2022 IEEE 18th International Conference on Automation
Science and Engineering (CASE)
August 20-24, 2022. Mexico City, Mexico

2022 IEEE 18th International Conference on Automation Science and Engineering (CASE) | 978-1-6654-9042-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/CASE49997.2022.9926448

Selecting Objects on Conveyor Belts Using Pointing Gestures Sensed by a Wrist-worn Inertial Measurement Unit

Gabriele Abbate, Alessandro Giusti, Antonio Paolillo,
Luca Maria Gambardella, Andrea Emilio Rizzoli and Jérôme Guzzi

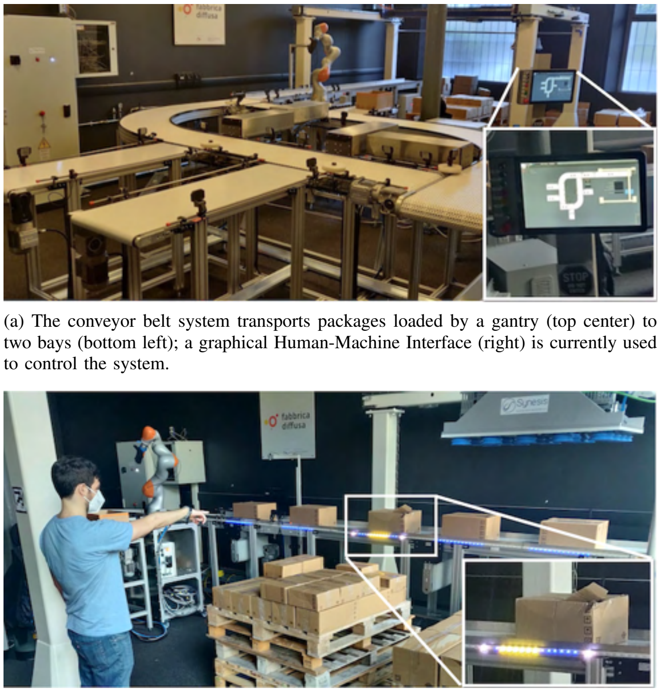
Abstract— **We introduce an intuitive pointing-based interface to select objects moving on a system of conveyor belts. The interface has minimal sensing requirements, as the operator only needs to wear an Inertial Measurement Unit on the wrist (e.g., a smartwatch). LED strips provide the required visual feedback to precisely point to the objects and select them. We test the proposed approach in three environments of different complexity. Experiments compare our approach with a graphical interface where the user clicks on packages with a mouse; quantitative results show that our interface compares favorably, especially in difficult scenarios involving many packages moving fast.**

SUPPLEMENTARY MATERIAL

Code, data, and videos available at `https://github.com/idsia-robotics/pointing-belts`.

I. INTRODUCTION

We consider a system of conveyor belts carrying packages or other objects in an industrial environment, e.g., for logistics applications; these machines can be very large, complex, and highly automated, sometimes involving complex sensing and manipulation; they often operate in environments shared with human operators. In this context, we consider the case in which an operator needs to occasionally interact with the system by *selecting* one or more packages on the belt; for example, an operator, who is normally occupied with a different activity, notices that a package might be damaged or otherwise unusual (Figure 1b): providing this piece of information to the system allows countermeasures to be taken (e.g., route the package to a specific bay for manual inspection).



(a) The conveyor belt system transports packages loaded by a gantry (top center) to two bays (bottom left); a graphical Human-Machine Interface (right) is currently used to control the system.

(b) Using the proposed interface, a user selects a damaged package by pointing to it: the system then unloads the package on a predefined bay; the LED strip provides feedback for pointing (yellow), package tracking (blue), and selection (white dots).

Fig. 1: The industrial demonstrator test-bed (see Section V-C).

For this human-machine-interaction (HMI) task different interfaces might be used. In small, low-automation installations, the operator could just move closer and pick the package up from the moving belt. In larger installations, humans are not allowed to stay close to moving belts, which might carry heavy objects at dangerous speeds. One could then install an ad-hoc button beside the belt, which, when pressed, selects the package passing nearby; this solution requires a fixed infrastructure and needs the operator to be at the right place at the right time. Instead, one could provide a graphical user interface (GUI), on a fixed screen like in Figure 1a or on a tablet held by the operator; the GUI would depict a map of the belt and shows an icon for each package, whose position is updated in real time (see Figure 5a); the operator selects packages by clicking or touching their icon. This requires the operator to figure out which icon corresponds to the real package they want to select: even assuming perfect conditions (i.e., the map orientation matches the operator's viewpoint), this *indirection* step could be very challenging, especially with many fast-moving packages on a large and complex installation.

We propose to remove this indirection step by using a *pointing gesture* to directly indicate the actual package on the belt to be selected. This is a natural and intuitive interface: it is the same approach a human would use to communicate with another human ("*that* package!"). The gesture is perceived through an IMU-equipped bracelet (e.g., a smartwatch) worn by any operator that might need to perform this interaction. Because gestures are inherently imprecise, our system has strips of LEDs installed along the belt to provide real-time feedback to the operator concerning the currently-pointed location, and the current selection state

Authors are with the Dalle Molle Institute for Artificial Intelligence (IDSIA), USI-SUPSI, Lugano, Switzerland, `gabriele.abbate@idsia.ch`.
This work is supported by the European Commission through the Horizon 2020 project 1-SWARM, grant ID 871743.

of different packages.

After an overview of related work (Section II), we provide background information on pointing gestures for human-machine interaction (Section III) and then describe our **main contribution** in Section IV a robust HMI approach for pointing-based selection of packages on a conveyor belt, usable from any location in direct view of the belt, and requiring only minimal infrastructure. Section V describes the setup of three user studies in different environments; in the first, we compare the proposed interface with a graphical interface; in the others, we measure how the proposed interface performs in a challenging setup and in a real installation. Qualitative and quantitative results are reported in Section VI and discussed in Section VII with concluding remarks.

II. Related Work

In industrial environments, human-machine-interaction (HMI) and human-robot-interaction (HRI) have raised an increasing interest over the years [1], with the aim to achieve efficient and safe cooperation between machines, robots and co-located humans. To this end, physical-HRI [2] has been investigated to efficiently detect and react to contacts occurring between robots and humans [3]. Approaches of HMI and HRI that do not involve physical contact are also investigated [4]: these interfaces aim to be intuitive to use [5] and may adopt voice-based dialogue systems [6], or non-verbal approaches, e.g., based on gaze or gestures [7].

In noisy industrial environments, gestures are preferred to speech [8], [9]. Among all the possible gestures, pointing is an innate, intuitive, and practical mean to indicate objects, directions, and locations [10]: therefore, pointing has been used in several robotics applications, e.g., to select a robot [11] or to indicate to the robot a target object [12], [13], location or direction [14], [15]. For industrial applications, pointing has been used along with voice [16] to ease HRI tasks; the analysis carried out by Profanter et al. [17] shows that pointing is an HRI modality well-received by users.

The use of pointing requires a model of human pointing and a method for perceiving the gesture. The geometric pointing model consists in choosing two anatomic points on the user's body defining a *pointing ray*, on which the pointed location lies. *Head-rooted* models assume that the pointing ray starts from a point in the user's head (which could be the dominant eye, the "cyclopic eye" in the middle of the eyes, or the head centroid) and passes through the centroid of the hand or the index finger [18], [19], [12], [20], [21], [22], [23]. In the *arm-rooted* models, the origin of the pointing ray is placed in the user's shoulder, elbow, wrist or at the base of the finger, whereas the second point is normally placed at one of the subsequent links on the arm structure, i.e., on the elbow, wrist, or the tip of the finger [13], [19], [12], [20], [21], [22], [23]. Two sensing configurations are possible to perceive the gesture: an external sensor looking at the user [13], [22], [23], [18]) or wearable sensors on the arm [15], [11].

Our previous work on pointing-based interfaces for conveyor belts has focused on contributing an open source software stack for ROS2 [24], on using Virtual Reality (VR) for experimentation [25], and on integrating with state-of-the-art industrial automation control frameworks [26]. Instead, this work focuses on the methodology and experimental validation.

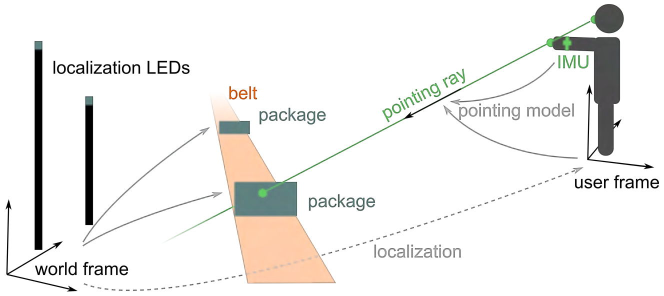


Fig. 2: The pointing ray is reconstructed from IMU readings in the user frame using the pointing model described in Section III. The pointing ray is intersected with objects (green packages on the belt) whose position in the world frame is tracked. The user can localize themselves in the world frame by pointing at localization LEDs in known positions.

III. Background — Pointing-Based HMI

We now introduce the problem of detecting the event in which an operator, wearing an IMU sensor on their wrist, points to an object whose position is known with respect to a fixed *world frame*.

Humans point to an object by standing in such a posture that the pointing ray passes through the object; we adopt a head-finger model in which the pointing ray originates at the user's eyes and passes through their index fingertip. This model forms the basis of our interface: we reconstruct the pointing ray and find the objects intersecting it (see Figure 2); this implies estimating the finger position with respect to the eyes, which we obtain by measuring the forearm orientation (through an IMU worn on the wrist), and applying forward kinematics based on the users' bio-metrics (which we assume known), assuming that the users keep their arm and wrist straight. Neither the human pose in the world frame nor the absolute IMU orientation is assumed to be known at the start of the interaction. The goal of this work is using this interface for object *selection*, which requires first to fulfill three requirements: *triggering*, *localization*, and *feedback*.

a) Triggering: First, the system must understand that the operator wants to point at an object; this can be implemented through explicit (e.g., pressing a button on the wearable device) or implicit triggers (e.g., recognizing a pointing gesture directly from the IMU stream [27], [28]).

b) Localization: To intersect the pointing ray with objects in the world frame, we need to know the pose of the user in the world frame. Assuming the user stands on a floor, and that their height is known, this implies computing a two-dimensional pose (horizontal position and rotation around the common vertical axis). In industrial settings, this localization [29] may rely on smart cameras [30], Radio Frequency networks (in particular Ultra-wideband (UWB) [31]), inertial sensors [32], or, more generally, on a combination of technologies [33]. For this work, we use an alternative method adapted from previous work [34] that relies only on the wrist-worn IMU sensor, by implementing the procedure above in reverse: the operator is asked to point at objects that lie at known world positions; then, the operator is localized at the pose that minimizes the error in the reconstructed pointed positions.[1]

c) Feedback: Pointing gestures are inherently inaccurate; however, if the system provides real-time feedback on the estimated position being indicated, the user can adapt their stance in a closed-loop way to achieve good accuracy. This is similar to moving the mouse pointer to a specific point on a screen, or indicating a small spot with a laser pointer; in both cases, the user acts to minimize the perceived position error (the observed position of the mouse or laser pointer with respect to the target). This mechanism, which is intuitive for users, compensates for several inaccuracies: (i) the head-finger pointing model does not match perfectly human pointing; (ii) users generally point with their arm not perfectly straight; (iii) the IMU is affected by noise and its readings may slightly drift around the vertical axis; (iv) inaccuracies in localization translate to errors in the reconstructed pointed location. *Visual spatial feedback* of the pointed location allows users to be aware and compensate for any misalignment. When interacting with a fast moving robot, such feedback can be provided by the position of the robot itself, while it tracks the pointed location in real time [34]; to interact with a fixed infrastructure, we can use lights to provide such feedback, as we describe in the next section. Previous research [34], [14] shows that a similar setup with real-time feedback of the pointed location yields a median pointing error of $7\,\mathrm{cm}$ at a distance of $2.5\,\mathrm{m}$.

Once all requirements are satisfied, users can start selecting objects or locations, and triggering actions. How this is best implemented depends on the application scenario: for controlling a moving robot, pointing can directly provide way-points to the robot [14]; for interacting with an industrial automation system, users may instead select objects on which predefined actions will be performed: for example, they could first indicate the object and then the location where it should be moved by the system.

IV. POINTING-BASED SELECTION ON A CONVEYOR BELT

We now consider a specific scenario composed of a set of conveyor belts transporting packages to two or more unloading bays. We describe the geometry and topology of the system as a graph of conveyor belts, where, for each edge e, we store the center line of a belt. We assume that the system tracks the location of packages along the belts, i.e., it keeps a list of $\langle i, e, s \rangle$, where: i is the unique identifier of a package; e denotes the belt on which the package currently lies; s is the linear coordinate along the center line of the belt. Tracking of packages may be implemented by an external tracking system or by integrating in time the linear coordinates according to the speed of the belts. We also assume that some of the packages are damaged or otherwise anomalous, and that operators located nearby (but not the system itself) are able to detect them; once an operator notices a damaged package, they can provide the identifier to the system, which then unloads the package to a predefined bay. In order to provide spatial feedback, LED strips are installed all along the sides of the belts.

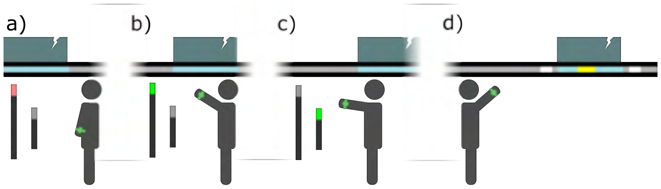


Fig. 3: Interaction of an operator with a linear conveyor belt, whose control system tracks, and displays (cyan) on a LED strip (light grey), the location of packages: a) the operator sees a defective package; b-c) the operator triggers the interaction and starts pointing to two or more subsequently activated LEDs to localize themselves; d) the operator points to the package for a short while to select it, receiving real-time visual feedback about the pointed location (yellow dot) and the selection state (white points).

We now describe how we instantiate the generic pointing-based HMI presented in Section III to provide an interface for the operator, wearing on one wrist a smart device with a button and an IMU, to select packages, as illustrated in Figure 3. One interaction between system and operator follows these steps:

1) The operator notices a defective package.
2) The operator *triggers* the interaction by pressing a button on the smart device.
3) The system starts a *localization* procedure by activating in sequence at least two large LEDs in known and prominent positions, which the operator has to point to; the system receives the IMU stream which it pairs with the location of the active LED to localize the operator; the system notifies when the localization step is completed by switching off all LEDs.
4) Every time it receives an IMU update, the system finds the nearest belt e to the pointing ray and computes the linear coordinate s along e nearest to the ray. Provided that the distance between the ray and the belt is smaller than a threshold, the system projects the pointed location onto the adjacent LED strips, drawing marks as *visual feedback*.
5) The system checks if any package overlaps with the pointed location; in this case, it provides additional visual feedback.
6) The system *selects* the packages that overlap long enough, and provides additional visual feedback that a package is selected.
7) Once a package gets selected, its identifier is communicated to the conveyor belt controller which unloads the package to a predefined bay.

[1] In industrial settings that requires faster reactions (e.g., where objects move very fast), this method may be too slow and localization technologies for real-time operators tracking should be used, such as UWB-based tracking systems.

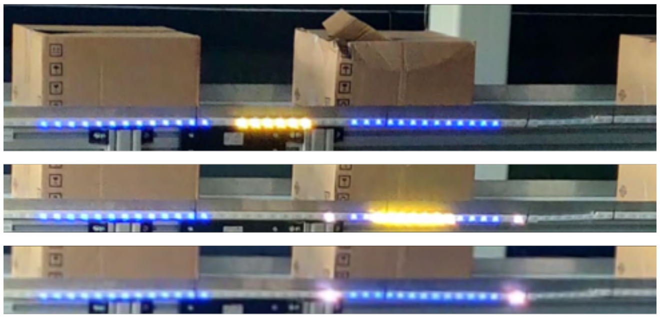


Fig. 4: LED feedback for package selection phases in a real installation. From top to bottom: package unselected, user pointing outside; user pointing inside, package selected; user pointing away, package still selected.

V. EXPERIMENTAL SETUP

We test the pointing-based interface introduced in Section IV in the three environments illustrated in Figure 5: a simple setup in our *Laboratory*, a more complex setup in *VR*, and a *Demonstrator* at a real facility; we compare it against a graphical interface with mouse input. The supplementary video captures several experimental runs for each environment.

We focus on *selection*: we investigate how well different subjects select packages using one of the interfaces, and how difficult they perceive this task to be. To this aim, except in the real facility, we ignore the other interaction phases by having users stand at a predetermined location and trigger the interaction automatically. Subjects gave their consent to collect the data we report in Section VI. In the remaining of this section, we first introduce the commonalities and then detail the specificity of each environment.

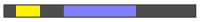
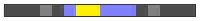
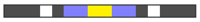
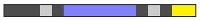
Tracking: Packages moving on the belt are tracked; to mark them, we draw a blue line on the adjacent LED strip (see Figure 3).

Task: Some of the packages traveling on the belt are easily identified as anomalous by observers (e.g., they are red). The users have to select all of them and none of the normal packages, as fast as possible; they are allowed to deselect packages, in case they notice a mistake.

Selection feedback: For both interfaces, we display the selection state of packages on the LED strips: we draw bright white dots flanking each selected package. We also play sounds each time a package gets selected or unselected.

Graphical interface: A GUI (see Figure 5a) — mimicking common Human-Machine Interfaces available in industrial settings (Figure 1a) — provides a live view of the location of packages on the belts, drawn as large disks. The user selects or unselects packages by clicking on them with a mouse. To allow a fair comparison, the GUI leverages the same environmental information as the pointing-based interface, i.e., just the location of packages, therefore all disks are drawn in the same color, with selected disks darker.[2]

Pointing-based interface: Users wear an IMU (Metawear's MetaMotionR) on one wrist; when the user points to a package long enough (more than $0.7\,\mathrm{s}$), the package toggles its state between unselected and selected (as if the user would click on it on the GUI). The LED strips provide two additional pieces of visual feedback: the current pointed location (a short bright yellow segment); dim white dots flanking the package (if any) overlapping the currently pointed location. Using the pointing-based interface, the selection of one package goes through following steps:

1) the package is unselected, the user is pointing outside of the package 
2) the user starts pointing inside the package 
3) after a time interval τ the package gets selected 
4) the user stops pointing to the package: the package stays selected 

By repeating the steps (starting by pointing outside), the user would deselect the package. All experiments are performed with $\tau = 0.7\,\mathrm{s}$. Figure 4 shows steps 1, 3 and 4 in the *Demonstrator* setup.

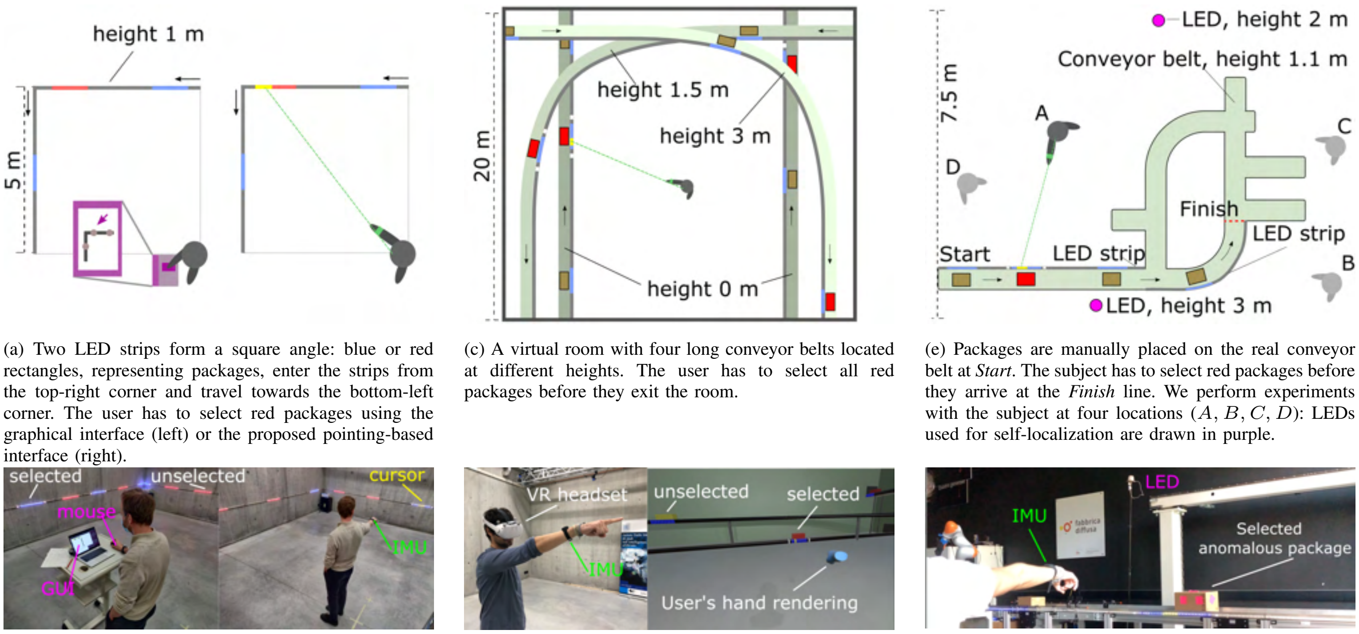
A. Laboratory

To flexibly test various scenarios with multiple users in a safe environment, we run the first part of our user study on an emulated rather than a real conveyor belt. We emulate packages traveling on a conveyor belt by drawing lines on LED strips: defective packages are colored in red, while normal packages are in blue. Therefore, the LED strips play a double role in this setup: provide feedback as described above, and display packages; to guarantee a fair comparison with the GUI, we separate the two roles as much as possible: the displayed color of packages is a constant intrinsic property of the packages, and visual feedback is independent of it. Figure 5a illustrates the setup.

Experiment execution: We perform tests with 8 user subjects with no previous experience in any of the two interfaces, but all familiar with generic mouse-based interfaces. For each subject, we run a range of experiments where they use one of the two interfaces to select all red packages while standing at a fixed place; we tune the difficulty of the task by 1) the number of packages: N1 (1 red, 2 blue), N2 (2 red, 4 blue), N3 (3 red, 6 blue); and 2) the speed at which packages travel: static, low ($0.2\,\mathrm{m\,s^{-1}}$), medium ($0.5\,\mathrm{m\,s^{-1}}$), fast ($0.9\,\mathrm{m\,s^{-1}}$). For each subject, we test 6 increasing difficulty levels in order: 1) N1 static, 2) N2 static, 3) N3 static, 4) N3 at low speed, 5) N3 at medium speed, 6) N3 at high speed. For each difficulty level, we execute 5 consecutive runs with one interface and 5 consecutive runs with the other. To equalize learning effects, 4 subjects

[2] We note that such a graphical Human-machine Interface could be made more powerful at addressing specific industrial requirements, for instance displaying a top-down video overlay, using machine-vision to identify defective packages, or exploiting augmented reality. We limit our scope to interfaces with minimal environmental information and minimal equipment carried by operators.

	Laboratory	Virtual Reality	Demonstrator
Environment:	$9 \times 9\,\mathrm{m}$ real	$20 \times 20\,\mathrm{m}$ VR	$15 \times 10\,\mathrm{m}$ real
Belt:	$10\,\mathrm{m}$, emulated, simple topology	20+20+35+35 m, complex topology	$22\,\mathrm{m}$, medium complexity
Sensing:	real IMU	real IMU	real IMU
Localization:	known	known	estimated (at 4 different locations)
Operators:	8	3	1
Belt speed:	$0\,\mathrm{m/s}$ to $0.9\,\mathrm{m/s}$	$0.5\,\mathrm{m/s}$ to $10\,\mathrm{m/s}$	$0.25\,\mathrm{m/s}$



(a) Two LED strips form a square angle: blue or red rectangles, representing packages, enter the strips from the top-right corner and travel towards the bottom-left corner. The user has to select red packages using the graphical interface (left) or the proposed pointing-based interface (right).

(c) A virtual room with four long conveyor belts located at different heights. The user has to select all red packages before they exit the room.

(e) Packages are manually placed on the real conveyor belt at *Start*. The subject has to select red packages before they arrive at the *Finish* line. We perform experiments with the subject at four locations (A, B, C, D): LEDs used for self-localization are drawn in purple.

(b) Two snapshots from experimental runs in the laboratory at difficulty N3 static: left, using the GUI, and right, using the pointing-based interface.

(d) Left: the subject, wearing a VR headset and an IMU bracelet, interacts with the simulated system in VR. Right: the first-person-view experienced by the subject.

(f) A snapshot from the experimental run where the subject is at location A. One of the two LEDs used for self-localization is visible at the top.

Fig. 5: Experimental environments described in Section V. The top row shows top-down maps of the environments (where we draw pointing rays in green), the middle row displays images from experimental runs, and the last row summaries the main characteristics of the environments. In all environments, the LED strips display visual feedback: tracked packages in blue, a yellow pointing cursor, and white marks enclosing selected packages.

(picked randomly) first use the GUI and then pointing, while the remaining 4 use the opposite order. In total, the user study comprises $8 \times 6 \times 2 \times 5 = 480$ runs. At the beginning of each run, we randomly draw the position of the packages. Once the packages appear on the strip, the subject can start selecting them as depicted in Figure 5b. We record the times at which packages get selected or deselected. The run terminates when all and only red packages are selected. At the end of the 5 runs for each interface for each difficulty level, we ask the subject about the perceived task difficulty using a scale ranging from 0 (“extremely easy”) to 10 (“extremely hard”); therefore, we collect a total of $8 \times 6 \times 2 = 96$ difficulty ratings.

B. Virtual Reality

In order to test the proposed interface in a more complex and challenging setup, we built a simulated environment (see Figure 5c) with four long conveyor belts (two straight and two with a gentle 90 degrees curve), located at different heights and positions in a room of $20\,\mathrm{m}$ of side, which transport normal and anomalous (red) packages. The subjects wear a VR headset to immerse themselves in the simulation [25], standing at the center of the room. The pointing-based interface uses the exact same sensing (IMU bracelet) and software as in the other, real-world, setups. The simulated belt is equipped with LED strips that accurately replicate the visual feedback of real LED strips, in order to minimize the reality gap and let subjects experience a very realistic interaction.

Experiment execution: We perform tests with 3 subjects that use the pointing-based interface to select packages. Similarly to the previous setup, for each subject, we perform runs with increasing speed of the belts, ranging over 20 runs from $0.5\,\mathrm{m\,s^{-1}}$ to $10\,\mathrm{m\,s^{-1}}$. At the beginning of each run, the belts are empty; then, for $20\,\mathrm{s}$, we add packages at the beginning of the belts, randomly drawing their color ($2/3$ of chance for normal and $1/3$ for red), leaving a gap of $4\,\mathrm{s}$ to $6\,\mathrm{s}$ between packages on the same belt: the subject has to select red packages as soon as possible, non-selected red packages that exit the room represent a failure. We record the times at which packages enter and exit the system, are selected, or deselected. Each run terminates after all packages have exited. In total, this experiment covers more than 30 minutes of interaction.

C. Demonstrator

Fabbrica Diffusa [35] is a set of demonstrative setups to experiment and showcase innovation linked to Industry 4.0. One of them is hosted by the innovation hub Como-Next and features a small but complete de-palletizing demonstrator for distribution centers (see Figure 1) consisting of: a gantry robot to unload packages from a pallet; a conveyor belt to distribute them, through diverters, to different bays; and a robotic arm to pick objects from the packages. The belt runs at a constant speed of $0.25\,\mathrm{m\,s^{-1}}$. Packages are tracked by integrating their initial position according to the belt speed, using light-traps as correction. We mounted LED strips on the initial segments of the conveyor belt (see Figure 5e).

Experiment execution: In this more realistic environment, we tested the whole proposed pointing-based interaction with one subject: from the subject self-localizing, using the procedure described in Section III, to they selecting a package.[3] As illustrated in Figure 5e, at the beginning of each run, the subject is located at a position unknown to the system. We manually place packages at the start of the conveyor belt, one of them marked in red. Once the subject notices a red package, they have to first localize themselves by pointing at two fixed LEDs (a procedure that takes about 4 seconds) and then to select the package before it arrives in front of the first bay. We record if they are able to do so and the time it takes. We perform a total of 8 runs, i.e., two per location.

VI. Results

A. Laboratory

1) Quantitative task performance: Figure 6 compares the graphical and proposed pointing-based interfaces over all 8 subjects, reporting separate metrics for each of the 6 difficulty levels. Figure 6a reports the completion time of each run, defined as the time elapsed since the start of the run at which all the packages have the expected selection state. Figure 6b reports the average number of mistakes per run; a mistake is defined as the event of selecting a blue box or deselecting a selected red box. These events sometimes happen as subjects click on the wrong box (when using the graphical interface), or mistakenly hover on a box they did not intend to select/deselect (when using the pointing interface).

We observe that as the scenario gets harder, the completion time increases; in static scenarios, using a GUI is marginally better than pointing; in the hardest scenario (N3 at fast speed), the graphical interface exhibits higher completion time and significantly increased mistakes; conversely, the performance with the pointing interface does not degrade; this suggests that the pointing interface might scale better to harder tasks, in particular in presence of long conveyor belts with many packages.

TABLE I: Laboratory experiment: data on each subject, and perceived task difficulty (from 0 to 10) averaged for static and moving scenarios.

			Static		Moving	
Subject	Sex	Age	GUI	Pointing	GUI	Pointing
1	M	39	3.0	3.0	5.3	3.3
2	F	26	2.7	3.3	9.0	6.3
3	M	29	3.7	4.0	4.7	4.3
4	M	27	2.0	2.3	2.0	2.0
5	M	40	2.0	3.0	4.7	3.0
6	M	21	2.7	1.3	5.7	5.0
7	M	28	7.3	3.3	5.0	8.7
8	F	26	1.3	1.0	2.0	1.7

2) Quantitative analysis of user surveys: Figure 6c shows the difficulty rating that subjects reported immediately after completing the 5 runs of each scenario; with both interfaces, moving scenarios are perceived as more difficult than static scenarios; differences between interfaces consistently favour the pointing interface except in the easiest scenario (N1, static), which is also the first on which each subject is tested. This could be explained as a learning effect: while the graphical interface uses a mouse and screen – a familiar interface for all users – the pointing interface is for most subjects the first experience with pointing-based user interfaces, and thus causes some initial confusion; reported difficulty decreases already in the second experience with the pointing interface.

In static scenarios, with the graphical interface, reported difficulty increases as more packages are displayed; in fact, it is increasingly difficult for subjects to relate the packages seen in the world to those represented on screen; this is an indirection step intrinsic to any approach representing packages on-screen; the pointing interface does not require this indirection step, and does not exhibit the same increase in perceived difficulty for static scenarios; it is reasonable to expect that the gap would become wider as the number of packages further increases.

The absolute difficulty scores given by different users have a large variance, as some users tend to give generally higher scores than others; this explains the wide confidence intervals in Figure 6c, which reports the average scores over all users. To better compare the interfaces, we rely on the fact that each subject evaluated both interfaces: Figure 6d shows the distribution of the *difference* in the difficulty reported by each user for the pointing vs the graphical interface. Values above 0 mean that the pointing interface was reported to be harder than the GUI; values below 0 imply that it was easier. In this case, we observe that confidence intervals are narrower. Pooling data from all scenarios, the pointing interface is found to be less difficult in a statistically significant[4] way ($p = 0.033$); the same conclusion holds if we only consider the moving scenarios ($p = 0.030$); in contrast, when considering only

[3] In the presented experiments, we limited ourselves to testing the interface, without actually controlling the automation system. See [26] about how to interface with the automation control system to close the loop.

[4] We use the one-tailed paired non-parametric Wilcoxon signed-rank test; it tests the null hypothesis (the distribution of the differences in reported difficulty is symmetric around 0) against the alternative hypothesis that the pointing interface yields lower difficulty scores on average

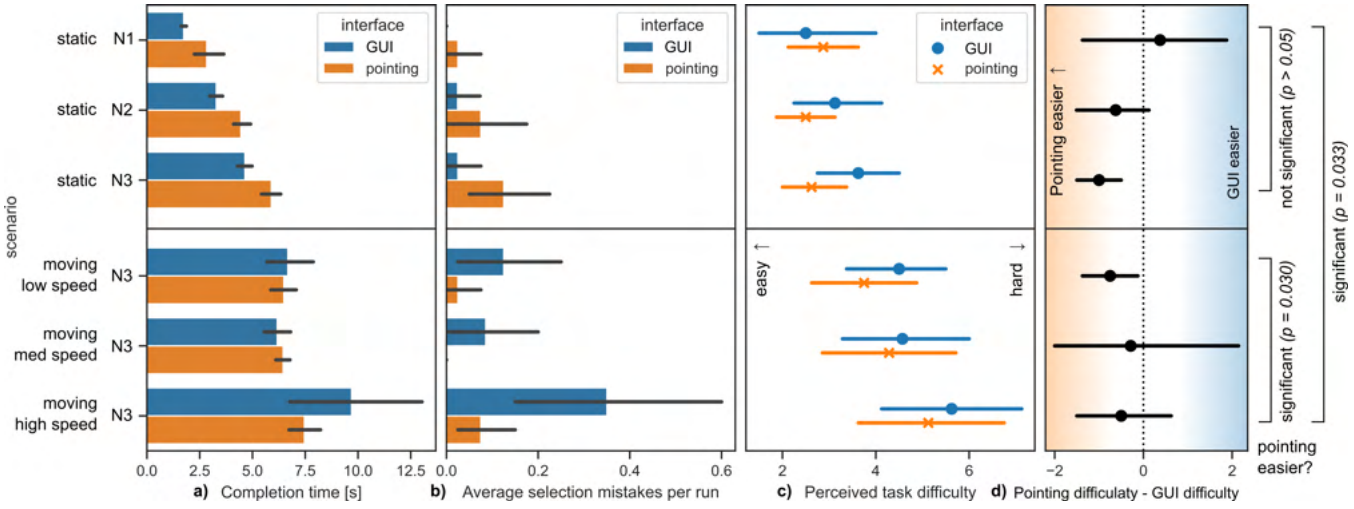


Fig. 6: Laboratory experimental results: from left to right: completion time; number of mistakes per run; difficulty rating given by subjects; difference between difficulty assigned to pointing-based and graphical interfaces. For each scenario (rows), reported values are averaged over the 8 subjects; in the two leftmost plots, each bar considers 5 runs per subject (N=40); in the two rightmost plots, we have one grade per subject per interface (N=8). In the three left plots, lower is better; in the rightmost plot, negative values denote better performance for pointing compared to the GUI. All error bars depict 90% confidence intervals for the mean.

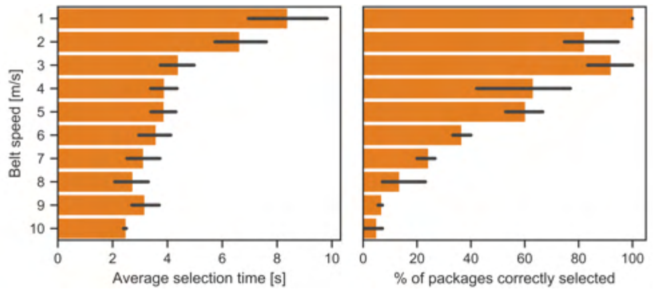


Fig. 7: VR experimental results averaged over all users. Left: mean time between a red package appearance and its selection. Right: fraction of packages that are correctly selected by the end of the run.

TABLE II: Demonstrator experimental results

User locations	Runs	Failures	Median interaction duration [s]
4	8	1	10.5

static scenarios, the difference among the two interfaces is not statistically significant ($p > 0.05$)

B. Virtual Reality)

Figure 7 summarizes the results on the VR environment. As the task gets more difficult, failures increase as expected, while selection time initially decreases, as users get more attentive and react faster, and then stabilizes at $2\,\mathrm{s}$. The interface remains usable for all subjects up to about a speed of $6\,\mathrm{m\,s^{-1}}$. For higher speeds, we note a difficulty to point accurately long enough time to select the package, as well as the intrinsic challenge of keeping track of all fast-moving packages. Subjects report that the interface is intuitive and suitable for the task.

C. Demonstrator

In Table II we record one failure (when the subject was in location B) out of 8 trials due to a large error in localization. For all the other runs, the subject had no difficulty in selecting the package, averaging about $10\,\mathrm{s}$ from when they start interacting (by triggering the self-localization procedure) to when the package is marked as selected. The subject rated the interface as appropriate to the task with sufficient visual feedback from the LED strips.

VII. Discussion and Conclusions

We designed and implemented an interface for selecting packages on conveyor belts using pointing gestures sensed by a wrist-worn IMU, and experimentally validated it against a GUI, which was designed in such a way to be as easy to use and suitable to the task as possible. We found that, despite being an unfamiliar system for our subjects, our interface is perceived as even easier to use than a GUI, and is competitive in terms of efficiency especially in challenging scenarios with many fast-moving packages. The experiments in a real facility, although limited, show promising results towards the deployment of the proposed solution.

The main advantage of our approach is that it does not need an indirection step between a screen and the real world: when the operator sees a package, they don't have to find its representation on the screen to select it. This indirection step becomes especially challenging with long conveyor belts, with complex topology and many fast-moving packages; the experiments in VR suggest that our pointing-based interface is suitable to handle such demanding installations. A further advantage is that our approach requires minimal infrastructure, is usable from any location with direct line-of-sight to a part of the conveyor belt, and does not require the use of handheld devices.

The main drawback of the current implementation is that confirming the selection of a package requires hovering on it for a short time; this penalizes performance in simple scenarios compared to the GUI, which requires a short mouse click to select a package. In scenarios where handheld devices are acceptable, using a button (rather than hovering) to select a package is a good option. When packages are very fast, the current selection criterion is too strict: the system should relax it when the risk of confusion (e.g., between two packages) is low.

As a further extension to the system, we are planning to use additional gestures (iconic or pointing) to express commands to be executed on the selected package(s): for example, after selection is completed the operator could point to a specific plant bay; this could trigger actions that take the selected packages to that bay.

References

[1] J. Berg and S. Lu, "Review of interfaces for industrial human-robot interaction," *Current Robotics Reports*, vol. 1, no. 2, pp. 27–34, 2020.

[2] A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi, "An atlas of physical human–robot interaction," *Mechanism and Machine Theory*, vol. 43, no. 3, pp. 253–270, 2008.

[3] S. Haddadin, A. De Luca, and A. Albu-Schäffer, "Robot collisions: A survey on detection, isolation, and identification," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1292–1312, 2017.

[4] C. Breazeal, A. Takanishi, and T. Kobayashi, "Social robots that interact with people," in *Springer Handbook of Robotics*. Springer, 2008, pp. 1349–1369.

[5] E. A. Kirchner, J. de Gea Fernandez, P. Kampmann, M. Schröer, J. H. Metzen, and F. Kirchner, *Intuitive Interaction with Robots – Technical Approaches and Challenges*. Springer, 2015, pp. 224–248.

[6] H. Chen, X. Liu, D. Yin, and J. Tang, "A survey on dialogue systems: Recent advances and new frontiers," *ACM Sigkdd Explorations Newsletter*, vol. 19, no. 2, pp. 25–35, 2017.

[7] N. Mavridis, "A review of verbal and non-verbal human–robot interactive communication," *Robotics and Autonomous Systems*, vol. 63, pp. 22–35, 2015.

[8] S. Sheikholeslami, A. Moon, and E. A. Croft, "Cooperative gestures for industry: Exploring the efficacy of robot hand configurations in expression of instructional gestures for human–robot interaction," *The International Journal of Robotics Research*, vol. 36, no. 5-7, pp. 699–720, 2017.

[9] B. Gleeson, K. MacLean, A. Haddadi, E. Croft, and J. Alcazar, "Gestures for industry intuitive human-robot communication from human observation," in *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2013, pp. 349–356.

[10] G. Butterworth, "Pointing is the royal road to language for babies," in *Pointing: Where Language, Culture, and Cognition Meet*, 2003.

[11] B. Gromov, L. M. Gambardella, and G. A. Di Caro, "Wearable multimodal interface for human multi-robot interaction," *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 240–245, Oct 2016.

[12] D. Droeschel, J. Stückler, and S. Behnke, "Learning to interpret pointing gestures with a time-of-flight camera," *ACM/IEEE International Conference on Human-robot Interaction (HRI)*, pp. 481–488, 2011.

[13] B. Großmann, M. R. Pedersen, J. Klonovs, D. Herzog, L. Nalpantidis, and V. Krüger, "Communicating Unknown Objects to Robots through Pointing Gestures," in *Annual Conference on Advances in Autonomous Robotic Systems (TAROS)*. Springer, 2014, pp. 209–220.

[14] B. Gromov, L. M. Gambardella, and A. Giusti, "Guiding quadrotor landing with pointing gestures," in *12th International Workshop on Human Friendly Robotics*. Springer, Oct 2019.

[16] I. Maurtua, A. Ibarguren, J. Kildal, L. Susperregi, and B. Sierra, "Human–robot collaboration in industrial applications: Safety, interaction and trust," *International Journal of Advanced Robotic Systems*, vol. 14, no. 4, 2017.

[15] M. T. Wolf, C. Assad, M. T. Vernacchia, J. Fromm, and H. L. Jethani, "Gesture-based robot control with variable autonomy from the JPL BioSleeve," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1160–1165, 2013.

[17] S. Profanter, A. Perzylo, N. Somani, M. Rickert, and A. Knoll, "Analysis and semantic modeling of modality preferences in industrial human-robot interaction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 1812–1818.

[18] K. Plaumann, M. Weing, C. Winkler, M. Müller, and E. Rukzio, "Towards accurate cursorless pointing: the effects of ocular dominance and handedness," *Personal and Ubiquitous Computing*, vol. 22, no. 4, pp. 633–646, Dec. 2017.

[19] K. Nickel and R. Stiefelhagen, "Pointing Gesture Recognition based on 3D-Tracking of Face , Hands and Head Orientation Categories and Subject Descriptors," *International Conference on Multimodal interfaces*, pp. 140–146, 2003.

[20] S. Mayer, K. Wolf, S. Schneegass, and N. Henze, "Modeling Distant Pointing for Compensating Systematic Displacements," in *ACM Conference on Human Factors in Computing Systems (CHI)*, vol. 1, 2015, pp. 4165–4168.

[21] A. Cosgun, A. J. B. Trevor, and H. I. Christensen, "Did you Mean this Object?: Detecting Ambiguity in Pointing Gesture Targets," in *HRI Workshop Towards a Framework for Joint Action*, 2015.

[22] K. Kondo, G. Mizuno, and Y. Nakamura, "Analysis of human pointing behavior in vision-based pointing interface system - difference of two typical pointing styles," *IFAC-PapersOnLine*, vol. 49, no. 19, pp. 367–372, 2016.

[23] S. Mayer, V. Schwind, R. Schweigert, and N. Henze, "The effect of offset correction and cursor on mid-air pointing in real and virtual environments," in *ACM Conference on Human Factors in Computing Systems (CHI)*, Apr. 2018.

[24] G. Abbate, A. Giusti, A. Paolillo, B. Gromov, L. M. Gambardella, A. E. Rizzoli, and J. Guzzi, "PointIt: A ROS toolkit for interacting with co-located robots using pointing gestures," in *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2022, pp. 608–612.

[25] J. Guzzi, G. Abbate, A. Paolillo, and A. Giusti, "Interacting with a conveyor belt in virtual reality using pointing gestures," in *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2022, pp. 1194–1195.

[26] A. Paolillo, G. Abbate, A. Giusti, S. Trakić, H. Dzafic, A. Fritz, and J. Guzzi, "Towards the integration of a pointing-based human-machine interface in an industrial control system compliant with the iec 61499 standard," *Procedia CIRP*, vol. 107, pp. 1077–1082, 2022.

[27] D. Broggini, B. Gromov, L. M. Gambardella, and A. Giusti, "Learning to detect pointing gestures from wearable IMUs," in *AAAI Conference on Artificial Intelligence*. AAAI Press, Feb 2018.

[28] G. Abbate, B. Gromov, L. M. Gambardella, and A. Giusti, "Pointing at moving robots: Detecting events from wrist IMU data," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[29] D. Dardari, P. Closas, and P. M. Djurić, "Indoor tracking: Theory, methods, and technologies," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1263–1278, 2015.

[30] S. Fleck, F. Busch, P. Biber, and W. Straber, "3d surveillance a distributed network of smart cameras for real-time tracking and its visualization in 3d," in *Conference on Computer Vision and Pattern Recognition (CVPRW)*, 2006, pp. 118–118.

[31] M. Kuhn, C. Zhang, B. Merkl, D. Yang, Y. Wang, M. Mahfouz, and A. Fathy, "High accuracy uwb localization in dense indoor environments," in *IEEE International Conference on Ultra-Wideband*, vol. 2, 2008, pp. 129–132.

[32] E. M. Diaz, F. de Ponte Müller, A. R. Jiménez, and F. Zampella, "Evaluation of ahrs algorithms for inertial personal localization in industrial environments," in *IEEE International Conference on Industrial Technology (ICIT)*, 2015, pp. 3412–3417.

[33] J. A. Corrales, F. Candelas, and F. Torres, "Hybrid tracking of human operators using imu/uwb data fusion by a kalman filter," in *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2008, pp. 193–200.

[34] B. Gromov, G. Abbate, L. M. Gambardella, and A. Giusti, "Proximity human-robot interaction using pointing gestures and a wrist-mounted IMU," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 8084–8091.

[35] "Fabbrica diffusa," https://www.comonext.it/laboratori/, accessed: 2021-09-01.