



HAL
open science

Target Expansion in Context: the Case of Menu in Handheld Augmented Reality

Patrick Perea, Denis Morand, Laurence Nigay

► **To cite this version:**

Patrick Perea, Denis Morand, Laurence Nigay. Target Expansion in Context: the Case of Menu in Handheld Augmented Reality. AVI '20: International Conference on Advanced Visual Interfaces, Sep 2020, Ischia Island, Italy. pp.1-9, 10.1145/3399715.3399851 . hal-02960631

HAL Id: hal-02960631

<https://hal.science/hal-02960631v1>

Submitted on 7 Oct 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Target Expansion in Context: the Case of Menu in Handheld Augmented Reality

Patrick Perea^{1,2}

¹ Univ. Grenoble Alpes, CNRS,
Grenoble INP, LIG
F-38000 Grenoble, France
patrick.perea@etu.univ-grenoble-
alpes.fr

Denis Morand²

² Schneider Electric
F-06510 Carros, France
denis.morand@se.com

Laurence Nigay¹

¹ Univ. Grenoble Alpes, CNRS,
Grenoble INP, LIG
F-38000 Grenoble, France
laurence.nigay@univ-grenoble-
alpes.fr

ABSTRACT

Target expansion techniques facilitate pointing by enlarging the effective sizes of targets. As opposed to the numerous studies on target expansion solely focusing on optimizing pointing, we study the compound task of pointing at a Point of Interest (POI) and then interacting with the POI menu in handheld Augmented Reality (AR). A POI menu in AR has a fixed position because it contains relevant information about its location in the real world. We present two techniques that make the cursor jump to the closest opened POI menu after pointing at a POI. Our experimental results show that 1) for selecting a POI the expansion techniques are 31 % faster than the baseline screen-centered crosshair pointing technique, 2) the expansion techniques with/without a jumping cursor to the closest opened POI menu offer similar performances and 3) Touch relative pointing is preferred by participants because it minimizes physical movements.

CCS CONCEPTS

• **Human-centered computing** → **Mixed / augmented reality**

KEYWORDS

Handheld Augmented reality, Target expansion, Pointing, Menu

ACM Reference format:

Patrick Perea, Denis Morand, Laurence Nigay. 2020. Target Expansion in Context: the Case of Menu in Handheld Augmented Reality. In *Proceedings of International Conference on Advanced Visual Interfaces (AVI'20)*. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3399715.3399851>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

AVI '20, September 28-October 2, 2020, Salerno, Italy
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7535-1/20/09...\$15.00
<https://doi.org/10.1145/3399715.3399851>

1 Introduction

A common task in handheld Augmented Reality (AR) is to access digital information linked to physical objects. We illustrate this task with two use cases.

Use case 1 – Supermarket/Museum: To get additional information about a product sold in a supermarket or a painting exposed in a museum, users select Points of Interest (POIs), i.e. digital objects placed near the real objects they refer to. A POI enables access to several types of information such as the nutritional table of a food product, the history of a piece of art etc. When a POI is selected, the list of pieces of information is displayed as a menu and users can select one of them.

Use case 2 - Industrial maintenance: To examine the state of production equipment, machine operators access real time information such as the temperature of a welding oven or the current production rate of the machine. To visualize the information, the operators select Points of Interest (POIs). A POI enables access to several types of technical information related to the corresponding physical spot of the machine. When a POI is selected, the list of pieces of information is displayed as a menu and the operator can select one of them.

These two use cases highlight that in handheld AR the users very often need to (1) select POIs to acquire updated data about the augmented object, (2) when a POI is opened, select the desired piece of information, and (3) close the POIs once the task is completed. Thus, to access information in handheld AR, pointing at augmented content (e.g. POIs) is an elementary interactive task universally present.

In handheld AR, pointing techniques are impaired by the instability of the augmented scene that the users interact with on screen. Indeed different factors as defined in [23] can impair the stability of the augmented scene on screen: first natural hand tremor leads to small changes in the camera's viewpoint making the on-screen content unstable. Second, the underlying device's tracking system may lead to incorrect knowledge of the camera's location in the 3D space. The augmented content may then not be accurately anchored in the physical world, and lead to jittery registration. Finally, holding the device while interacting with it may lead to instability when touching the screen [23]. To overcome these limitations, we adopt a target expansion approach that facilitates the selection of POIs by enlarging them. The paper contributes: a) a target expansion technique that dynamically expands POIs in handheld AR, b) Two jumping

cursor techniques that enhance the selection of menu items when a POI is selected/opened c) a study in a pseudo-industrial environment that compares the POI expansion technique, the POI expansion techniques with jumping cursors and the baseline technique, i.e. a screen-centered crosshair technique. This contribution goes beyond previous work (1) by applying a target expansion technique for the case of handheld AR pointing and quantifying the benefits 2) by studying the next interaction step after optimizing POI pointing (i.e. selecting an item in the menu of the selected POI).

In this paper we first review related work and then present the design of our POI expansion technique. Putting our POI expansion in interaction context we then present the design alternatives for the next interaction step after pointing at a POI. We then report an experiment comparing three design alternatives. We conclude with a discussion of our results and directions for future work.

2 Related Work

2.1 Handheld AR Pointing

Based on the analytical framework for the design of interaction techniques for handheld AR presented in [24], pointing is usually performed with either direct touch input (with a pen or bare fingers) on screen [18, 20] or with a screen-center crosshair [19, 20]. Unlike direct-touch on screen, crosshair is not impaired by occlusion or an ambiguous active point. Moreover, a screen-center crosshair supports hover interaction (a missing interaction state with standard touch devices today that do not sense hover information). In both cases (direct touch and screen-centered crosshair) and as explained above pointing techniques at augmented content (e.g. POI) are impaired by the instability of the augmented scene (hand tremor, registration jitter) that the users interact with on screen. In comparison with the issues raised by mobile graphical interaction, the interaction issues in handheld AR are more important because objects on screen do not remain still on the screen during interaction.

Techniques have been designed and refined to overcome these limitations. These techniques can be divided into two main strategies: freeze-frame techniques and techniques based on stabilizing input in the frame of reference of the physical object. Freeze-frame techniques freeze the live view of the camera. This strategy allows users to work in a comfortable pose to interact with the augmented scene. However, freezing the frame leads to two issues: (1) real-time update about the real world is lost. This can be problematic when POIs are highly connected to what can be seen in the real world as for instance for industrial maintenance. (2) Users may be disoriented when resuming the live view of the camera, as the viewpoint has changed. Instead of a freezing mode, Shift&Freeze [24] is a technique that extends direct touch pointing with a precise quasi-mode based on freezing the frame.

A second strategy to overcome the instability of on-screen content, is to use a cursor stabilized in the frame of reference of the physical object, namely the Relative Pointing technique [24].

The cursor is no longer impaired by viewpoint instability. Changing the camera's viewpoint automatically modifies the cursor's position on screen, but not in the physical environment. The cursor's position in the physical world is modified through finger movements on screen (indirect relative touch inputs). The cursor is no longer centered on screen. Results from an experimental study [23] show that Relative Pointing was less error prone than the screen-centered crosshair technique. Moreover, Relative Pointing extends the screen-centered crosshair technique. Both techniques can be fruitfully combined: the screen-centered crosshair technique for coarse and fast pointing and then Relative Pointing if accuracy is needed.

The two above strategies for pointing assistance techniques are target-agnostic: they do not strictly require target (e.g. POI) awareness to operate. In our study we adopt a third strategy that is target-aware: target expansion techniques [15].

2.2 Target expansion

For a single target expansion technique, Yang et al. [26] explore a direct touch variant of expanding widgets [15]. The technique detects the finger proximity above the display surface, and expands the widget as the finger approaches. The technique relies on hover sensing.

For multiple targets, expansion techniques rely on a partitioning of the motor space. One layout strategy is the Voronoi tessellation [8] that maximizes the use of empty space between targets and is unambiguous since only one target is contained in each Voronoi cell. Several target expansion techniques [9, 11, 16, 21] are based on the Voronoi tessellation as a static decomposition of the motor space. Like other pointing techniques based on increasing the sizes of targets, Voronoi-based target expansion techniques depend on the amount of empty space. Target expansion has been first studied for pointing techniques for desktop pointing [11] and distant pointing [10]. Guillon et al. [10] classify existing target expansion techniques according to the provided visual aid. The visual aid is key since the users rely on the visual aid to take full advantage of the target expanded areas.

More closely related to our study, Baudish et al. [5] study target expansion on a mobile device. Starburst [5] is a target expansion technique that has been tested on a tablet computer. Starburst addresses the issue of dense clusters of targets where a Voronoi tessellation would lead to small cells. Starburst proposes a new partitioning of the motor space that allocates space on the outskirts of the cluster to all targets. Starburst defines a static targeting assistance by directly displaying the expanded target boundaries on screen. It is then possible to apply this technique to both direct touch and screen-centered crosshair pointing techniques.

As opposed to the above target expansion studies solely focusing on optimizing pointing, the literature on cascading pull-down menus considers interaction after pointing to a first-level item. This is relevant for our study that not only focuses on optimizing POI pointing but considers the next interaction step after selecting a POI: selecting an item in the POI menu.

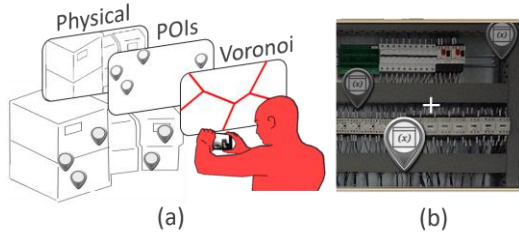


Figure 1: POI expansion technique in handheld AR.

Kobayashi&Igarashi consider overlapping menus [13]: the sub-menu jumps to the current cursor position making the submenus overlap the parent menus.

Bubbling menus [22] enhance the selection of menu items by expanding them based on the bubble cursor [11]. To facilitate the selection of a second-level item, the submenu is moving vertically, following the movement of the cursor in the parent menu. Beyond moving the submenu and thus changing the graphical layout, other strategies fully or partially control the cursor. With Jumping menus [3] a selection of a first-level menu item opens up a submenu and lets the cursor jump to the center of the first submenu item. Instead of fully controlling the cursor, Alhlström [2] proposes to partially overrule the user's control of the cursor by applying a virtual force which pushes the cursor toward the first submenu item. A last strategy that does not change the graphical layout and lets the users fully control the cursor relies on multiple cursors. For instance the Satellite Cursor [27] associates every target with a separate cursor in its vicinity for pointing.

3 POI Expansion: Design

A target-expansion approach has not yet been applied for handheld AR pointing and is particularly promising when focusing on interacting with POIs:

- Several AR applications offer two distinct modes: editing (to link digital information with physical objects) and exploring (to locate the POIs) [1, 6]. While the editing mode requires in-situ positioning of POIs and therefore the need for pointing at any position on screen, the exploring mode, as in the two uses cases above, involves selecting POIs. Thus prior knowledge of the targets enables us to explore target-expansion techniques to facilitate selection of POIs.
- To enhance pointing tasks, POI expansion techniques allocate larger selection areas to POIs. These techniques require empty motor space between POIs, which is often the case since a POI is linked to a physical object or location. The motor space can be altered without any functionality loss as illustrated in Figure 1.

For describing the design of the POI expansion techniques, we present the two basic elements of a target expansion technique: the expansion algorithm and the visual aid on the effective expanded POIs.

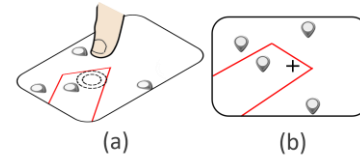


Figure 2: Pointing anywhere inside the POI Voronoi cell. (a) Touch pointing technique: sensing the finger proximity above the display surface. (b) Screen-centered crosshair technique. The Voronoi cells are drawn for explanation purpose (see Section 3.2 on visual feedback).

3.1 Dynamic Voronoi For Handheld AR

The Voronoi tessellation [8] maximizes the selection area of each POI. It divides the screen space into convex polygons, called Voronoi cells (Figure 1). Each cell contains one POI, such as all the points inside that cell are closer to the POI than to any other POIs. The resulting Voronoi cells define the extended size of the POIs. Different expansion algorithms than the one based on a standard Voronoi tessellation can be applied as in Starburst [5] for the case of clusters of targets. Expanding the POI improves pointing performances since users can point anywhere inside a Voronoi cell instead of pointing exactly at the POI.

- Applied to direct touch pointing technique (Figure 2a), sensing the finger proximity above the display surface is required [26] in order to provide feedback on the pointed POI (see Section 3.2) before the moment of touch that will provoke the selection of the POI. This extends the work of Yang et al. [26] to the case of multiple target expansion.
- Extending the screen-centered crosshair technique (Figure 2b), the user can select a POI by bringing the screen-centered cursor anywhere inside the POI Voronoi cell. In the rest of the paper we focus on this screen-centered crosshair technique that can be implemented on standard mobile devices as opposed to POI expansion for direct touch pointing technique.

In handheld AR, users work in a 3D egocentric view. Computing the Voronoi diagram involves considering all the 3D points placed in the AR environment. The technique consists of projecting the POIs onto the screen's plane of the handheld device. However, by projecting all the points, some Voronoi cells may be empty: these cells correspond to POIs not visible on screen that are located behind the camera, and whose projection are on screen. Coupled with an off-screen target localization technique like Halo3D [17], computing the Voronoi diagram for all the POIs in the environment could help users locate and select off-screen POIs. Focusing on the selection of POIs on screen as a first step, we calculate the Voronoi diagram only for POIs in front of users that are visible on screen inside the camera's field of view. At any time, each Voronoi cell corresponds to a POI displayed on screen. Processing the algorithm at the camera's frame rate leads to a dynamic Voronoi tessellation: the cells automatically adapt to moving POIs on screen. One drawback is that the Voronoi tessellation can abruptly change on the border of the screen when a POI enters

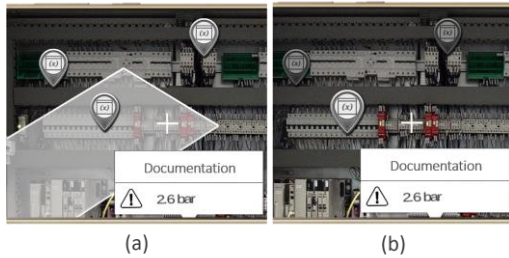


Figure 3: (a) CELL PAINTING (b) TARGET.

(resp. leaves) the screen: a new cell is created (resp. deleted) as a POI moves into (resp. out) the screen space. With the crosshair technique, the cursor is at the center of the screen and is less prone to changing cells on the edges of the screen.

3.2 Visual Feedback

Visual feedback is key since users rely on the visual aid to take full advantage of the POI expanded areas during the pointing tasks. For handheld AR, minimizing the visual intrusion is crucial since the physical environment is also displayed on screen. For instance, considering the above industrial use case, machine operators need to see the physical machine to understand the information provided by POIs. While there is empty space between POIs, the background is not empty and has meaning for users.

Guillon et. al [10] define a 3-axes design space to characterize visual feedbacks of target expansion techniques and have evaluated the impact of several visual feedbacks on performance. For 2D space they found that a static highlight of the closest target is efficient, a result recently confirmed for 3D interaction by Baloup et al. [4].

Based on these results, there are two design choices left: (1) CELL PAINTING (Figure 3a) a technique that displays the entire Voronoi cell in which the cursor is currently located. (2) TARGET (Figure 3b) a technique that highlights the closest target to the cursor. We implemented the two visual feedbacks CELL PAINTING and TARGET for the case of the screen-centered crosshair technique as shown in Figure 3. We performed a pilot study to determine which visual feedback is preferred. We conducted a pilot study with three professionals of industrial maintenance. Our goal is to test the two design options with users for which the background (i.e. the physical environment) has a meaning. Moreover, the screen-centered crosshair technique is currently used by professionals of industrial maintenance. The three participants were asked to select eight POIs located along a machine. We used a model of a production line which size was 3m x 0.75m. The task was an exploration one and no order was imposed on the selection of the eight POIs. Before starting we explained the target expansion mechanism. The three professionals preferred TARGET because of its minimal visual intrusion on screen. Moreover, they reported that TARGET enables them to better perceive the physical machine, which is the focus on the maintenance task.

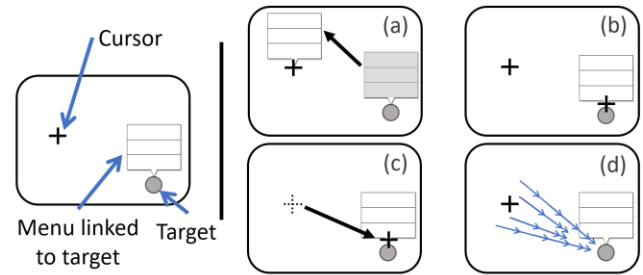


Figure 4: Target expansion in interaction context: Four Design solutions (a) Menu jumping (b) Multiple cursors (c) Cursor jumping (d) “Force Fields”: force enhancement approach.

According to the application domain and the task at hand, such choice could be different. More studies that must be conducted in pseudo ecological settings are needed to establish design guidelines on the visual aid that presents the resulting target expansion to the users in handheld AR. TARGET was the solution used in the experiment below. It is important to mention that with TARGET the Voronoi cells are not explicitly displayed (implicit visual feedback in Guillon et al.’s design space [9]). Thus there is no need to compute the complete Voronoi tessellation and as also explained in [5] a simpler algorithm consists in computing the closest POI to the screen-centered crosshair.

4 POI Expansion In Context: Design

As discussed earlier, numerous studies on target expansion techniques focus solely on optimizing pointing. Putting the POI expansion in interaction context leads us to consider the following interaction step, i.e. the selection of an item of the POI menu. In this section we first explore the design possibilities before presenting the implemented techniques for the compound task of selecting a POI and then selecting an item in the POI menu.

4.1 Design Choice

Users can select a POI by bringing the screen-centered cursor anywhere inside the POI Voronoi cell. It is then possible to select/open a POI without having the cursor exactly on the POI. While greatly facilitating interaction with POI, this raises the issue of how to interact with the POI menu when the POI is selected. The menu is displayed at the position of the POI since it is tightly related to a physical object or location. After selecting a POI with the POI expansion technique, the cursor is no longer positioned on the POI menu as depicted on the left of Figure 4.

Based on the above review of the literature we identify four design solutions depicted in Figure 4. The first immediate solutions are either to make the menu jump to the cursor (Figure 4a) or to make the cursor jump to the menu (Figure 4c). Two other solutions that do not require to change the visual layout nor to fully loose the control of the cursor consists (1) in multiple cursors, by adding a POI menu-specific cursor (Figure 4b) and (2)

in partially overruling the user's control of the cursor with "force fields" that attracts the cursor towards the menu (Figure 4d).

For handheld AR, moving the menu (Figure 4a) is not suitable because it would break the real-digital world relationship. Menus only display relevant information about their location in the real world. For instance in industrial environments, a menu showing the temperature of an engine inside a machine only makes sense at the spot the temperature is measured. We also did not explore the design choice of multiple cursors (Figure 4b) because it mainly applies for large screens. Even for a more classical screen, the satellite cursor technique (one of the most recent techniques with multiple cursors [27]), it is suggested to apply the satellite cursor technique outside the view region. For handheld AR, the screen is limited by the form factor of the mobile device. With limited size screen of mobile devices, even if users' attention is focused on the intended target (i.e. the menu item), users will be visually distracted by the other cursor.

Finally overruling the users' control of the cursor (Figure 4d) implies an indirect pointing device. The screen-centered crosshair technique implements an absolute physical pointing technique by relying on the device's pose for pointing. To be able to push the cursor towards the menu (cursor attracted in the menu direction), the screen-centered crosshair technique must be extended with a relative pointing mode where the cursor is no longer fixed at the center of the screen. Ahlström et al. [3] performed a comparison of the force enhancement approach (Figure 4d) and the jumping cursor (Figure 4c, called *Jumping Menu System* in [3]) for the case of cascading pull-down menus: performance was similar but the authors highlight advantages of the jumping cursor approach over the force enhancement one. In particular "the user is not required to change his/her usual way of interaction and can choose to take advantage of the cursor jumps only in particular cumbersome situations ...". That is why we chose to experimentally study the jumping cursor approach.

4.2 Implemented Jumping Cursor Techniques

We implemented two techniques that make the cursor jump to the POI menu. To do so the absolute physical pointing technique based on the screen-centered crosshair is extended so that the cursor is no longer fixed at the center of the screen. We extend the screen-centered crosshair by a deliberate action performed by the user. In addition as for cascading pull-down menus studied by Ahlström et al. [3], the cursor has a deterministic behavior: jumping to the closest menu. Our hypothesis is that a deliberate action to trigger the jumping cursor and the deterministic behavior of the cursor limit the disorientation effect. If users choose to take advantage of the cursor jumps, they perform a specific action that enters a quasi mode extending the screen-centered crosshair as shown in Figure 5.

For entering this quasi mode several design solutions are possible including long/short presses as well as pressure events on a button (time multiplexing) or a dedicated button (space multiplexing). We chose to implement a dedicated button as part of a bimanual interaction. When holding the handheld device with both hands in landscape mode with thumbs on the front

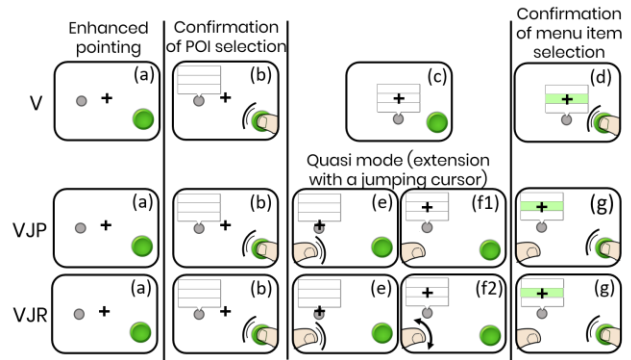


Figure 5: Techniques walkthrough. V: Voronoi POI expansion technique without cursor jumping. VJP: Voronoi Jumping-Physical. VJR: Voronoi Jumping-Relative.

and remaining fingers on the back, the two thumbs have a comfortable functional area for interaction at the bottom corners of the screen [25]. Inspired by the BiTouch design space for bimanual interaction defined by Wagner et al. [25] we designed a bi-manual interaction as follows: The left thumb is used to enter the quasi mode and manipulate the cursor while the right thumb is dedicated to confirm a selection both of a POI and of a menu item (Figure 5). Validating a selection when the cursor reaches an item would leave users no room for exploration/hesitation in a highly mobile context of handheld interaction. In this context the right button is necessary to confirm any selection. This bimanual design solution supports consistency and makes interaction easy to learn: this has been confirmed by the qualitative feedback during the experiment presented in the following section.

When the left thumb touches the screen, we have two cases according to the state of the closest POI. (1) If the POI is not selected (closed state), no menu is displayed and the cursor remains at the center of the screen. (2) If the POI is selected (open state), its menu being displayed, the cursor automatically jumps to the first item of the POI menu (Figure 5e).

The right thumb is used to press a button in order to validate a selection of an item in the menu (Figure 5g). The same button is also used to select/open and deselect/close a POI (Figure 5b). Therefore, the action of pressing the button with the right thumb has a different meaning if the cursor is located inside or outside a menu.

As a menu can include multiple items, when interacting with a menu the cursor could move closer to another POI. To avoid a non-desired jump of the cursor to another adjacent POI menu, we disable the expansion technique when the cursor is inside a menu. Finally, there are two techniques to manipulate the cursor once it jumps into the menu: (1) Absolute physical pointing: Fix the cursor in the screen space and select an item in the menu by physically moving the device (Figure 5f1), namely Voronoi Jumping-Physical technique. (2) Relative touch pointing: Fix the cursor in the physical world (Relative Pointing technique [24]), and control it with left thumb strokes on screen (Figure 5f2),

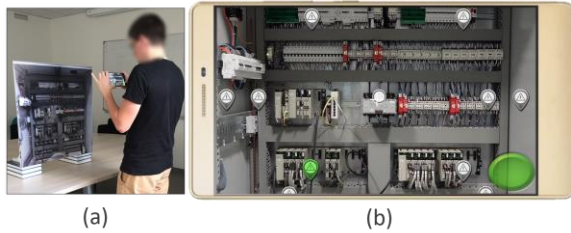


Figure 6: (a) A participant in front of the model of a production line. (b) Pointing task to select a POI.

namely Voronoi Jumping-Relative technique. When coupling these two techniques with the POI expansion technique, users are able to (1) select/open a POI without aligning the cursor with the POI, (2) place the cursor into the menu without additional movement, and (3) perform either a Physical or Relative pointing to select an item in the menu.

5 User Study

We experimentally compare the two techniques Voronoi Jumping-Physical (VJP) and Voronoi Jumping-Relative (VJR) with a Voronoi POI expansion technique without cursor jumping (V) (moving the cursor to the menu item by physical pointing figure 5c) and a baseline technique (B) the standard screen-centered crosshair technique. The goal of the experiment is twofold. The first goal is to quantify the benefits of POI expansion for pointing at a POI in handheld AR. Beyond solely optimizing the pointing, the second goal is to compare the designed techniques when performing a complete task by first selecting a POI and then interacting with the POI menu.

Inspired by the industrial use case of Section 1, the study considers a pseudo-industrial environment to compare the four techniques (VJP, VJR, V and B) when selecting POIs located on a machine and accessing information related to POIs. The look&feel of the displayed POIs, menus, and the button on the right corner of the screen as well as the baseline technique (screen-centered crosshair) is based on an existing industrial application, the Schneider's AR guidance system. The four techniques were developed in C# and rendered on the handheld device's screen using Unity3D engine.

5.1 Apparatus and Participants

Participants carried out the experiment on a Lenovo phab2PRO phone in landscape mode. The device is provided with full spatial awareness thanks to Google's Tango technology, a combination of depth perception and motion tracking sensors. The phone also features a 6.4 inches screen with a resolution of 2560 x 1440 pixels which was fully used in this experiment. To replicate an industrial environment, we took a photo of one side of a 9m x 3m production machine. We print it on 3mm depth forex surface to build a 0.75m x 0.75m panel (Figure 6a). The study involved 12 unpaid volunteers from a computer science university lab (5 females, 7 males; ranging in age from 23 to 41 years, mean=29.6, sd=5.7).

5.2 Experimental Task and Environment

The experimental task was designed to replicate our industrial use case: participants were asked to (1) select/open POIs located on a machine, and (2) to select an item of the opened menu for the given POIs. A 2D static target selection task, based on standard ISO 9241-9 was used for this experiment.

Participants were presented 8 POIs placed on a circle (Figure 6b). To constraint the expanded areas of those POIs, 8 other POIs were placed on a larger concentric circle. Therefore, all the POIs had the same expanded size. To take advantage of the expanded areas, participants were asked to perform the experiment at 40 cm from the machine. This allows participants to have large enough expanded areas to select/open a POI without having the cursor placed near the POI. The rectangular shape of the screen's device led to 2 issues: (1) the top and bottom POIs of the circle did not fit on screen, and (2) when a POI of the circle was selected, the opposite POI on the circle was always off-screen. To solve this issue related to the limited size and rectangular shape of the screen, we (1) eliminated the top and bottom POIs from the sequence of POIs to point at, and (2) ask participants to go back to a starting point between each pointing task. The starting point was placed at the center of the circle. This way, the next POI to select on the circle was always inside the camera's field of view. Finally, the starting point was a special target that was not included when computing the POI closest to the cursor. This option also prevents participants from selecting the starting point without aligning the cursor with it and also saves screen space for larger POIs' expanded areas.

Both the selection of a POI and of a menu item are validated by pressing a "Validate" button placed on the bottom right of the screen. Menus contained 5 items and the item to be selected was the second one from the top of the menu. Future work could complement this one by varying the menu items.

5.3 Hypotheses and Measured Data

We formulated the following hypotheses:

- **H1.** The V, VJP and VJR techniques perform faster than the baseline technique (B) when selecting/opening POIs.
- **H2.** VJP is perceived more usable than VJR: physically moving the device towards a POI is the default technique for the first part of the task. VJR thus implies to change the technique for the second part of task - selecting a menu item. This modification may disturb users' workflow to complete the task.

For each technique, we recorded the time spent to (1) select/open a POI (T1) (to verify H1), and (2) select the menu item (T2). The overall time equals $T1 + T2$. We also recorded the number of errors when selecting an item on screen: an error was computed when participants either selected/opened an incorrect POI or selected an incorrect item in the menu. We also consider subjective metrics to verify H2. After completing all pointing tasks for one technique, participants were asked to fill out a Raw-TLX [12] 5-point Likert scale questionnaire. At the end of

the entire session, participants were asked to rate, for each technique, how mentally and physically demanding the task was and how easy the techniques were to learn. Questions were to be answered on a 5-level Likert scale [14]. Finally, the participants were also asked to rank the 4 techniques.

5.4 Experimental Design and Procedure

The task involved a 4 (pointing techniques: B, V, VJP, VJR) within-subjects design. We divided the participants into 4 groups of 3 individuals. Each group started the experiment with a different technique. Each participant conducted 120 pointing tasks (4 techniques x 5 iterations x 6 POIs).

Participants were first given a short explanation about the experimental tasks and all 4 techniques. Due to the number of techniques, participants were invited, for each technique to perform the trial session immediately after the training session [7]. A training session consisted in selecting 15 POIs and items (selecting/opening a POI + selecting a menu item). For each technique, the experiment began only when the participants felt comfortable with the task. Otherwise, they were required to perform another training block of 15 pointing tasks. These lengthy training sessions before using each technique were designed to let participants become highly familiar with the techniques. It is very important because the techniques are of different complexity but also closely related. The lengthy training sessions were designed to mitigate the skill transfer between the techniques, enabling us to use a within-subject experimental design without the need to fully counterbalance the techniques across participants.

Participants performed the task at a specific spot in front of the machine marked with a white cross on the ground. The participants were instructed to be as fast and as accurate as possible. After completing all pointing tasks with one technique, participants filled out a Raw-TLX questionnaire and returned to the marked location to start with the following technique. We concluded the study with a semi-structured interview to collect participants' feedback. The overall session lasted 30 minutes.

5.5 Results

5.5.1 Selection Times and Error Rates. We report the results by considering the effects of each technique on the error rate and the time needed to (1) select/open a POI and (2) select a menu item. A Shapiro-Wilk test indicated that the overall task completion time and POI opening time data follow a normal distribution. A one-way ANOVA did not find a strong effect of the techniques on the overall task completion time ($F_{(3,44)} = 2.471$, $p=0.07$). However, it showed a statistically significant difference of the techniques on the time needed to select a POI ($F_{(3,44)} = 20.47$, $p<0.001$). A post hoc test conducted using pairwise t-tests and Bonferroni corrections revealed that statistically, participants were significantly longer to open POIs with the baseline technique than with V ($p<0.001$), VJP ($p<0.001$) and VJR ($p<0.001$). The menu item selection time data did not follow a normal distribution. The non-parametric Kruskal-Wallis test did not reveal a strong effect of the techniques on the time needed to

	POI selection time (s)	Menu item selection time (s)	Overall Task completion time (s)
B	1.65; 1.6 (0.22)	1.26; 1.26 (0.17)	2.91; 2.93 (0.36)
V	1.11; 1.12 (0.15)	1.31; 1.2 (0.23)	2.42; 2.34 (0.36)
VJP	1.17; 1.13 (0.2)	1.45; 1.3 (0.4)	2.61; 2.43 (0.59)
VJR	1.14; 1.12 (0.2)	1.38; 1.35 (0.33)	2.52; 2.46 (0.51)

Table 1: Mean, median (and standard deviation) of POI selection times (s), menu item selection times (s) and overall task completion times (s) for all 4 techniques.

	Mental tiredness	Physical fatigue
B	1.75; 1.5 (0.97)	3.08; 3 (1.08)
V	1.67; 1 (0.98)	2.17; 2 (0.94)
VJP	2.25; 2 (0.87)	1.83; 2 (0.83)
VJR	1.92; 1.5 (1.24)	1.67; 1 (0.98)

	Easy to learn	Overall note
B	3.67; 5 (1.97)	2.42; 2 (1.24)
V	3.42; 4 (1.83)	3.58; 4 (0.79)
VJP	3.17; 3 (1.27)	3.67; 4 (0.89)
VJR	3.25; 3 (0.97)	4.5; 5 (0.8)

Table 2: Mean, median (and standard deviation) of the ratings (out of 5) assigned to the techniques for the mental tiredness, the physical fatigue, how easy the techniques to learn were and the final ranking.

select a menu item ($\chi^2_{(3)} = 0.93$, $p=0.817$).

The error rate data did not follow a normal distribution. The Kruskal-Wallis test did not reveal a strong effect of the techniques on the error rate, either for selecting a POI ($\chi^2_{(3)} = 3.93$, $p=0.26$) or selecting a menu item ($\chi^2_{(3)} = 6.34$, $p=0.09$).

5.5.2 Subjective Preferences: Final Survey. A non-parametric Friedman test found a strong effect of the techniques on the physical fatigue ($\chi^2_{(3)} = 16.9$, $p<0.001$), and on the overall note ($\chi^2_{(3)} = 16.1$, $p=0.001$). Pairwise comparisons with Bonferroni corrections revealed that (1) the baseline technique was rated significantly more physically tiring than both VJP ($p<0.01$) and VJR ($p<0.05$), and (2) the baseline was rated worse than V ($p<0.01$) and VJR ($p<0.01$). VJR on the other hand was considered as the preferred technique. No statistically significant differences in all other measurements were found: mental tiredness ($\chi^2_{(3)} = 3.79$, $p=0.28$), easy to learn ($\chi^2_{(3)} = 2.32$, $p=0.50$).

5.6 Discussion

5.6.1 POI Selection and Menu Item Selection. The baseline technique (B) performed worst to open POIs (**Table 1**). The time needed to select/open POIs was 31% longer with the baseline technique than any other techniques. This supports our hypothesis H1. Even though we did not find statistically significant differences in the error rates, all participants stated that the baseline technique required extra precision and attention to correctly select a POI on screen. Our results confirm and quantify previous results i.e. target expansion techniques are effective to facilitate pointing tasks. Results also show that target

expansion techniques can effectively be applied to dynamic augmented reality targets.

Although VJP and VJR reduce the distance to reach the menu item, we did not find a statically significant difference between all the techniques in the time needed to select the item (**Table 1**).

5.6.2 Distance to Reach the Menu Item. The second menu item to be selected was always located in the upper section of a menu. As menus always open above POIs, two cases arise when using target expansion techniques (V, VJP, VJR): (1) the opened POI is located in the lower part of the screen, and the distance between the cursor and the menu item is reduced, as the cursor is already close to the upper section of the menu. (2) The opened POI is in the upper part of the screen, and the path to reach the menu item is longer, as the cursor is located below the menu.

Compared to VJP and VJR, the V technique is the only technique that allows participants to point directly at the menu item once the POI has been selected. With VJP and VJR, the path is indirect as the cursor first jumps at the bottom of the menu, and then needs to be moved to the menu item. Therefore, the V technique is favored (rep. disadvantaged) with the POIs located in the lower (resp. upper) part of the screen. When examining only the POIs in the lower part of the screen, the non-parametric Kruskal-Wallis test did reveal a strong effect of the techniques on the time to select a menu item ($\chi^2_{(3)} = 8.42, p=0.04$). Pairwise comparisons with Bonferroni corrections revealed that participants took longer to select a menu item with VJP than with V. For POIs located in the upper section of the screen, we did not find a strong effect of the techniques on the menu selection time: although VJP and VJR techniques reduce the distance to reach the menu item through a cursor jump, the interaction and context switch induce two reaction times. A first reaction time is needed when a POI has been selected/opened, in order to switch to a two-handed interaction and make the cursor jump. After the foreseen cursor jump, another reaction time is needed to (1) spot the new position of the cursor and (2) understand the new context in which the cursor is located.

5.6.3 Interaction modality switch. To perform the compound task of POI selection + menu item selection, VJP and VJR techniques both change (1) the number of hands needed to perform the task, as participants are required to additionally use their left hand to make the cursor jump to the menu, and (2) the visual context surrounding the cursor when it jumps to another place on screen. In addition to these changes, the VJR technique also involves to switch interaction modalities from physically moving the device in order to select a POI, to using thumb strokes on screen in order to select a menu item. Even with all those interaction and context switches, VJR technique was ranked best (**Table 2**), contradicting H2. 10/12 participants reported that VJR limited the physical fatigue. Although we did not find statically significant differences between VJR and VJP for the physical fatigue, participants stated that with VJR, they were not required to move the device to select a menu item anymore. Therefore, this is consistent with the first part of the pointing task: the target expansion technique limits the physical movements

necessary to select a POI and the Relative Pointing interaction limits the physical movements by allowing the cursor to be moved with thumb strokes on screen. 6/12 participants stated that they found it more reasonable to move their left thumb with VJR than perform physical movements with VJP: while VJP only uses the left thumb to make the cursor jump to a menu, VJR takes the advantage of the left thumb on screen to move the cursor. Therefore, participants found coherent to let the left thumb on screen and use it to select the menu item.

5.6.3 Augmented Reality layers as mental model. Whether VJP or VJR were used, 4 participants reported that the left thumb added a “layer” to help them with the pointing task. During the first part of the task, users are required to select a POI. They navigate through the layer containing all the POIs. Once opened, the left thumb makes the cursor jump to the corresponding menu. This is perceived as a new layer containing the menu only. This mental model helps them ignore the rest of the digital objects on screen. Once the menu item is validated, the cursor returns to the center of the screen, back to the layer containing all the POIs (**Figure 1a**).

5.6.4 Overall task. We did not find a strong effect of VJP and VJR on the overall time to complete the compound task (**Table 1**). The gain made by reducing the distance between the cursor and the menu are balanced with the time users needed to mentally process the interaction and cursor context switch. However, participants reported significantly less physical fatigue with VJP and VJR techniques than with B (**Table 2**). Finally, VJR has been ranked with the best overall note (**Table 2**): participants (1) found more easy to continue using the left thumb to interact with the cursor and (2) were not required to physically move the device anymore. Although no significant difference was found on how easy it was to learn the techniques, all the participants reported that the learning phase with VJP and VJR needed more attention than with V and B. Moreover, 9/12 participants needed 2 training blocks to correctly handle the two-handed interaction. Thus users performing many pointing tasks, like machine operators, may find it beneficial to put an extra effort in learning VJR, and then be efficient while minimizing the fatigue in the long run.

6 Conclusion

While target expansion techniques are not new, applying them to the case of selection of POIs in handheld AR is new. The known limitations of target expansion techniques disappear in an AR context, making them promising and directly answering a concern raised by operators (industrial use case). Our study also considers a complete task of pointing to a POI and then interacting with its menu: expanding target in interaction context. Results show that: (1) Voronoi target expansion techniques are efficient to select POIs and fit well to handheld AR applications. (2) When considering the compound task of POI selection + menu item selection, combining physical pointing with relative pointing is preferred as it minimizes the physical fatigue.

REFERENCES

- [1] Abate, A. F., Nappi, M., Narducci, F., & Ricciardi, S. (2019). Mixed reality system for industrial environment: an evaluation study. *CAAI Transactions on Intelligence Technology*, 2(4), 182-193.
- [2] David Ahlström. 2005. Modeling and improving selection in cascading pull-down menus using Fitts' law, the steering law and force fields. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*. ACM, New York, NY, USA, 61–70.
- [3] David Ahlström, Rainer Alexandrowicz, and Martin Hitz. 2006. Improving menu interaction: a comparison of standard, force enhanced and jumping menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. ACM, New York, NY, USA, 1067–1076.
- [4] Marc Baloup, Thomas Pietrzak, and Géry Casiez. 2019. RayCursor: A 3D Pointing Facilitation Technique based on Raycasting. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, Paper 101, 12 pages.
- [5] Patrick Baudisch, Alexander Zotov, Edward Cutrell, and Ken Hinckley. 2008. Starburst: a target expansion algorithm for non-uniform target distributions. In *Proceedings of the working conference on Advanced visual interfaces (AVI '08)*. ACM, New York, NY, USA, 129–137.
- [6] Blanco-Novoa, O., Fernández-Caramés, T. M., Fraga-Lamas, P., & Vilar-Montesinos, M. A. (2018). A practical evaluation of commercial industrial augmented reality systems in an industry 4.0 shipyard. *IEEE Access*, 6, 8201-8218.
- [7] Géry Casiez, Daniel Vogel, Qing Pan, and Christophe Chaillou. 2007. RubberEdge: reducing clutching by combining position and rate control with elastic feedback. In *Proceedings of the 20th annual ACM symposium on User interface software and technology (UIST '07)*, 129-138.
- [8] S. Fortune 1987. A sweepline algorithm for Voronoi diagrams. *Algorithmica* 2, 1 (1987), 153-174.
- [9] Maxime Guillon, François Leitner, and Laurence Nigay. 2016. Target Expansion Lens: It is Not the More Visual Feedback the Better!. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '16)*, Paolo Buono, Rosa Lanzilotti, and Maristella Matera (Eds.). ACM, New York, NY, USA, 52-59.
- [10] Maxime Guillon, François Leitner, and Laurence Nigay. 2015. Investigating Visual Feedforward for Target Expansion Techniques. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 2777–2786.
- [11] Tovi Grossman and Ravin Balakrishnan. 2005. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*. ACM, New York, NY, USA, 281-290.
- [12] Hart, Sandra G., and Lowell E. Staveland. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Advances in psychology*. Vol. 52. North-Holland, 1988. 139-183.
- [13] Masatomo Kobayashi and Takeo Igarashi. 2003. Considering the direction of cursor movement for efficient traversal of cascading menus. In *Proceedings of the 16th annual ACM symposium on User interface software and technology (UIST '03)*. ACM, New York, NY, USA, 91–94.
- [14] Rensis Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 22 140:55–, 1932.
- [15] Michael J. McGuffin and Ravin Balakrishnan. 2005. Fitts' law and expanding targets: Experimental studies and designs for user interfaces. *ACM Trans. Comput.-Hum. Interact.* 12, 4 (December 2005), 388–422.
- [16] Martez E. Mott and Jacob O. Wobbrock. 2014. Beating the bubble: using kinematic triggering in the bubble lens for acquiring small, dense targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*.
- [17] Patrick Perea, Denis Morand, and Laurence Nigay. 2019. Spotlight on Off-Screen Points of Interest in Handheld Augmented Reality: Halo-based techniques. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces (ISS '19)*. ACM, New York, NY, USA, 43–54.
- [18] R. L. Potter, L. J. Weldon, and B. Shneiderman. 1988. Improving the accuracy of touch screens: an experimental evaluation of three strategies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '88)*, J. J. O'Hare (Ed.). ACM, New York, NY, USA, 27-32.
- [19] Michael Rohs and Antti Oulasvirta. 2008. Target acquisition with camera phones when used as magic lenses. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 1409-1418.
- [20] Anne Roudaut, Stéphane Huot, and Eric Lecolinet. 2008. TapTap and MagStick: improving one-handed target acquisition on small touch-screens. In *Proceedings of the working conference on Advanced visual interfaces (AVI '08)*. ACM, New York, NY, USA, 146-153.
- [21] Xiaojun Su, Oscar Kin-Chung Au, and Rynson W.H. Lau. 2014. The implicit fan cursor: a velocity dependent area cursor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 753-762.
- [22] Theophanis Tsandilas and m. c. schraefel. 2007. Bubbling menus: a selective mechanism for accessing hierarchical drop-down menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, New York, NY, USA, 1195–1204.
- [23] Thomas Vincent, Laurence Nigay, and Takeshi Kurata. 2013. Handheld Augmented Reality: Effect of registration jitter on cursor-based pointing techniques. In *Proceedings of the 25th Conference on Interaction Homme-Machine (IHM '13)*. ACM, New York, NY, USA, Pages 1, 6 pages.
- [24] Thomas Vincent, Laurence Nigay, and Takeshi Kurata. Precise pointing techniques for handheld Augmented Reality. In *Proc INTERACT 2013, LNCS 8117, IFIP-Springer (2013)*. pp. 122-139.
- [25] Julie Wagner, Stéphane Huot, and Wendy Mackay. 2012. BiTouch and BiPad: designing bimanual interaction for hand-held tablets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 2317-2326.
- [26] Xing-Dong Yang, Tovi Grossman, Pourang Irani, and George Fitzmaurice. 2011. TouchCuts and TouchZoom: enhanced target selection for touch displays using finger proximity sensing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 2585–2594.
- [27] Chun Yu, Yuanchun Shi, Ravin Balakrishnan, Xiangliang Meng, Yue Suo, Mingming Fan, and Yongqiang Qin. 2010. The satellite cursor: achieving MAGIC pointing without gaze tracking using multiple cursors. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology (UIST '10)*. ACM, New York, NY, USA, 163–172.