision point cloud. In this case, the resulting end-effector velocity ($u_{eef\ limited}$) will be reduced depending on the level of safety parameter as:

$$u_{eef\ limited} = u_{eef} \cdot S^2. \quad (3)$$

The use of the square of S is inspired by the fact that the kinetic energy depends on the square of the speed.

Thus, the magnitude of the commands on the end-effector will decrease its maximum allowable speed as it gets closer to the collision point-cloud, arriving with velocity tending to zero as it touches the collision point cloud.

3. Semi-sphere region intersects the collision point cloud and the cuboid does not. In this case the end effector speed is limited to 50% of the maximum speed. In other words, if a higher effector speed is attempted under this condition, the system will automatically limit the speed. In case the effector movement is attempted at a lower speed, the system will not make any adjustment to the speed.

2.7. Inverse Kinematics

This module adapts desired velocities on the end-effector to the corresponding joint velocities. To do this task, the Jacobian pseudo inverse of the kinematic model is calculated using the current state of the machine to perform this conversion as:

$$u_{joints} = J_{pseudo\ inv} \cdot u_{eef} \quad (4)$$

where u_{eef} corresponds to a vector with the velocities on some axis of the Cartesian space, {x, y, z, pitch} in this case; u_{joints} corresponds to the speed on each articulation on the machine {q0, q1, q2, q3} related to its respective joint, and $J_{pseudo\ inv}$ corresponds to the pseudo-inverse of the Jacobian of the kinematic model of the arm, which is shown in Figure 12.

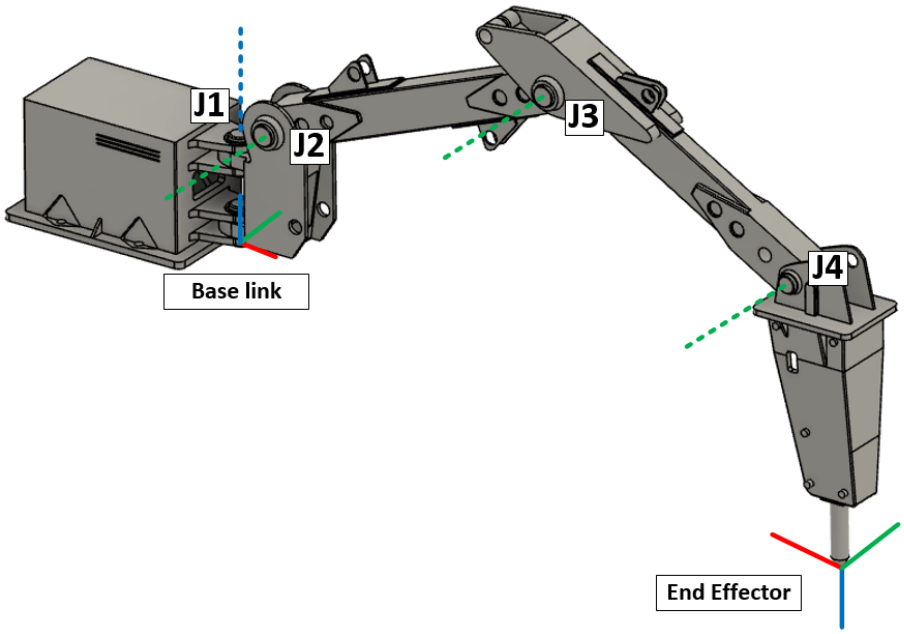


Figure 12. Simulated kinematic model of the impact hammer. J1, J2, J3, and J4 correspond to the hammer joints. The *X*-axis is shown in red, the *Y*-axis in green, and the *Z*-axis in blue.

The pseudo inverse method is implemented to reduce the dimension of the Jacobian, which is originally declared in Cartesian space {x, y, z, roll, pitch, and yaw}. The kinematic model of impact hammer is generated by a kinematic tool (*moveit*) and allows for simplifying the number of the states on the Cartesian space to {x, y, z, pith}.

2.8. Haptic Controller

The haptic control has the task of alerting the user of possible collisions using the haptic device. The form of warning that has been implemented is to restrict the range of movement of haptic device (control axes). In this way, as the effector gets closer to collision, the restriction to the movement of the axes will be greater, exerting a counter force to the user if they want to pass the calculated limits.

Therefore, the restriction of movements to the axes of the haptic device will allow on the one hand to alert the user of possible collisions (and how close the hammer is to collision), and on the other hand, it will be able to slow down the hammer by limiting the possible movement actions.

In the same way as the haptic control interface, the limitation sent to the haptic controller depends on three states of haptic operation:

Rectangular cuboid and semi-sphere regions without intersection with the collision point cloud: The haptic control does not exert any counteracting force to the operator's actuation. In this case, the user will not feel any restriction on the movement of any of the control axes;

4. Rectangular cuboid intersects the collision point cloud: In this case the movement of each axis is limited, depending on the remaining distance to the object to collide. The axel movement limitation is:

$$Axis_{\mathrm{max}possible\ pose} = Dist_{object} \cdot \left(Axis_{\mathrm{max}pose} - Axis_{\mathrm{init}pose} \right) + Axis_{\mathrm{init}pose}. \quad (5)$$

Thus, as the effector approaches the object to collide (distance in range {0,1}), the control will limit the movement of its axes, opposing the user's action. In case the user operates under the calculated limit, he/she will not feel the movement limitations of the control.

5. Semi-sphere region intersects the collision point cloud and the cuboid does not. When this condition is activated, the movement is limited to 50%, which means that the operator will feel that the control only allows them to move the axes, normally the first half in each axis. When trying to exceed the second half, the haptic control will exert a repulsive force opposite to the operator's direction.

2.9. User Interface

The user interface is used to provide the operator with multiple perspectives of the grid material. Here the operator can move the point cloud presented to him, and adjust the view to their liking, in order to improve the perception of the objects within the hammer workspace. The visual output of this interface is in the form of a visualization of the point clouds, previously processed by the modules described above. Figure 13 shows the interface implemented in this project. The interface shows the model of the UR5 robot, which can be rotated as desired by the user of the interface, and the cloud of collision points (in yellow). This interface includes visual signals to inform the operator of the current operation conditions.

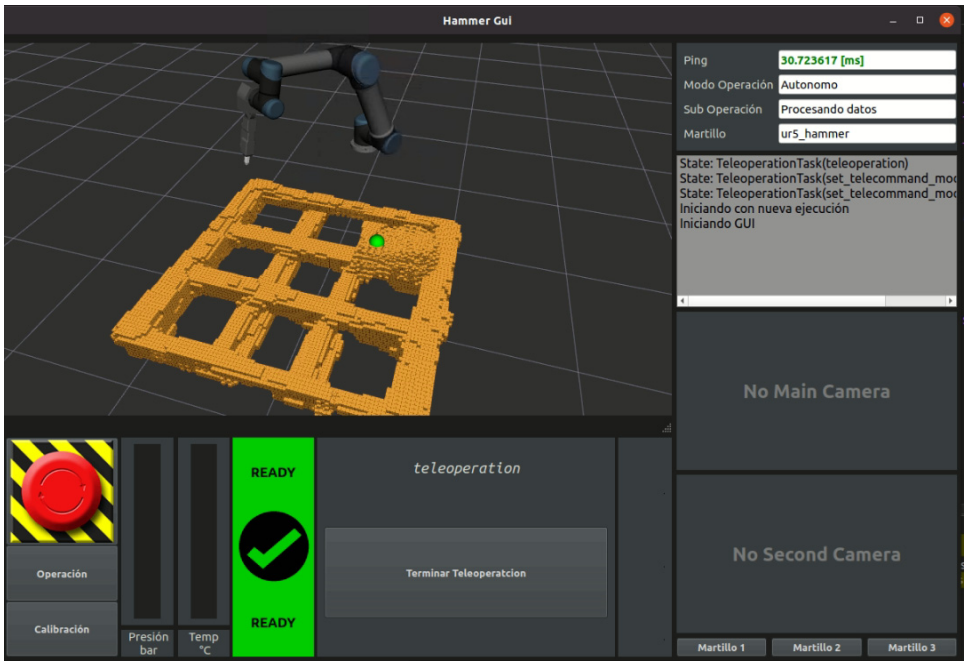


Figure 13. User interface. The yellow shows the visual output of the processing of the point clouds obtained by the sensors.

3. Results

3.1. Experimental Setup

As a proof of concept, we built a realistic laboratory setup. The testing environment used for validation consists of a real, scaled grizzly with real rocks. The scales of both elements are in relation of 1:3. For the impact hammer, an UR5 robotic arm was used with a scale of 1:5, just as a real impact hammer. For having a similar kinematic model rather than a real rock breaker, two articulations of the UR5 were fixed, and the other four were tele-operated (see Figure 14). The UR5 robot was mounted, related to the grizzly, in the configuration used in a real mining operation. Figure 15 shows how the UR5 robot was installed for the experimental tests. Next to it, there is a metallic test grizzly to scale, with material (rocks) on it.

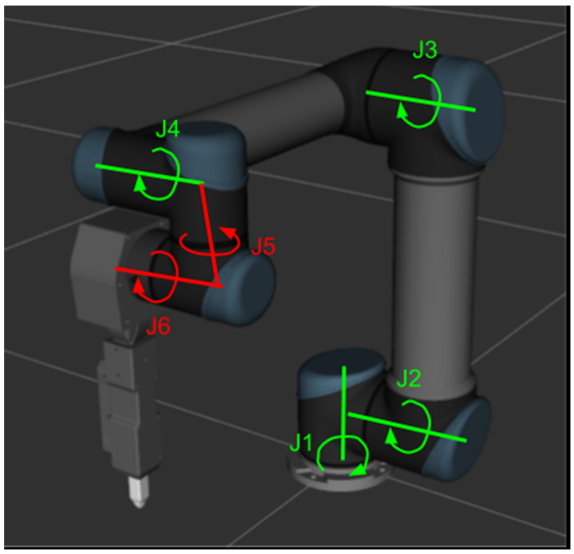


Figure 14. Articulated model of the adapted UR5. The green joints (J1, J2, J3, and J4) correspond to the articulated joints. The red (J5 and J6) correspond to the fixed joint of the kinematic model.



Figure 15. Experimental operation of the rock breaker set-up.

Two mounting-bases, each one containing a camera and a 3D LIDAR, are placed around the operation environment. This allows for the sensors to be located above the operational set-up, pointing to the grizzly and covering it completely (see Figure 16). In this work, Livox model MID-40 3D LIDARS and Arecont model AV5225 PMIR visible spectrum cameras are used.

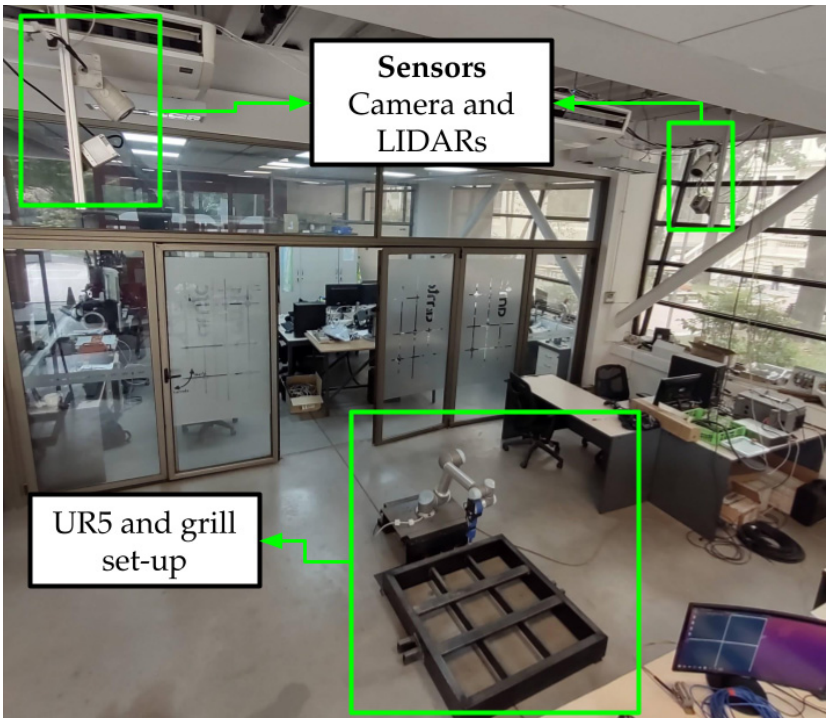


Figure 16. Experimental set-up.

To test the haptic part, 2 Brunner CLS-E controls were used, which allowed up to four joints to be controlled simultaneously (each joystick has 2 degrees of freedom). For convenience, these were installed on a chair used for the tele-operation. Figure 17 shows an operator using the UR5 robot, the Brunner haptic controls (installed on a chair), and the user interface.



Figure 17. User operating the UR5 arm, using the haptic controls (installed on a chair), and the user interface.

3.2. System Configurations to Be Tested

The main purpose of the reported experiments is to validate that the haptic system facilitates the control of the impact hammer to perform the task of positioning the tip of the end-effector above the rock, without colliding. That means that it improves the tele-operation experience.

Several tests are recorded doing this task with different configurations of the haptic system in order to study the impact of this feature. In particular, three different configurations of the system are tested. The configuration combines a different feature of the haptic system, which are:

1. Type of control: Determine how the machine will control the arm (impact hammer). In normal tele-operation, the user directly controls each joint of the arm. In the proposed system, the user directly controls the pose of the end-effector;
2. Use of the 3D reconstruction of the environment: This feature considers the use of the 3D reconstruction to get a multi-view of the operational environment. In normal tele-operation of the impact hammer, just one or two views are used, which are obtained directly by the cameras;
3. Use of haptic feedback: This feature will give the operator the feedback of the joystick to avoid colliding with the grill or other rocks

Finally, these features are combined in different configurations, A to D, which are defined in Table 1.

Table 1. Types of system's configurations used in tests.

Configuration Id	Type of Control	Use 3D Reconstruction	Use Haptic Feedback
Configuration A	Joints	No	No
Configuration B	End effector	No	No
Configuration C	End effector	Yes	No
Configuration D	End effector	Yes	Yes

3.3. Tasks

Predefined tasks are elaborated to generate a set of tests that can be replicated by different operators. This way, a predefined task contains the following information:

1. The initial state of the arm;
2. The target position;
3. A collision point cloud: The same collision space will be used in the same task. This way the operator has the same level of difficulty for avoiding the grizzly and rocks.

The experiment consists of generating the predefined environment test and giving the operator a target position. The operator has to move the arm using the different configurations of the system. An example of a pre-defined task is shown in Figure 18.

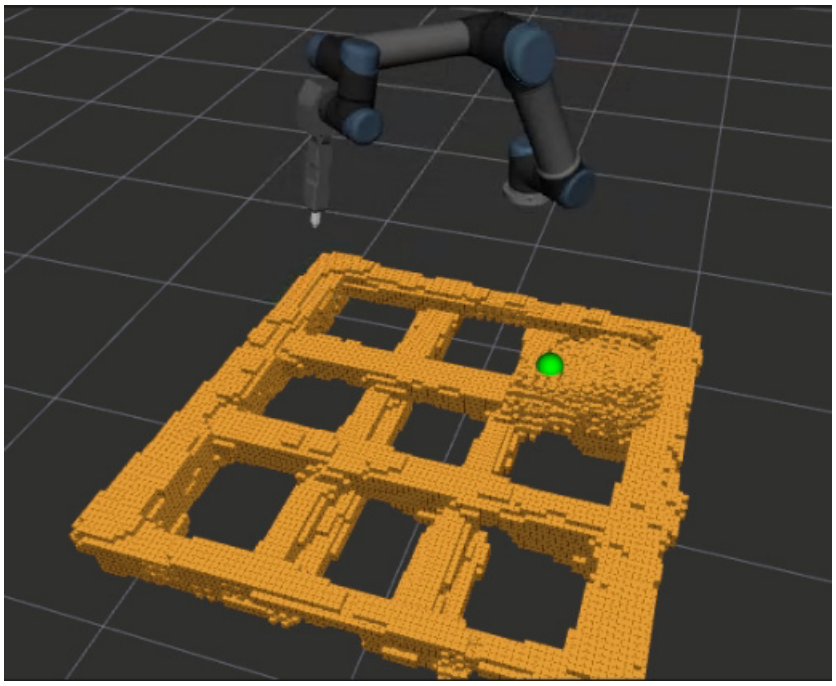


Figure 18. A pre-defined experimental test (Task 1): The arm tip is located out of the grill. The avoiding environment (the point cloud in boxes) consists of a single rock over the grill. The green circle corresponds to the target goal.

3.4. Experimental Results

Each configuration was evaluated using two different tasks, executed by two different non-expert operators. In total, 20 trials were carried out. The obtained results are shown in Table 2.

Table 2. Table with metrics of performance of different tasks with different configurations.

Task and Configuration Id	Average Success Rate	Average Reaching Time [s]	Average Tip Path Length [m]
Task 1, conf. A	0%	—	—
Task 1, conf. B	25%	28.66	0.947
Task 1, conf. C	100%	20.07	1.005
Task 1, conf. D	100%	10.70	0.633
Task 2, conf. A	25%	47.16	1.920
Task 2, conf. B	50%	37.80	1.077
Task 2, conf. C	80%	34.45	1.250
Task 2, conf. D	100%	24.71	1.080

A task is considered successful if the operator was able move the end effector to the target position with a precision of at least 0.015 [m], without colliding with the grizzly or rocks. If the operator cannot reach the target position, the task is considered a failure.

Table 2 shows the average success rate, representing the percentage of successful tasks. The average reaching time shows how many seconds it takes for the operator to reach the target. The average tip path length represents the distance in meters that the tip of the end-effector has to move to reach the desired target position. It can be observed that the success rate increases as more features are integrated into the system. In Task 1 conf. A, the operators were not able to achieve the target point without colliding with the rocks. The integration of more features allows the operator to complete the task and reduce collisions. Configuration D allows complete completion of the task without collisions because the haptic system does not allow commands that generate collisions to be executed.

In order to illustrate how the system operates, Figure 19 shows the trajectories of the tip for a different configuration when executing task 2. It can be seen that the trajectory becomes simpler as more features of the system are added.

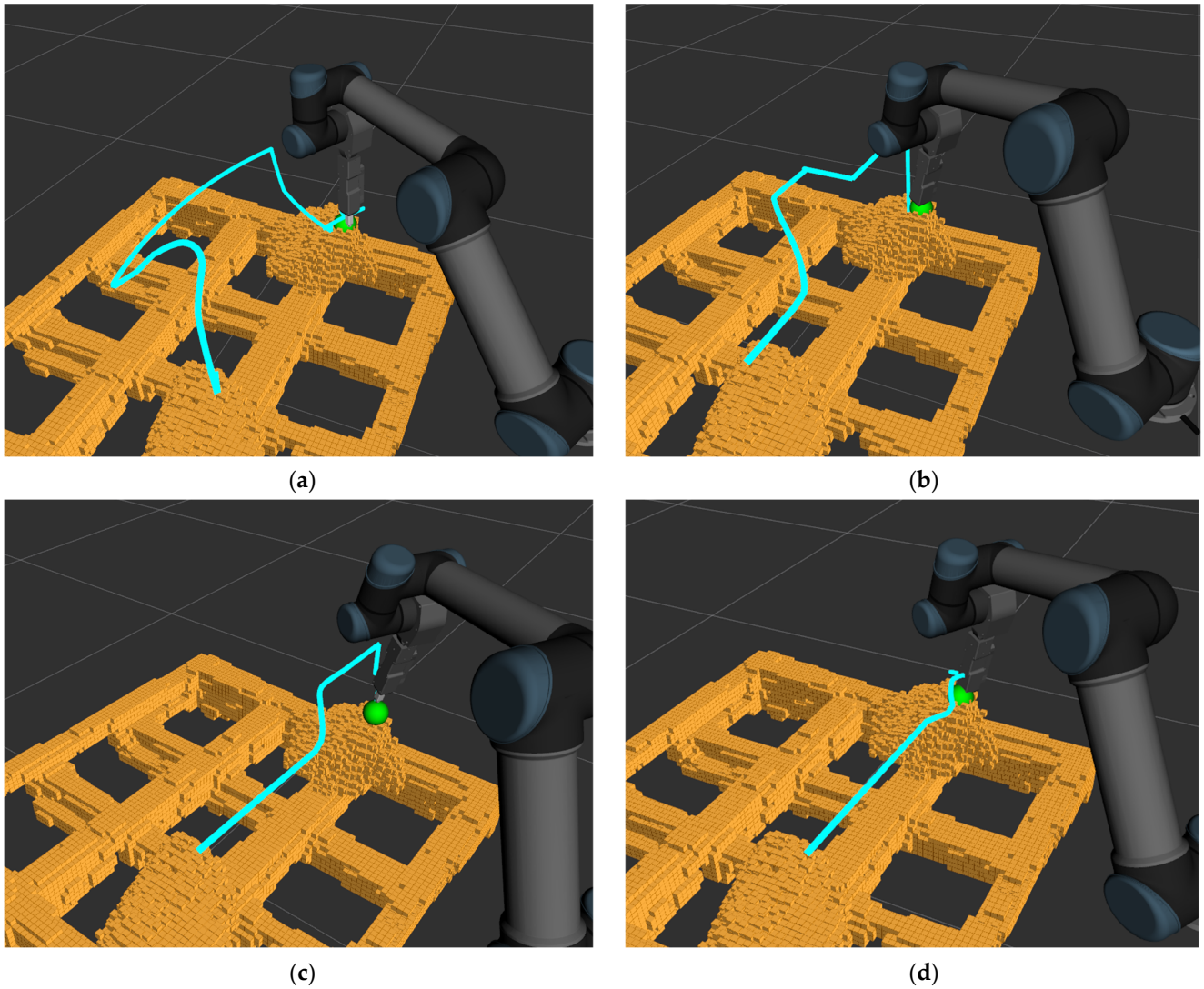


Figure 19. Different hammer tip trajectory paths of task 2 (same operator). The path is on a cyan line. (**a**) Conf. A (user controls each joint of the arm). (**b**) Conf. B (user controls the pose of the end-effector). (**c**) Conf. C (user controls the pose of the end-effector and 3D reconstruction of the environment) (**d**) Conf. D (user controls the pose of the end-effector, 3D reconstruction of the environment, and haptic feedback).

4. Discussion

The reported experiments show that the developed haptic tele-operation system allows for improving the tele-operation experience of the user. The processing of the different point clouds acquired using 3D LIDARs manages to correctly separate the rocks from the grizzly. The haptic controller and control interface manage to provide enough force feedback to the operator to get their attention, so that they can perceive elements close to the end-effector, thus increasing the perception of the environment in the operating area. In addition, the user interface improves the visual perception of the elements, improving the control over the robot, even if the haptic feedback part is not used.

The results depicted in Table 2 allow for comparing the different configurations, and in this way the different features that the proposed haptic tele-operation system includes.

When comparing configurations, A and B, a significant improvement of the performance of the task is observed when the end-effector controller is used (configuration B). The main reason for this, is that controlling an arm by actuating directly the joints is a difficult task that requires much preparation and practice. This way, controlling the end-effector makes the operation more affordable for new users.

When comparing configurations B and C, it can be observed a slight improvement in the case of using a 3D model of the environment (configuration C). The reason for this is that in case B, the operator cannot change the perspective, and therefore, in some cases it is not possible to have a proper visualization to avoid colliding. When the operator uses configuration C, they have the possibility of changing the viewpoint and in this way avoid collisions.

Finally, when comparing configurations C and D, a more notorious performance improvement is detected in the case of using haptic feedback. The reason for that is that the operator does not have to worry about colliding with the environment or rocks. This produces that the operator is able to increase the speed, and can move the arm without worrying about collisions.

By comparing the different configurations using the hammer tip trajectories (see Figure 19), it is possible to visualize the effect of each configuration on the end effector path. In this aspect, configuration A and B generate a larger deviation when approaching the target, due to the poor perspective that make it more comfortable to first raise the end-effector and then approaching the target from a higher position in order to get a better perspective. In contrast, configurations C and D generate a path that approach the end-effector directly to the target, as they are able to manage the perspective, but then finish with different maneuvers of positioning. In particular, configuration D presents a slight rise over the rock when approaching as a result of haptic feedback that allow the operator to perform fine maneuvers more easily, unlike configuration C which tends to perform wider maneuvers at the final stage of the positioning process.

Similar to [9–11], the results show that using a haptic system improves the performance and safety of performing a certain task using a robot manipulator. In addition to analyzing the number of collisions, our study evaluates the level of precision of the task being executed, and this type of consideration is not studied in other works and it is relevant for the context of this type of task in particular, which is positioning the chisel for breaking rock. In addition, our system uses the virtual environment to generate multi-perceptive views to aid in performing a task. This feature is not analyzed in the other studies describing haptic tele-operation based on point-clouds [9–11].

It should also be noted that although the development is intended for industrial environments with a real impact hammer, a small-scale robotic arm was used for the laboratory tests, which implies different behaviors due to the different construction of these elements (electric vs. hydraulic). This may imply that for final implementation, major adjustments may be needed for the control and haptic feedback modules of the system.

5. Conclusions

This paper addressed the haptic tele-operation of impact hammers. The proposed haptic tele-operation system is based on a 3D model of the environment, which is used to estimate repulsion forces that are transferred to the operator via a haptic device, so that the hammer does not collide with the structures of the mine. The system also allows for identifying the oversized boulders deposited on the grizzly, notifying the operator every time the orepass is blocked, as well as providing them with different 3D views of the environment.

A proof of concept was presented using a scaled setup. The reported experiments show that the use of a 3D model of the environment and the use of haptic feedback improves the tele-operation experience. When using the proposed system, operators were able to increase the success rate of the tele-operation task, and at the same time to reduce the completion time and the length of the path that the end-effector must follow.

As part of our future work, we will validate the tele-operation system using a real impact hammer operating in a real production environment of an underground mine. In that environment, it is expected that several aspects of the operation will emerge which are not reproducible or were not considered in our laboratory environment. The analysis of these new experiments allows for improving and adapting the haptic feedback algorithm to be applied in production systems. In addition, the system will be validated with expert rock breaker operators in order to be able to quantify the real operational impact of its use. Given that a professional operator has a higher level of expertise in controlling the hammer with the current limitations of traditional systems, the performance improvement could be different in relation to a non-expert operator.

Further improvement will focus on the use of haptic feedback to develop assistive tele-operation to guide the operator to follow optimal trajectories. Assistive tele-operation requires defining hammer tip trajectories for successful positioning and providing haptic feedback when the operator uses commands that move the hammer tip away from these trajectories.

Author Contributions: Conceptualization, J.R.-d.-S. and M.C.; methodology, J.R.-d.-S., D.C. (Daniel Cárdenas), D.C. (Diego Carvajal) and M.C.; software, D.C. (Daniel Cárdenas) and D.C. (Diego Carvajal); validation, D.C. (Daniel Cárdenas) and D.C. (Diego Carvajal); formal analysis, J.R.-d.-S., D.C. (Daniel Cárdenas), D.C. (Diego Carvajal) and M.C.; investigation, J.R.-d.-S., D.C. (Daniel Cárdenas), D.C. (Diego Carvajal) and M.C.; resources, J.R.-d.-S. and M.C.; data curation, D.C. (Daniel Cárdenas) and D.C. (Diego Carvajal); writing—review and editing, J.R.-d.-S., D.C. (Daniel Cárdenas), D.C. (Diego Carvajal) and M.C.; supervision, J.R.-d.-S.; project administration, M.C.; funding acquisition, J.R.-d.-S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Chilean National Research Agency ANID under project grant Basal AFB180004 and FONDECYT 1201170.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study is contained in the article itself.

Acknowledgments: We thank Isao Parra, Mauricio Mascaro, Francisco Leiva, and Patricio Loncomilla for their valuable discussions and support.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cermak, G. Multimedia Quality as a Function of Bandwidth, Packet Loss, and Latency. *Int. J. Speech Technol.* **2005**, *8*, 259–270. [CrossRef]
2. Ohm, J.; Sullivan, G.J.; Schwarz, H.; Tan, T.K.; Wiegand, T. Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding (HEVC). In *IEEE Transactions on Circuits and Systems for Video Technology*; IEEE: New York, NY, USA, 2012; Volume 22, pp. 1669–1684. [CrossRef]

3. Lampinen, S.; Niu, L.; Hulttinen, L.; Niemi, J.; Mattila, J. Autonomous robotic rock breaking using a real-time 3D visual perception system. *J. Field Robot.* **2021**, *38*, 980–1006. [CrossRef]
4. Takahashi, H.; Sano, K. Automatic detection and breaking system for boulders by use of ccd camera and laser pointer. *Fragblast* **1998**, *2*, 397–414. [CrossRef]
5. Hubert, G.; Dirdjosuwondo, S.; Plaisance, R.; Thomas, L. Teleoperation at freeport to reduce wet muck hazards. *MassMin* **2000**, *2000*, 173–179.
6. Rusu, R.B.; Cousins, S. 3D is here: Point Cloud Library (PCL). In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 1–4. [CrossRef]
7. Han, X.-F.; Jin, J.; Wang, M.-J.; Jiang, W. Guided 3D point cloud filtering. *Multimed. Tools Appl.* **2018**, *77*, 17397–17411. [CrossRef]
8. Ryden, F.; Chizeck, H. A Proxy Method for Real-Time 3-DOF Haptic Rendering of Streaming Point Cloud Data. *IEEE Trans. Haptics* **2013**, *6*, 257–267. [CrossRef] [PubMed]
9. Valenzuela-Urrutia, D.; Muñoz-Riffo, R.; Ruiz-del-Solar, J. Virtual Reality-Based Time-Delayed Haptic Teleoperation Using Point Cloud Data. *J. Intell. Robot. Syst.* **2019**, *96*, 387–400. [CrossRef]
10. Tang, X.; Zhao, D.; Yamada, H.; Ni, T. Haptic interaction in tele-operation control system of construction robot based on virtual reality. In Proceedings of the 2009 International Conference on Mechatronics and Automation, Changchun, China, 9–12 August 2009; IEEE: New York, NY, USA, 2009; pp. 78–83.
11. Xu, X.; Song, A.; Ni, D.; Li, H.; Xiong, P.; Zhu, C. Visual-haptic aid teleoperation based on 3-D environment modeling and updating. *IEEE Trans. Ind. Electron.* **2016**, *63*, 6419–6428. [CrossRef]