Motion Strategies for a Cobot in a Context of Intermittent Haptic Interface

Vamsikrishna Guda, Stanley Mugisha, Christine Chevallereau, Matteo Zoppi, Rezia Molfino, Damien Chablat

Vamsikrishna Guda
Laboratoire des Sciences du Numérique de Nantes, UMR CNRS 6004,
1 rue de la Noë, 44321 Nantes
Emails: Vamsikrishna.Guda@ls2n.fr

Stanley Mugisha
PMAR Robotics Group, DIMEC
University of Genoa,
Via Opera pia 15/A,16145 , Genova, Italy
Emails: Stanley.Mugisha@edu.unige.it

Christine Chevallereau
Laboratoire des Sciences du Numérique de Nantes, UMR CNRS 6004,
1 rue de la Noë, 44321 Nantes
Emails: Christine.Chevallereau@ls2n.fr

Matteo Zoppi
PMAR Robotics Group, DIMEC
University of Genoa,
Via Opera pia 15/A,16145 , Genova, Italy
Emails: Matteo.Zoppi@unige.it

Rezia Molfino
PMAR Robotics Group, DIMEC
University of Genoa,
Via Opera pia 15/A, 16145 ,Genova, Italy
Emails: Rezia.Molfino@unige.it

Damien Chablat
Laboratoire des Sciences du Numérique de Nantes, UMR CNRS 6004,
1 rue de la Noë, 44321 Nantes
Emails: Damien.chablat@cnrs.fr

Motion Strategies for a Cobot in a Context of Intermittent Haptic Interface

From the list of interfaces used in virtual reality systems, haptic interfaces allow users to touch a virtual world with their hands. Traditionally, the user's hand moves the end-effector of a robotic arm. When there is no contact in the virtual world, the robotic arm is passive; when there is contact, the arm suppresses mobility to the user's hand in certain directions. Unfortunately, the passive mode is never completely seamless to the user. Haptic interfaces with intermittent contacts are interfaces using industrial robots that move towards the user when contact needs to be made. As the user is immersed in the virtual world via a virtual reality Head Mounted Display (HMD), he cannot perceive the danger of a collision when he changes his area of interest in the virtual environment. The objective of this article is to describe four movement strategies for the robot to be as fast as possible on the contact zone while guaranteeing safety. This work uses the concept of predicting the user's intention through his gaze direction and the position of his dominant hand (the one touching the object) and safe-points outside the human workspace. Experiments are done and analyzed with a Pareto front with a UR5 robot, an HTC vive tracker system for an industrial application involving the analysis of materials in the interior of a car.

1 Introduction

Virtual Reality (VR) aims to immerse a human being in a virtual environment using all his senses. In most collaborative systems, the main senses are sight, then hearing, and finally, touch [6]. The sense of vision can be rendered by using large screens that occupy the user's entire field of vision or by using a Head Mounted Display (HMD). In the latter case, the user's vision becomes completely disconnected from the real world and all his movements can become dangerous. In some cases, user may lose his spatial landmarks and have the feeling of falling on the ground. By using immersion HMD and headphones, the user can free himself from his environment. Sound immersion further increases this immersion and separation from the real world.

Haptic interfaces, such as Virtuose 6DOF [25], are used in product design by engineers [18]. In [4,12], a five-fingered haptic interface robot with a 6 degree of freedom (DOF) arm and a 15 DOF hand was used to provide multipoint contact between the user and a virtual environment through force and tactile feeling to the fingertips of the human hand. These interfaces are safe and well mastered but if the user can apply force/torque, he cannot feel the textures and appreciate the quality of the materials. Among the main shortcomings of these interfaces are limited workspace, low stiffness and high cost.

New haptic interfaces using an industrial robot or a cobot (robots specially designed to work in human-robot environments) can be used as haptic interfaces with intermittent contacts [2,13]. For the application envisaged in this document, the cobot carries several texture specimens on its end-effector, to allow contact between a user's finger and the robot. They are called Intermittent Contact Interfaces (ICI) [15].

When the user uses HMD vision interfaces and has to perform haptic evaluations, he no longer sees the real scene, but only a virtual world. His physical reference points quickly disappear except for objects he touches such as his seat and the floor.

When users reach to grasp objects, they look at the target first, then bring the hand to the center of gaze to grasp the object. Eye-hand coordination is a fundamental behavior that humans use to interact with the world [9,11,20,21]. The head movement facilitates subsequent gaze shifts toward the future position of the hand to guide object manipulations, thus leading to a strong correlation between head and hand movement parameters [23,28,29].

Through the user's gaze and hand movements, and the position of areas to be studied, it is possible to predict the tasks that the user will perform. The study aims to ensure that the robot end-effector will be available for intermittent contact in complete safety when the human hand is close to the surface to touch.

The outline of this article is as follows. First, we present the context of the study and the material used. Then, a human-robot interaction framework is introduced with its hardware, the virtual environment and its data flow. Next, we present two velocity profiles to ensure user safety and improve the performance of our system by introducing safe-points. Four strategies are then intro-

The original version of this paper has been accepted for presentation at the ASME 2021 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, MR71518, August 14 – 17, 2021

duced to predict the intention of the user by taking into account the movement of his hand and his gaze, and five criteria are introduced to characterize the performance of these strategies. This work concludes with an experiment and analysis based on seven trajectories recorded from three users. From this analysis, several Pareto fronts are calculated.

2 Description of the context

The context of the study is the evaluation of the perceived quality of a virtual car interior during the first design phases. In a given scenario, the user sits in the real world for a visual virtual reality experience inside the car. The user wears a HMD and cannot see the robot, which explains the safety problem (Fig. 1). While the user is trying to interact with the virtual object of the environment, the robot must come and position a sample of the material associated with the local surface, to provide a tactical sense of touching the object [19,27]. A motion capture system based on HTC Vive trackers is used to know the position of the body and especially the hand used for interaction as well as the position of the chair and the robot [31] (Fig. 2). Currently, the prop can carry six different materials. The robot is fixed on a 0.8 m high table and the user sits on a seat 0.6 m above the floor. The placement of the robot in the scene has been chosen to be able to reach all the places where the user's hand will want to have haptic interaction with the robot's probe[7].

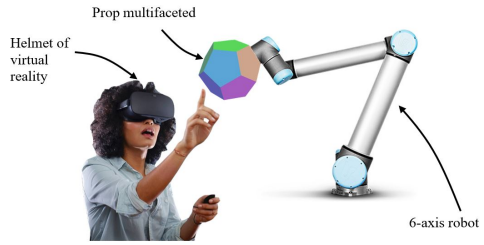

Fig. 1 Conceptual scheme of the experimental platform with a user touching a prop carried by a cobot and wearing a HMD [7]

Fig. 2 The complete system setup for human-robot interaction

A virtual model in the Unity3D ® software represents the fixed objects in the environment, as well as the moving objects, which are the robot thanks to the encoders of the motors and the user thanks to HTC trackers located on the hands and on its seat. The industry partner provided the virtual model of the car design (Fig. 3).

An industrial robot can perform powerful and fast movements that can be dangerous for the humans around it. Involuntary contact between the robot and humans is a threat. This is particularly important in a virtual reality context where humans equipped with an HMD will not be able to anticipate the robot's movements. Today more than ever, humans work closely with robots. In the case

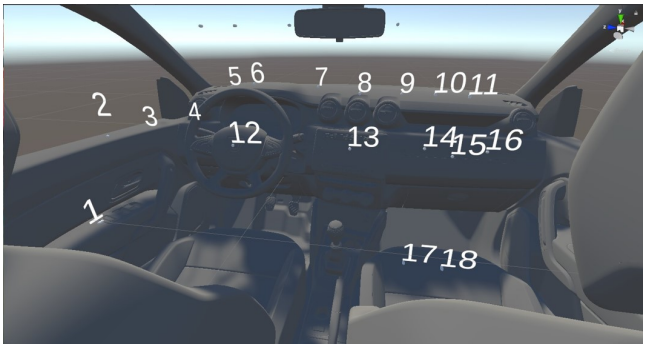

Fig. 3 The Unity environment

of intermittent contact interface ICI, contact is inevitable between humans and robots. Cobots are best suited to such a scenario, but in terms of human safety, accident prevention can always be improved [3]. These robots are designed to work at limited speeds during potential contacts. Moreover, it must be ensured that the desired contact with the robot during interaction will not result in a necessary restart of the robot after a safety stop [16].

Modulation of the robot's speed according to the robot's location in relation to human is one of our objective.

3 Human intention prediction in VR environment

3.1 Detection of the target of human motion. Robots need to anticipate human's future actions and act accordingly while performing collaborative tasks. In most human-robot collaboration systems, the motion of robots is based on some predefined programs, which are task-based. However, most tasks are highly complex and it is difficult to redefine a complete set of instructions for such situations. In such tasks, the roles of the robot should be changed from purely automated machines to autonomous companions. Previous works relied on supervised learning methods to build models of human motion, which relied on understanding the environment, offline training or manual labeling, adaptation to new people, and motion styles.

An expectation-maximization (E-M) algorithm and a neural network to infer human intentions in a 3-dimensional (3D) space were used in [26]. They modeled a function with intentions as parameters and developed a neural network to learn human arm dynamics. In [24] time series analysis for the motion of the human arm based on demonstrations of human arm reaching motion, which synthesized anticipatory knowledge of human motions and subsequent action steps to predict was used. A combination of a two-layer framework of Gaussian mixture models and unsupervised learning to predict a remainder of the trajectory from a prior observed human arm motion in reaching tasks was used in [17].

In [14], a Markov decision process to anticipate a belief about possible future human actions was used by modeling the human's and robot's behavior and then constructed a graph to represent the human motion and interaction with objects.

Human intention is mainly expressed through the behavior of humans and the objects they interact with. Most of the current research on human intention prediction just focuses on action classification, in which the human action is classified into several categories, such as running, walking, jumping [5] which is inadequate for accurate inference of human intention in human-robot collaboration.

We propose a human robot interaction framework that combines hand motion with gaze direction to build models on the fly which predict human intention in virtual reality and move the robot to the required position in a virtual space without offline training.

3.2 Proposed model. The aim of the work is for the human to make contact with different parts of the car, in a design phase

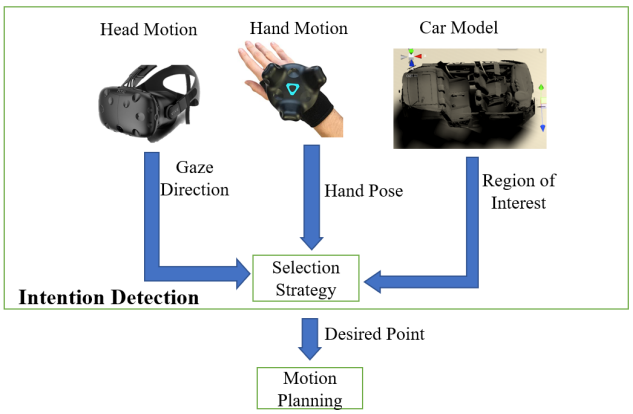

Fig. 4 Diagram of the inputs used to choose a robot movement strategy

where only a virtual model exists, to be able to assess the quality of the materials. The areas to be explored are limited, driver's door, passenger seat, dashboard, touch pad. Depending on where the human wants to touch, the robot must position itself so that the human can touch the appropriate material placed on the probe. The probe has a certain surface area, so a limited number of Regions of Interest (ROI) in the car have been defined that the robot will have to reach to allow contact with the human. The set of 18 ROI considered is described in section 3.3. The objective is therefore to determine as soon as possible the ROI that the user wants to reach and even more so that the robot's probe is positioned as soon as possible on this ROI. If the probe arrives before the human, the human will be able to make contact without being aware that he is in a virtual world, otherwise the waiting time before making contact should be as short as possible.

The major elements involved in our approach are summarized in Figure 4. Measurements of the pose of the hand and the gaze direction via the orientation of the HMD are used to select ROI where the human hand will touch the prop of the robot.

It should be noted that as the objective is that the robot arrives at the target as soon as possible, several strategies are possible and can be combined:

- Detect the target at the earliest,
- Move the robot as soon as possible in the right direction even if the final target is not yet known,
- Move the robot as quickly as possible.

As we are in a cobotic context with a human locked in a virtual world that does not see the robot (the robot can also be visualized in the virtual mode but the immersion will be less), safety is a priority. A description of the methods implemented to have a fast speed of movement of the robot and ensure safety will be discussed in section 4.2.

3.3 Scene Information. From the model of the car in Unity virtual reality software, we defined the ROI the user is to interact with. Each ROI is represented as a capsule placed at the center of the surface. For each surface the desired orientation of the probe is defined. We have defined 18 ROI to be studied inside the car (Figure 5). They are located as follows:

- Four capsules on the door,
- Four capsules on the chair,
- Four capsules on the dash board,
- One capsule on the steering wheel,
- One capsule on the touch pad,
- Three capsules on the glove compartment,
- One capsule on the speedometer.

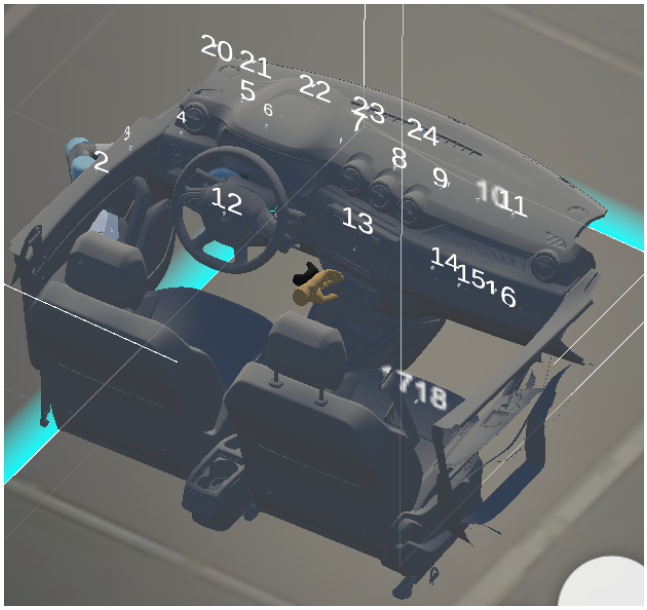

Fig. 5 Location of the ROI 1 to 18 inside the car and safe-points 20 to 24

3.4 Architecture of data flow. The proposed architecture for the project describes the different interactions of each element of the system. It provides an overview of how the instances share information and communicate with each other, as shown in Figure 6. The architecture is a description of how the system works and what tasks are supported by the different instances.

ROS [30] is the middle ware that communicates with the robot and Unity. Based on this information, pre-computed trajectories are selected so that the robot reaches the desired positions while knowing the current states of all objects in the scene. Once the trajectory is selected, we communicate with the UR5 robot using the *ur_modern_driver*. Thus, we can move the UR5 robot with the ROS control, and send as output the current states of the robot's joints for visualization in Unity.

4 Safe and fast motion of the robot

4.1 Cobot motion and user avoidance. The robot will navigate between a finite number of points which are our ROI. How-

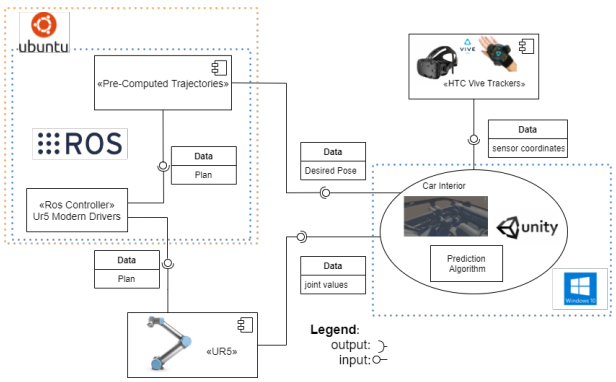

Fig. 6 Flow of data between the systems

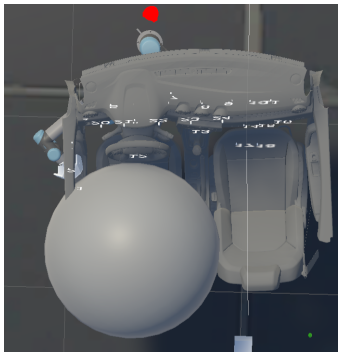

Fig. 7 In the definition of the robot motion to joint the ROI, the sphere that represents the user occupancy zone is avoided.

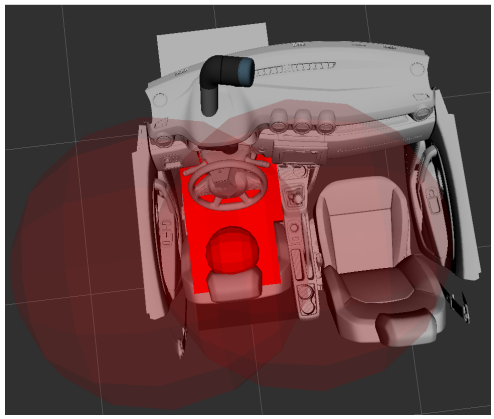

Fig. 8 Car interior and user workspace

ever the movements must ensure that collisions with the human are avoided. To do this, we will generate offline robot movements that ensure that no part of the robot enters an area encompassing the human at rest in the driver's seat. The area to be avoided is composed of a sphere, and is illustrated in Figure 7. The dimension of the sphere covers the human head and torso and part of the arm but the hands can be outside since they must be able to reach the ROI. For this we need to construct 18×17 off-line trajectories that we will assemble in line according to the ROI detected to accomplish the task. These trajectories being close to the human, they are realized with a maximum speed of 0.25m/s to ensure the use of the UR5 cobot according to the ISO standard for human-robot collaboration [10]. This condition guarantees that a possible collision with the human will not hurt him. The implementation of the trajectory planning algorithms is described in [8].

4.2 Definition of velocity zones. The robot must be moved closer to the target point to prepare for the interaction. The movement must be fast so that the robot has arrived before the human and thus avoid unpleasant waiting but the maximum speed of the robot must be limited for safety reasons. Figure 8 shows the scene of the VR environment, it consists of car interior and user model.

Based on this, we distinguish three velocity zones:

- The human workspace (HW), defined as two spheres whose radius is the size of the arm centered on the shoulders of the mannequin. This workspace will evolve according to the movements of the human. We could also consider a constant space if we limit the realistic movements of the torso. This space is represents by blue circles in Figure 9.
- The inside of the car (IC): this space delimits the area where we know the human must move. Even if the Unity model is complex, this zone can be approximated by a larger simple region that includes the real interior of the car. The gray rectangle, in general, represents the entire Unity model and we define a plane, depicted by the red line in Figure 9, that separates the region that can be reached by the user.
- The free space (FS) cannot contain points that are in the HW. We can have a certain safety margin to define this zone. In our example, this zone is simply limited by a plane represented in red in Figure 9.

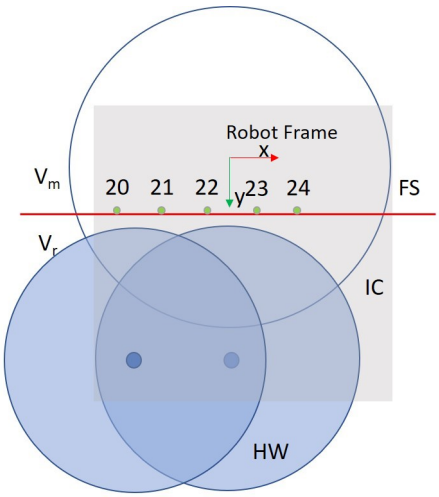

Fig. 9 The spaces defining the robot velocity. The two blue spheres described the human workspace when seated. The grey part shows the car model. The transparent sphere is the robot's working area. The red line delimits an area where the speed of the robot can be higher because there is no risk of collision with the human.

The limit on the robot velocity is chosen according to the space:

- When the robot moves in FS, it can do so at maximum speed V_m (all parts of the robot are in FS),
- When the robot moves outside of FS, it must move at reduced speed V_r,

The speeds are chosen such as $V_m \geq V_r \geq 0$. The different spaces are shown in Figure 9. The blue hollow circle is the robot workspace, two blue-filled circles are the workspace of the user's hands. The grey rectangle is the complete interior model of the car and the red line is the plane that we use to differentiate the reachable and unreachable parts of the car by the user.

4.3 Velocity profiles based on zones. To ensure safety and also have better response time we defined two velocity profiles with maximal velocities $V_r = 0.25\ m/s$ and $V_m = 4\ m/s$ based on the zones defined above. This idea imposed that we move the robot in the FS. To illustrate the idea, we devise the scene as shown in Figure 10. We define four points:

- Two points A' and B' are on the plane boundary, these are in the FS, and fast motion between these points can be produced. For the application studied in the paper the capsule denoted 20 to 24 are in the FS.
- Two points B and A are inside the car (IC), one point on the dashboard and another on the passenger's seat. These points play the same role as the ROI 1 to 18.

We analyze two different scenarios based on different velocity combinations.

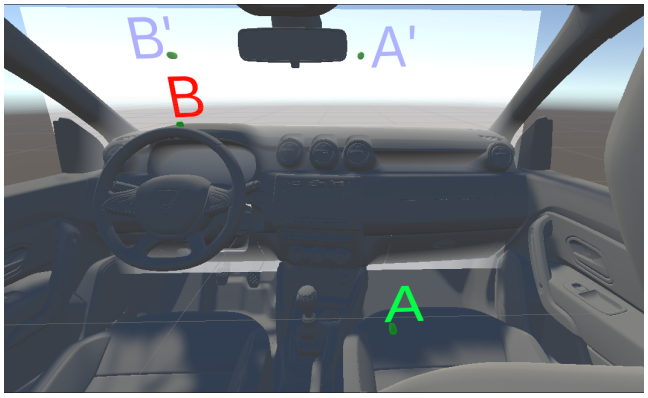

Fig. 10 Illustration for the comparison of motion using safe-points.

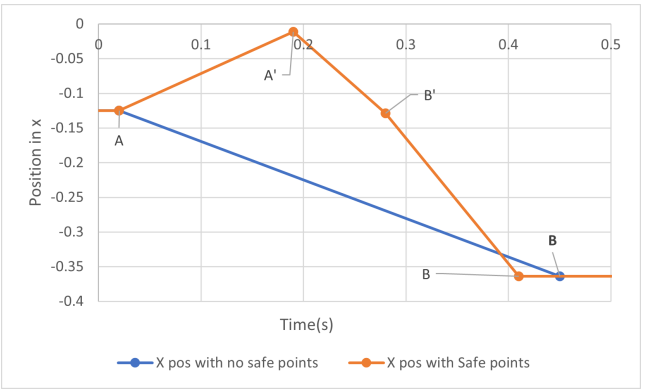

Fig. 11 Comparison of motion through safe-points and without safe-points.

- *The shortest way*: Knowing the target point, the robot moves towards it, and adapts its speed according to the spaces it crosses. In the studied example, it goes directly from A to B with a maximal velocity less than $0.25m/s$.
- *Safe-points*: We use safe-points to keep the robot's speed high. In the studied example, the points A' and B' belongs on the plane that limits FS. The robot goes from A to A' with a maximal velocity less than $0.25m/s$, then from A' to B' with a maximal velocity less than $4m/s$, B' to B with a maximal velocity less than $0.25m/s$.

The movement of the robot inside the car should be performed with reduced speeds for safety reasons. In some cases, when the desired point is far away from the user space, it takes longer to reach it due to the low speed. For such situations, we use the via-points on plane boundary in FS, called safe-points, between which the robot can move at high speed. The path is longer but its execution can be faster.

New trajectories are calculated off-line to connect the ROIs and the safe-points with the two motion speeds.

4.4 Safe-points. As we have shown in the previous section, the interest to move the robot in FS is to speed up the robot movements. Five points on the plane boundary of FS have been defined SP_{20}, SP_{21}, SP_{22}, SP_{23}, SP_{24} to be used for this purpose.

The interest of moving the robot in the FS is also to increase the safety of the operator by moving the robot away from the human. To quantify this notion of safety, we will define an average distance between the robot's end-effector and the sphere encompassing the human's torso shown in Figure 7. The higher this distance, the safer the human/robot interaction will be. This distance is calculated in an approximate way from the points of passage of the robot (P_1,...P_{18} and SP_{20},...SP_{24}), by making the hypothesis of a straight line displacement at constant speed between the points (defined as the distance between the points divided by the duration of the displacement).

$$d_s = \frac{\sum_{i=1}^{N}(\|p_i - C\| - R)}{N} \quad (1)$$

where the robot motion is sampled in N instant, p_i is the coordinate vector of the end-effector of the robot for sample i, C is the coordinate vector of the center of the sphere shown in Figure 7 and R is its radius.

Considering the user safety and robot velocity it is of interest to pass through points on the plane boundary in FS, when we have long robot trajectories to make. In the next part of the study we will show that this can also be useful when a movement is initiated by the human by hand and gaze but without knowing yet where the human will stop. Placing the robot on one of the safe-points SP_{20}, SP_{21}, SP_{22}, SP_{23}, SP_{24} will allow the robot to get closer to the goal more quickly.

4.5 Comparison of motion with or without safe-points. An example is illustrated in Figure 11 where $A = P_{17}$ and $B = P_5$, $A' = SP_{24}$ and $B' = SP_{21}$. The points A' and B' are positioned on the plane that divides the two different velocity zones.

We compared the time taken by the robot to move between two points, taking into account presence of safe-points and without them.

The results in Figure 11 show the position of points in X coordinate with respect to time. The Figure 11 is a representation to show the point and at what time the robot reaches that point. From the recorded trajectories of the robot to reach the points, the X-coordinates of the robot are plotted at the beginning and end of the trajectory (and connected by a straight line) against time in Figure 11. The direct motion is shown in blue, when the motion with intermediate safe point is shown in red. It proves that by passing through safe-points A' and B', the robot takes less time than going directly.

By travelling through safe-points, the robot starts from point A and moves through A', B' at a higher velocity and then to the final point B. The total time taken to reach the final point was 0.41s. For the motion inside the car, the robot arrives at the final point after 0.45s.

5 Proposed strategies to predict human intention

We will study and compare four strategies that integrate the position of the hand, the direction of the gaze, and the use of safe-point to efficiently move the robot to one of the ROI that the human want to reach.

5.1 Strategy A. As the user's objective is to touch with his hand the ROI, the first and simplest strategy proposed is to consider that the point to be reached is the closest to the hand position. The strategy is presented in Figure 12, in the example the selected point is P_2. The main advantage is the simplicity in the approach. For the search of the nearest point, a nearest neighbor search algorithm is used to find the nearest point to the hand. An implementation of this algorithm for VR environment is done in [22]. The strategy is summarized in Algorithm 1.

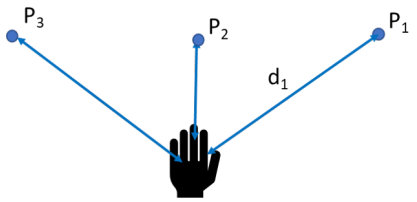

Fig. 12 Pictorial representation of strategy A

Algorithm 1 Strategy A: Predictions with hand

Input: Hand position $P_h \in \mathbb{R}^3$.
Output: Nearest Point P in the set of P_i, $i = 1...18$.

1: **function** STA(P_h)
2: Using hand pose as a query point q, return nearest point.
3: **return** P
4: **end function**

The main characteristics of the approach are:

(1) The point is detected only when the human has almost reached the points;
(2) If two points are equidistant, a prediction fluctuation can occur with small changes in hand motion;
(3) During the movement of the hand, intermediate points can be detected, which will allow the robot to start its movement before the desired end point is detected.

5.2 Strategy B. Head gaze direction is introduced to limit the detection of points to only what the user can see. The detection of the interest point is only possible if the point is in the field of view. If the no point is detected the the previous desired point is conserved. The strategy is presented in Figure 13. The pose of the hand is used to selected 2 points, the closest to the hand, in the example P_1 and P_2. From these 2 points the closest to the gaze direction is selected, by comparison of the angle between the line connecting the point and the line of view. In the example the closest point is P_2. The strategy is summarized in Algorithm 2.

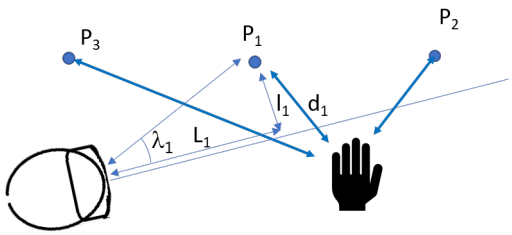

Fig. 13 Pictorial representation of strategy B

The aim of this approach is to try to find the point of interest with a little anticipation compared to the previous method, by being able to choose a point that may be a little further from the hand but directed according to the direction of the gaze.

5.3 Strategy C. If the hand is far from the point of interest, it is probably on the way but still far from the goal, so it may be appropriate to move the robot to a safe-point to prepare for a higher speed movement. This strategy is an extension to strategy B, but with added extra safe-points. This strategy is designed such that the robot will always go to the safe-point if the distance between the hand and closest point is above a threshold, here $0.3m$. The strategy is presented in Figure 14. In the example, the point P_2,

Algorithm 2 Strategy B: Predictions with head gaze

Input: Hand position $P_h \in \mathbb{R}^3$, HMD position $P_s \in \mathbb{R}^3$, Head Orientation $O_s \in \mathbb{R}^4$.
Output: Nearest Point P.

1: **function** STB(P_h, P_s, O_s)
2: Using hand pose as a query point q, return nearest 2-points.
3: Find the gaze direction as a unit vector from the central point of the eyes and draw a ray in the gaze direction.
4: Calculate the distance l_1 of each point in n_p from the ray.
5: Find the angle λ_i for each point such that $\lambda_i = l_i/L_i$ where L_i is the distance from the HMD to the projection of the point on the line. (Shown in Figure 13).
6: The point with $\min(\lambda_i)$ is the closest point P.
7: **return** P
8: **end function**

the closest to the view line among the two closest to the hand, is at more than $0.3m$ from the hand, thus the robot will go to the safe-point which is the closest to P_2 among SP_{20}, SP_{21}, SP_{22}, SP_{23}, SP_{24}. The strategy is summarized in Algorithm 3.

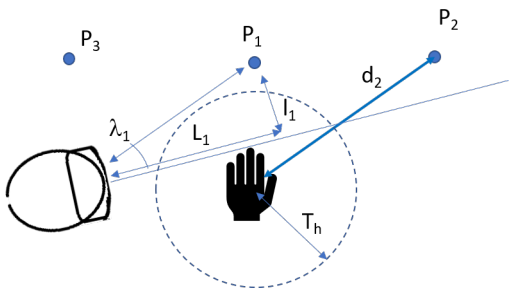

Fig. 14 Pictorial representation of Strategy C

The main characteristics of this strategy are as follows:

- The difference between this approach and strategy B can only be seen for long displacements (of more than $0.6m$ between the points) for which the hand displacement can be quite far from the points P_i, $i = 1, ...18$.
- There is a risk that the robot will move to a safe-point in an inefficient way with a longer path that will not allow the robot to arrive faster
- The results obtained can vary with the choice of the threshold T_h.

5.4 Strategy D. All the strategies presented so far are based on a selection or pre-selection of the point of interest based on the hand position. Here the approach is different and the selection is based on the direction of the gaze which can greatly anticipate the movement of the hand. As in strategy C, safe-points will be used if the point of interest is more than $T_h = 0.3m$ away from the hand. The strategy is presented in Figure 15. In the example, the point P_2, the closest to the view line among the points in the frustum of the HMD. As the point P_2 is at more than $0.3m$ from the hand, thus the robot will go to the safe-point which is the closest to P_2 among SP_{20}, SP_{21}, SP_{22}, SP_{23}, SP_{24}. The strategy is summarized in Algorithm 4.

Algorithm 3 Strategy C: Addition of safe-point

Input: Hand position $P_h \in \mathbb{R}^3$, HMD position $P_s \in \mathbb{R}^3$, Head Orientation $O_s \in \mathbb{R}^4$, Hand Threshold T_h (Shown in Figure 14).
Output: Nearest Point P.

function STC(P_h, P_s, O_s, T_h)
  function STB(P_h, P_s, O_s)
    return P
  end function
  if distance(P,P_h) < T_h **then**
    return P
  else
    for all $SP_i \in SP$ **do**,
      d_i = distance(P, $\mathbf{SP}_i$).
      min(d_i);
    end for
    return SP_i
  end if
end function

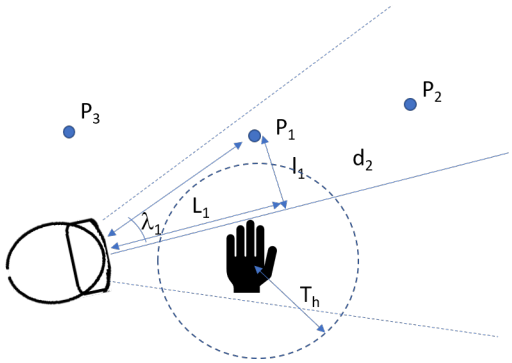

Fig. 15 Pictorial representation of Strategy D

Algorithm 4 Strategy D: Predictions with head gaze and safe-points

Input: Hand position $P_h \in \mathbb{R}^3$, HMD position $P_s \in \mathbb{R}^3$, head orientation $O_s \in \mathbb{R}^4$, hand threshold T_h.
Output: Nearest Point P.

function STD(P_h, P_s, O_s, T_h)
  n_i = points in the view frustum of HMD.
  Find the gaze direction as a unit vector from the central point of the eyes and draw a ray in the gaze direction.
  Calculate the distance of each point in n_i from the ray.
  Find the angle λ_i for each point such that $\lambda_i = l_i/L_i$ where l_i represents the distance between a point and it's projection on the line as calculated in previous step and L_i the distance from the HMD to the projection of the point on the line (Shown in Figure 15).
  The point with min(λ_i) is the closest point P.
  if distance(P,P_h) < T_h **then**
    return P
  else
    for all $SP_i \in SP$ **do**,
      d_i = distance(P, $\mathbf{SP}_i$).
      min(d_i);
    end for
    return SP_i
  end if
end function

This strategy is based on the assumption that the task will be carried out with coordination of gaze and movement. In general, it is reasonable to think that the gaze anticipates the movement. In the context of the study, where the human is enclosed in a virtual world, it is likely that he will not be disturbed by external elements and that he will remain focused with his gaze directed towards the point of interest. The context should therefore be favourable to this approach.

5.5 Strategy to move the robot. When the target is defined, the robot must be moved. For this a series of trajectories that avoid obstacles have been defined (see section 4.1) between point of interest and/or safe-point. While the robot is moving, a new point of interest can be defined. One could stop the robot's movement and recalculate an obstacle-free trajectory online. However, in order to avoid wasting time in this calculation, while predefined trajectories have generally short execution times, so the robot is let to perform its movement. And it will take into account the new target at the end of its movement.

5.6 Definition of the criterion to compare the strategies. The main research question was to find a selection strategy which maximizes user safety while minimizing robot response time. Four strategies explained in the previous section were tested against each criterion. The strategy selected needs to minimize robot time to reach a desired target pose while ensuring maximum user safety.

To evaluate the strategies, the following criteria were considered for strategy.

(1) Efficacy:
- Q_1: If the strategy detects the final point or not. A value of 1 or 0 was assigned if final end point was detect or not.

(2) Time for detection:
- Q_2: Time taken by the strategy to detect the desired/final point the user want to reach.
- $Q_{2_{norm}}$: In order to be able to compare the results for several strategies, we defined a normalized criterion. It a ratio of each value of Q_2 for a trajectory divide by the minimum value of Q_2 for this trajectory and the four strategies analyzed. [Q_2/min(Q_2)]. For the best strategy with respect to this criterion the $Q_{2_{norm}} = 1$.

(3) Time for robot:
- Q_3: Time taken by the robot to reach the final point (to move from start to desired point including all via points). It is the sum of the duration of all the pre-computed trajectories according to the strategy of motion of the robot according to section 5.5 and the time that the robot waited to have new point where to go.
- $Q_{3_{norm}}$: As for the criterion Q_2, we define a normalized criterion, to be able to compare the strategy for several trajectories. Its a ratio of each value of Q_3 for a trajectory divide by the minimum value of Q_3 for this trajectory and the four strategies analyzed. [Q_3/min(Q_3)]. For the best strategy with respect to this criterion the $Q_{3_{norm}} = 1$.

(4) Robot distance:
- Q_4: The distance travelled by the robot from start to end point for all via points the robot travels through.
- $Q_{4_{norm}}$: The ratio of distance travelled by the robot (from start to end point for all via points the robot travels through) with distance between start and end point.

(5) User distance:

- Q_5: The mean distance between the sphere centered on the driver's seat with a radius of 0.5m and all points on the trajectory. This distance is evaluated via the equation (1) and characterizes the safety of the user. The further the robot is from this sphere, the safer it is.

6 Experiments and Analysis of the Results

6.1 Experimental Setup. We used the hand motion sensor as a proximity sensor. The objective is to find the closest ROI with respect to the hand. This is an optimization problem of finding a point in a closed set that is closest to a given point. Using the Head mounted display, we find the points in the user view and if the gaze is directed towards a point P_i. We classify the distance of the points to the direction of the gaze as a function of l_1/L_1. The user will direct his hand towards a capsule (discrete set of N points). The goal is to detect as soon as possible which point to be reached by the human and to move the robot to that point:

(1) If the direction is known, the robot can be moved to intermediate points to facilitate the task.
(2) Safe-points SP_i ($i = 20 : 24$) are defined on the boundary plane, which is located outside the car's interior space. Between these points the robot can move quickly.
(3) At each moment, from the sensor data, we define the target point among the set of N points P_i ($i = 1 : 18$)

Seven trajectories were considered in the experimental design: two consisted of long distance trajectories (from points 2 to 11 and 5 to 18), three medium distances (from points 5 to 11, 5 to 15, 12 to 15) and two short distances (from points 3 to 4 and 17 to 16). Three different participants have done the seven trajectories.

The participant was seated in the car seat $0.6m$ above the ground and at a position of $0.9m$ in y and $-0.1m$ in x from the robot base frame. The sphere used for the obstacle avoidance of the user is centered at this reference point. An HTC vive HMD was worn by the user and vive sensors were attached to the user's dominant hand as shown in Figure 2. Then for each trajectory, the user was instructed to move his hand from a start point to a defined end point.

For each trajectory performed by the user, data was recorded. It comprised of position of the hand tracker, the head position and orientation. The user can do the task at the speed he wants. During this experiment the robot was not moved to ensure the security of the user. As the objective is to compare the strategy with exactly the same data as input, thus there is no reason to move the robot with one chosen strategy. The data recorded was used to perform the analysis in parallel with the four strategies in order to compare them on the same data set.

6.2 Analysis of one experiment. The four strategies are described and illustrated on one example, a trajectory done by one subject for moving his hand from P_2 to P_{11} was recorded. A visual trail of the user hand is shown in Figure 16. This recorded motion was used to analyze the four proposed strategies.

6.2.1 Detection of points of interest. Figure 17 shows the sequence of points that are detected for the four studied strategies. It can be visualized that different strategies have different intermediate points selected except for strategies A and B that produced the same sequence of points detected.

Based on the results from the different strategies, it can be observed that strategies A, B select intermediate points which are inside the car while strategy C and D select the safe-points as some of the intermediate points. For this example, all the strategies allows to find the desired final point P_{11}. However the strategy D success to detect this point earlier that the strategies A, B, C that detect the final desired point at the same time (as it can be seen in Table 1).

The obtained sequence of points of interest is now detailed:

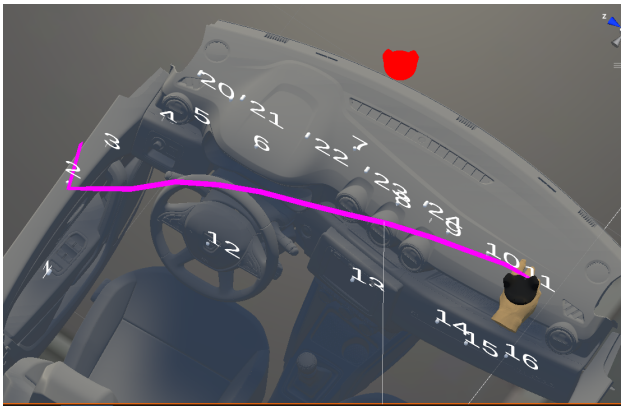

Fig. 16 User hand trail for motion from point 2 to 11

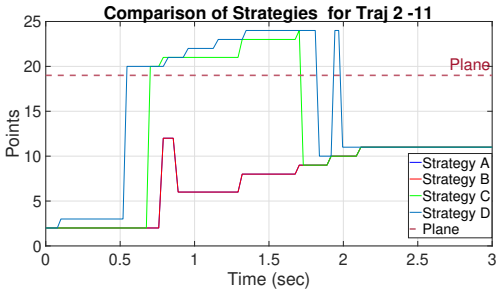

Fig. 17 Points of interest detected for different strategies for a hand motion from 2 to 11

- Strategy A: P_2, P_{12}, P_6, P_8, P_9, P_{10}, P_{11}.
The points are selected based on the least distance to the hand, the points selected can be easily explained by the hand trail described in figure 16.

- Strategy B: P_2, P_{12}, P_6, P_8, P_9, P_{10}, P_{11}.
The selection of this strategy is similar to strategy A for this example because all points were all the time the point closest to the hand is also closest to the direction of gaze. Since the set of points detected is the same as for the strategy A. The robot motion will be similar and analyzed simultaneously.

- Strategy C: P_2, SP_{20}, SP_{21}, SP_{23}, P_9, P_{10}, P_{11}.
For strategy C, a threshold is introduced around the detected points to choose whether the robot should go to the point or to a safe-point. It is observed on Figure 16 that the hand passes at a distance $> 0.3m$ from the points P_{12}, P_6, P_8. Consequently, the selected points will be the associated safe-points SP_{20} SP_{21} and SP_{23}. Then points P_9, P_{10}, P_{11} were detected and found to be within the required threshold of the distance from the hand.

- Strategy D: P_2, P_3, SP_{20}, SP_{21}, SP_{22}, SP_{23}, SP_{24}, P_{10}, SP_{24}, P_{11}
Strategy D uses the direction of gaze as the primary criterion and it is therefore more difficult to predict the sequence of points detected based on Figure 16. In this strategy P_3 is selected, it is done based on the user gaze and since the hand moved not far from this point P_3 is selected. For the following points selected by the gaze, the hand is farther from the points so the associated safe-points were selected. Then the gaze is directed toward the point P_{10}, since the distance from the hand was below a threshold T_h, the point P_{10} was selected. Then the gaze is probably oriented to point P_{11} since it's distance from the hand is above the threshold, the robot has to return to the safe-point P_{24} and finally when the the hand is close

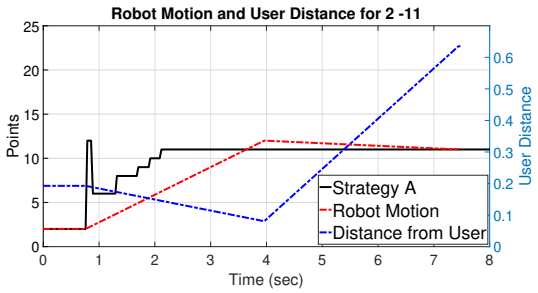

Fig. 18 Robot motion, user distance and time for detection, for strategies A and B for trajectory 2 to 11

to the final point P_{11}, the point is detected. This happens at a moment when the point P_{11} is not yet the closest point to the hand and is therefore not yet detected by strategy A. The points detected in this example show that the gaze does not go directly to the goal but sweeps along the path accompanying the hand. The results also show a certain sensitivity of the point sequence to the value chosen for the threshold.

6.2.2 The robot motion for the four strategies. For the three different selections of points (strategies A-B, strategy C, strategy D), the robot motion and the safe distance is now commented. Figures 18, 19, 20 illustrate, starting from the sequence of points detected represented in black as function of time, the corresponding robot motion represented in red as function of time. On the same figure, the distance between the end-effector of the robot and the sphere encompassing the human, is shown in blue as function of time and expressed in metre. This representation shows only the instants corresponding to the start and end points. Between these points straight dotted lines are drawn. This curve is directly calculated based on the robot motion and will be used for the evaluation criterion in Table 1. As for the robot motion, the exact value are calculated only at initial and final points and interpolated by strait line.

The sequence of progression of robot motion with time from start to end point is described below:

Strategy A and B:. The progression of the robot motion is indicated by the red line as shown in Figure 18. Starting from point P_2, the robot waits for a new point. When P_{12} is detected, the robot starts moving and before it arrives, P_6 is detected. However the robot has to continue and complete the motion so then arriving at P_{12} the robot now detects P_{11} as new desired point, so the robot moves directly to P_{11}. The robot avoids all the points that are detected during the motion towards P_{12}. This pattern continues until the robot reaches the last point detected.

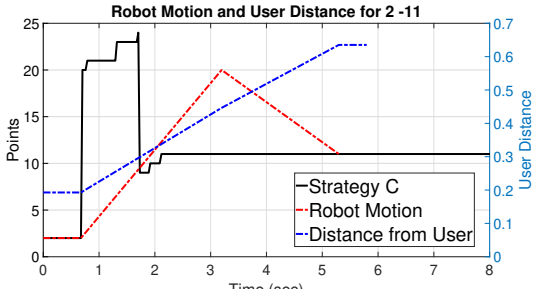

Fig. 19 Robot motion, user distance and time for detection, for strategy C using trajectory 2 to 11

Strategy C:. A graph of robot motion is indicated by the red line as shown in Figure 19. The robot starts at P_2 and waits for a new point. When point SP_{20} is detected, it starts moving but along the way, a new point SP_{21} is detected, so it has to complete the motion to SP_{20} first. By the time it has reached the point SP_{20}, points SP_{21}, SP_{23}, SP_{24} have been detected and passed by the prediction algorithm. Once a new point has been detected the goal state of the robots updates and neglects the previous points that have not been reached. After reaching SP_{20} the prediction now shows point P_{11} as the desired destination. So the robot moves to P_{11}.

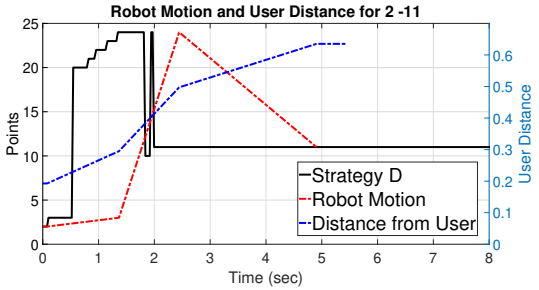

Fig. 20 Robot motion, user distance and time for detection, for strategy D using trajectory 2 to 11

Strategy D:. The motion of the robot is indicated by a red line as shown in Figure 20. The robot starts from P_2 and when P_3 is detected, it moves to point P_3. When P_{20} is detected, the robot is still in motion to P_3, so it ignores the point. Further SP_{21}, SP_{22}, SP_{23} and SP_{24} are detected, but when it the robot reaches P_3, the prediction system still predicts SP_{24} s it moves to SP_{24} without waiting. Again during the motion P_{10} and SP_{24} are detected, but before arriving, P_{11} is detected so on arrival at SP_{24}, it does not wait but continues to P_{11}, where it finally stops.

6.2.3 Criterion analysis for single trajectory. Table. 1 shows the complete data for all criteria proposed, for the single trajectory 2-11. Q_1 is the success of the strategy detecting the final goal state. It can been seen that all strategies are able to detect the goal point. From the table it can be seen that $Q_{2_{norm}}$ and $Q_{3_{norm}}$ has its best value for strategy D. These strategies have a better detection and robot travel time. However the distance traveled is not the best as this strategy allows the selection of safe-points which increase the robot travel distance. When considering safety, strategy D is second best. The best safety Q_5 is provided by strategy C.

After this detailed analysis for one trajectory, a discussion of the criterion evaluation for seven trajectories recorded is presented in the following sections.

6.3 Analysis of all recorded experiments. The results obtained are summarized in Table. 2. For all the tests, the final desired position is detected, thus criterion efficacy Q_1 is one and the criterion is not summarized in the Table 2. The analysis of the results will be separated into two parts. Firstly, an observation of the results obtained for the different trials will enable us to arrive at certain conclusions about variations of the results as function of the trajectories. Then, in a second part, we will use the average values of the different criteria, taking into account the seven trajectories, to highlight the particularities of each strategy by comparing the criteria two by two.

6.3.1 Variations of the results for the different trajectories. For almost all trajectories, the strategy B and A produce the same result, but for one test, the direction of the gaze is not well directed at the end of motion and a delay is observed with strategy B, contrary to what is expected. This delay affects the time of detection and also the time for the motion of the robot (trajectory 12-15).

From the point of view of efficacy of detection of interest point, strategies A and D were the most efficient: two times for strategy A

Table 1 Strategy analysis for the trajectory 2-11

Trajectory 2-11								
Strategy	Efficacy	Time for Detection		Time for Robot		Robot Dist.		User Dist.
	Q_1	Q_2	$Q_{2_{norm}}$	Q_3	$Q_{3_{norm}}$	Q_4	$Q_{4_{norm}}$	Q_5
St. A	1	2.0868	1.0600	2.1818	1.0181	1.4648	**1.1959**	0.4370
St. B	1	2.0867	1.0600	2.1818	1.0181	1.4648	**1.1959**	0.4370
St. C	1	2.0867	1.0600	2.1818	1.0181	1.5690	1.2809	**0.4803**
St. D	1	1.9686	**1**	2.1430	**1**	2.1930	1.7903	0.4427

Table 2 Complete analysis for seven trajectories

	Criteria	Strategy	Long Trajectory		Medium Trajectory			Short Trajectory		Analysis	
			2-11	5-18	5-11	5-15	12-15	3-4	17-16	Mean	St. Dev
Time for Detection	Q_2	A	2.0868	2.3029	2.5878	2.1767	1.9842	0.6418	1.0381	1.8312	0.6597
		B	2.0868	2.3029	2.5878	2.1767	2.3366	0.6418	1.0381	1.8815	0.6825
		C	2.0868	2.5370	2.5878	2.1767	2.3366	0.6418	1.0381	1.9150	0.7076
		D	1.9687	2.5370	2.2383	2.0428	2.3366	0.3258	0.8709	1.7600	0.7687
	$Q_{2_{norm}}$	A	1.0600	**1**	1.1561	1.0655	**1**	1.9701	1.1919	1.2062	0.3190
		B	1.0600	**1**	1.1561	1.0655	1.1776	1.9701	1.1919	1.2316	0.3085
		C	1.0600	1.1016	1.1561	1.0655	1.1776	1.9701	1.1919	1.2461	0.2995
		D	**1**	1.1016	**1**	**1**	1.1776	**1**	**1**	**1.0399**	0.0663
Time for Robot	Q_3	A	2.1818	2.4212	2.6828	2.2642	2.0717	0.7506	1.1440	1.9309	0.6560
		B	2.1818	2.4212	2.6828	2.2642	2.4241	0.7506	1.1440	1.9812	0.6780
		C	2.1818	2.6846	2.6828	2.2642	2.4241	0.7506	1.1440	2.0189	0.7080
		D	2.1431	2.6846	2.3434	2.1303	2.4241	0.4345	0.9777	1.8768	0.7740
	$Q_{3_{norm}}$	A	1.0181	**1**	1.1448	1.0629	**1**	1.7273	1.1701	1.1605	0.2399
		B	1.0181	**1**	1.1448	1.0629	1.1701	1.7273	1.1701	1.1848	0.2309
		C	1.0181	1.1088	1.1448	1.0629	1.1701	1.7273	1.1701	1.2003	0.2214
		D	**1**	1.1088	**1**	**1**	1.1701	**1**	**1**	**1.0398**	0.0651
Robot Dist.	$Q_{4_{norm}}$	A	**1.1959**	**1.4943**	**1.0948**	**1.1582**	**1.0225**	**1.0000**	**1.3042**	**1.1814**	0.1601
		B	**1.1959**	**1.4943**	**1.0948**	**1.1582**	**1.0225**	**1.0000**	**1.3042**	**1.1814**	0.1601
		C	1.2810	1.6766	3.9707	1.5271	3.0161	1.0000	2.9585	2.2043	1.0273
		D	1.7904	3.9039	7.4400	2.8315	4.2133	1.0000	2.6207	3.4000	1.9464
User Dist.	Q_5	A	0.4370	0.3301	0.4846	0.4143	0.3285	0.3535	0.4429	0.3987	0.0570
		B	0.4370	0.3301	0.4846	0.4143	0.3285	0.3535	0.4429	0.3987	0.0570
		C	**0.4804**	**0.4134**	**0.4932**	0.4149	0.3165	0.3535	0.4684	**0.4200**	0.0616
		D	0.4428	0.3836	0.4199	**0.4327**	**0.3510**	0.3535	**0.5044**	0.4126	0.0506

and five times for strategy D. There is no clear correlation between the efficacy of strategy and the length of trajectory. However for short trajectories, the strategy D is most efficient.

6.3.2 User distance vs time for detection. For the purpose of analyzing, $-Q_5$ has been used so as the goal would be to minimize all the selected criteria. A comparison of $-Q_5$ and $Q_{2_{norm}}$ shows that strategies C and D belong to the Pareto front for these two criteria. Strategy D takes the least time to detect a desired point the user would like to reach and a slightly higher mean distance from the sphere as shown in Figure 21. Strategy C has the largest user distance. However, it takes the longest time to detect a point than all the strategies.

6.3.3 User distance vs robot distance. A comparison of $-Q_5$ and $Q_{4_{norm}}$ as presented in Figure 22 shows that strategies A-B and C belong to the Pareto front for these two criteria. Since strategies A and B don't use any safe points they have smallest user distance (max $-Q_5$) and always take the minimal robot distance to arrive to a desired point (min $Q_{4_{norm}}$). Strategies C and D use safe points so, have higher user distance (min $-Q_5$) and robot distance $Q_{4_{norm}}$. While strategy D uses head gaze as primary selection, it gives a value of Q_5 better than A and B but the worst robot distance.

6.3.4 User distance vs time for robot. A comparison of $-Q_5$ and $Q_{3_{norm}}$ as presented in Figure 23 shows that strategies C and

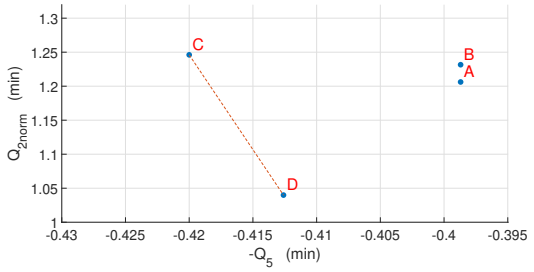

Fig. 21 Comparison of time for detection ($Q_{2_{norm}}$) vs user distance ($-Q_5$) for the four strategies, all trajectories

D belong to the Pareto front for these two criteria. As known Strategy C gives a higher value for safety (min $-Q_5$) followed by strategy D. However strategy D far outperforms strategy C in terms of time for the robot.

6.3.5 Time for robot vs time for detection. Visualization of $Q_{3_{norm}}$ vs $Q_{2_{norm}}$ is shown in Figure 24. It shows that the faster the strategy detects the point, faster the robot reaches the point. The best being strategy D. It uses the head gaze and an added use of safe-points helps it in reaching the desired point faster, as the robot travels at higher velocity than in strategies A and B. Strategy A has lots of intermediate point, but as the hand moves closer to

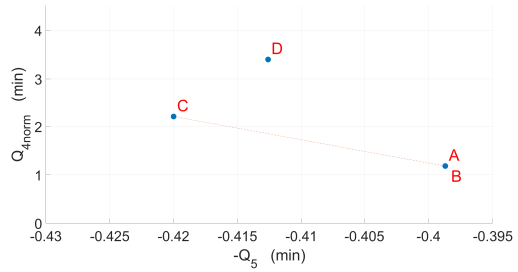

Fig. 22 Comparison of robot distance ($Q_{4_{norm}}$) vs user distance ($-Q_5$) for the four strategies, all trajectories.

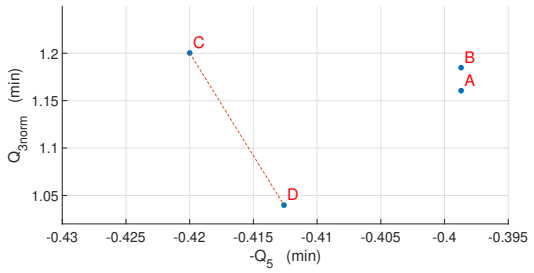

Fig. 23 Comparison of time for robot ($Q_{3_{norm}}$) vs user distance ($-Q_5$) for the four strategies, all trajectories.

the desired point the robot gradually moves closer. This is one of the reason why this strategy is the second best. Even thought strategies B and C have a head gaze, the primary selection is still based on the hand. Unless the hand is closer to the point the robot does not move to the desired point. For strategy D it is the user gaze that help in primary selection of points.

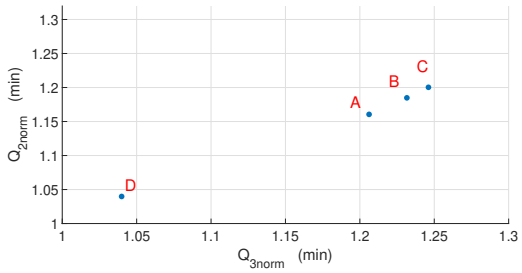

Fig. 24 Comparison of time for robot ($Q_{3_{norm}}$) vs time for detection ($Q_{2_{norm}}$) for the four strategies, all trajectories.

6.3.6 Robot distance vs time for robot. Visualization of $Q_{4_{norm}}$ vs $Q_{3_{norm}}$ is shown in Figure 25 strategies A and D belong to the Pareto front for these two criteria. In contrary to the assumption that longer robot distance implies longer time for robot, the results show that strategy D has minimum time for robot but has longer robot distance. This result achieved is due to combination of fast time for detection and use of safe-point as intermediate points. Strategy C uses also safe-points, but it alone does not guarantee a fast response time.

6.3.7 Time for detection vs robot distance. Visualization of $Q_{2_{norm}}$ vs $Q_{4_{norm}}$ is shown in Figure 26 strategies A and D belong to the Pareto front for these two criteria. From Figure 26,it can be see that Strategy A and B have least robot distance, but its detects the goal later than Strategy D. Strategy C is not ideal in both criteria.

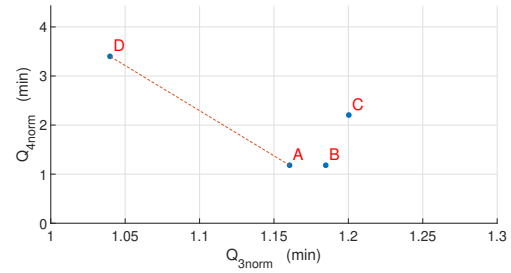

Fig. 25 Comparison of robot distance ($Q_{4_{norm}}$) vs time for robot ($Q_{3_{norm}}$) for the four strategies, all trajectories.

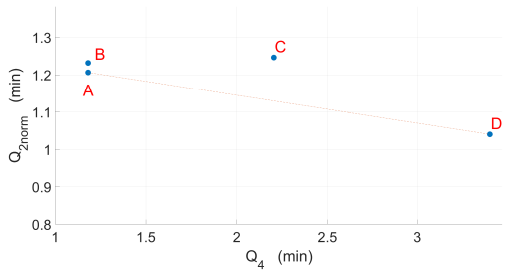

Fig. 26 Comparison of time for detection ($Q_{2_{norm}}$) vs robot distance ($Q_{4_{norm}}$) for the four strategies, all trajectories.

6.4 Discussion. A comparative analysis of data from all the trajectories shows that, if the objective is to maximize safety strategy C and D are good candidate. Both the strategies C and D ensure safety by selecting safe-points when the hand is far away from the desired point. The safe-points are located outside the user reach, such that the robot can travel fast and does not collide with the user. The selection of the safe-points mean that the robot will have to travel a longer distance to reach the desired point. While for strategies A and B, they select intermediate points which are inside the car and no safe-points. So since the points are all inside the car, and the robot has a shorter path distance but has reduced velocity.

Strategy D gives second best safety and at the same time minimize the time to detect/reach a desired point. Therefore it can be seen as the best strategy. The detection time for strategy D is the smallest because we used the gaze of the user to pre-select the points. This plays a big role in giving priority to vision information over information from the hand position. Fastest detection time allows the robot to start moving to the desired point at the earliest time and reach the desired point the fastest.

7 Conclusions

In this article, a collaborative robot is used as a haptic interface with intermittent contact. Four motion prediction strategies are used to select the areas with which the user intends to interact and to move the robot as fast as possible while ensuring user safety. A

We introduce two speed profiles for the user's safety. The robot moves at a higher speed when it is outside the user's workspace. In situations where there is a large distance between two points within the workspace, we introduce via points to reduce travel time. The time needed to go through via points can be less than the time needed to go directly inside the car while being much safer.

Seven trajectories done by three users were analyzed thanks to five criterion. A compromise must be made between user safety and speed for the robot to reach its target.

A simple realtime demonstration of the above system using strategy D can be found here. [Demo]

In future works, we will perform other experiments with dif-

ferent users to test the robustness of our analysis and to know the influence of the threshold value. We will also realize the robot movements at the same time as the user's movements to validate his feeling when he/she hears the robot moving. Additional work is also in progress to reassure the user by showing for example the position of the robot during its movement.

Acknowledgement

This work was funded under the LobbyBot project, ANR-17-CE33 [1]. The authors of the article thank the members of the project for their help in carrying out this work, Lionel Dominjon, Sandrine Wullens, Allexandre Bouchet, Javier Posselt, Anatol Lecuyer and Victor Rodrigo Mercado Garcia.

References

[1] Lobbybot project. https://www.lobbybot.fr/. Accessed: 2021-10-15.

[2] Bruno Araujo, Ricardo Jota, Varun Perumal, Jia Xian Yao, Karan Singh, and Daniel Wigdor. Snake charmer: Physically enabling virtual objects. In Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction, TEI '16, page 218–226, New York, NY, USA, Feb 2016. Association for Computing Machinery.

[3] Andrea Cherubini, Robin Passama, André Crosnier, Antoine Lasnier, and Philippe Fraisse. Collaborative manufacturing with physical human–robot interaction. Robotics and Computer-Integrated Manufacturing, 40:1–13, 2016.

[4] Takahiro Endo, Haruhisa Kawasaki, Tetsuya Mouri, Yasuhiko Ishigure, Hisayuki Shimomura, Masato Matsumura, and Kazumi Koketsu. Five-fingered haptic interface robot: Hiro iii. IEEE Transactions on Haptics, 4(1):14–27, 2011.

[5] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional Two-Stream Network Fusion for Video Action Recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1933–1941, jun 2016.

[6] Philippe Fuchs, Guillaume Moreau, and Pascal Guitton. Virtual reality: concepts and technologies. CRC Press, Florida, Florida, USA, Jul 2011.

[7] Vamsi Krishna Guda, Damien Chablat, and Christine Chevallereau. Safety in a human robot interactive: Application to haptic perception. In Virtual, Augmented and Mixed Reality. Design and Interaction, page 562–574, Berlin, Heidelberg, Jul 2020. Springer-Verlag.

[8] A Gutierrez, V Guda, S Mugisha, C Chevallereau, and Damien Chablat. Trajectory planning in dynamics environment : Application for haptic perception in safe humanrobot interaction. 2022.

[9] Mary Hayhoe, Pilar Aivar, Anurag Shrivastavah, and Ryan Mruczek. Visual short-term memory and motor planning. Progress in brain research, 140:349–363, 2002.

[10] ISO Central Secretary. Robots and robotic devices - safety requirements for industrial robots - part i: Robots, din en iso 10218-1, 2021.

[11] Roland S. Johansson, Göran Westling, Anders Bäckström, and J. Randall Flanagan. Eye–hand coordination in object manipulation. Journal of Neuroscience, 21(17):6917–6932, 2001.

[12] Haruhisa Kawasaki and Tetsuya Mouri. Design and control of five-fingered haptic interface opposite to human hand. IEEE Transactions on Robotics, 23(5):909–918, 2007.

[13] Yaesol Kim and Young Kim. Versatile encountered-type haptic display for vr environment using a 7-dof manipulator. In Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2016), 2016.

[14] Hema S. Koppula, Ashesh Jain, and Ashutosh Saxena. Anticipatory planning for human-robot teams. In M. Ani Hsieh, Oussama Khatib, and Vijay Kumar, editors, Springer Tracts in Advanced Robotics, volume 109, pages 453–470. Springer Verlag, Cham, 2016.

[15] Oscar La De Cruz, Florian Gosselin, Wael Bachta, and Guillaume Morel. Contributions to the design of a 6 dof contactless sensor intended for intermittent contact haptic interfaces. In 2018 3rd International Conference on Advanced Robotics and Mechatronics (ICARM), pages 130–135, Singapore, Jul 2018. IEEE.

[16] Philip Long, Christine Chevallereau, Damien Chablat, and Alexis Girin. An industrial security system for human-robot coexistence. Industrial Robot: An International Journal, 45(2):220–226, 2018.

[17] Ruikun Luo, Rafi Hayne, and Dmitry Berenson. Unsupervised Early Prediction of Human Reaching for Human—Robot Collaboration in Shared Workspaces. Auton. Robots, 42(3):631–648, mar 2018.

[18] William A. McNeely. Robotic graphics: a new approach to force feedback for virtual reality. In Proceedings of IEEE Virtual Reality Annual International Symposium, pages 336–341, Seattle, WA, USA, sept 1993. IEEE.

[19] Víctor Rodrigo Mercado, Maud Marchal, and Anatole Lécuyer. Entropia: Towards infinite surface haptic displays in virtual reality using encountered-type rotating props. IEEE Transactions on Visualization and Computer Graphics, 27(3):2237–2243, 2019.

[20] Land Michael, Mennie Neil, and Rusted Jennifer. The roles of vision and eye movements in the control of activities of daily living. Perception, 28(11):1311–1328, 1999. PMID: 10755142.

[21] Stanley Mugisha, Vamsi Krisha Guda, Christine Chevallereau, Matteo Zoppi, Rezia Molfino, and Damien Chablat. Improving haptic response for contextual human robot interaction. Sensors, 22(5), 2022.

[22] Stanley Mugisha, Matteo Zoppi, Rezia Molfino, Vamsi Guda, Christine Chevallereau, and Damien Chablat. Safe collaboration between human and robot in a context of intermittent haptique interface. In ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, 2021.

[23] Jeff Pelz, Mary Hayhoe, and Russ Loeber. The coordination of eye, head, and hand movements in a natural task. Experimental Brain Research, 139(3):266–277, 2001.

[24] C. Pérez-D'Arpino and J. A. Shah. Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification. In 2015 IEEE International Conference on Robotics and Automation (ICRA), pages 6175–6182, Seattle, WA, USA, may 2015.

[25] Jérôme Perret and Pierre Vercruysse. Advantages of mechanical backdrivability for medical applications of force control. In Conference on Computer/Robot Assisted Surgery (CRAS), pages 84–86, Genoa, Italy, Oct 2014.

[26] H. C. Ravichandar and A. Dani. Human intention inference and motion modeling using approximate E-M with online learning. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1819–1824, Hamburg, Germany, sep 2015.

[27] Steeven Villa Salazar, Claudio Pacchierotti, Xavier de Tinguy, Anderson Maciel, and Maud Marchal. Altering the stiffness, friction, and shape perception of tangible objects in virtual reality using wearable haptics. IEEE Transactions on Haptics (ToH), 13(1):167–174, 2020.

[28] Jeroen B. J. Smeets, Mary M. Hayhoe, and Dana H. Ballard. Goal-directed arm movements change eye-head coordination. Experimental Brain Research, 109(3):434–440, 1996.

[29] Alexander Stamenkovic. Do postural constraints affect eye, head, and arm coordination? Journal of neurophysiology, 120(4):2066–2082, 2018.

[30] Stanford Artificial Intelligence Laboratory et al. Robotic operating system.

[31] Marija Tomić, Christine Chevallereau, Kosta Jovanović, Veljko Potkonjak, and Aleksandar Rodić. Human to humanoid motion conversion for dual-arm manipulation tasks. Robotica, 36(8):1167–1187, 2018.

List of Figures

List of Tables