Received September 26, 2019, accepted November 5, 2019, date of publication November 18, 2019,
date of current version December 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2954101

CameraKeyboard: A Novel Interaction Technique for Text Entry Through Smartphone Cameras

ALESSIO BELLINO[1] AND VALERIA HERSKOVIC[2]
[1]Escuela de Informática y Telecomunicaciones, Facultad de Ingeniería y Ciencias, Universidad Diego Portales, Santiago 8370190, Chile
[2]Departamento de Ciencia de la Computación, Pontificia Universidad Católica de Chile, Santiago 7820436, Chile

Corresponding author: Alessio Bellino (alessio.bellino@mail.udp.cl)

This work was supported in part by the Comisión Nacional de Investigación Científica y Tecnológica/Fondo Nacional de Desarrollo Científico y Tecnológico (CONICYT/FONDECYT), Chile, under Grant 1181162, and in part by the Engineering Postdoc UC 2018.

ABSTRACT We present CameraKeyboard, a text entry technique that uses smartphone cameras to extract and digitalise text from physical sources such as business cards and identification documents. After taking a picture of the text of interest, the smartphone recognises the text through OCR technologies (specifically, Google Cloud Vision API) and organises it in the same way in which it is displayed in the original source. Next, users can select the required text - which is split into lines, and further split into words - by tapping on it. CameraKeyboard is designed to easily complement the standard keyboard when users need to digitalise text. In fact, the camera is integrated directly in the standard mobile QWERTY keyboard and can be accessed by tapping a button. CameraKeyboard is general-purpose like any other smartphone keyboard: it works with any application (e.g., Gmail, Whatsapp) and is able to extract text from any kind of source. We evaluated CameraKeyboard through a user study with 18 participants who carried out four different text entry tasks, and compared it to a standard smartphone keyboard (Google Keyboard) and a standard (physical) desktop keyboard. Results show that CameraKeyboard is the faster one in most cases.

INDEX TERMS Text-entry, smartphone interaction, camera interaction, OCR, interface design.

I. INTRODUCTION

Nowadays, smartphones are used for myriad activities, e.g. business, leisure, socialization, and informative purposes. Text entry is a common activity carried out on smartphones to leverage these capabilities. Several text entry methods have been proposed, from the basic on-screen touch keyboard (as are the standard iOS and Android keyboards) to keyboard optimizations, e.g. Swype, Swiftkey [1], as well as handwriting recognition, voice to text, and virtual keyboards that replace the traditional QWERTY layout [2].

When users need to transcribe a text on their smartphone from physical sources such as a contact card, document id, or lottery code, they will use one of the text entry methods available to them. Optical character recognition (OCR) technologies can help users simplify the transcription of text, and have been available on mobile for several years [3], with several currently available commercial applications [4]. Some applications are used to recognise structured documents, e.g. business cards [5], automatically recognising name, surname, phone number, and email address. However, OCR technologies are mostly used as stand-alone applications that have to be used separately from the application in which the text must be entered.

This paper presents *CameraKeyboard*, an on-screen keyboard that seamlessly uses the smartphone camera to extract text from physical sources by using OCR. Therefore, CameraKeyboard does not work as a standalone application, nor is intended to replace the standard keyboard. Rather, it is designed to easily complement the standard QWERTY keyboard when users need to digitalise text from physical sources. CameraKeyboard is general-purpose like any other smartphone keyboard: it works with any application and is able to extract text from any kind of physical source.

The paper is organized as follows. After discussing the related work, we present the design and implementation of CameraKeyboard. Next, we present the user evaluation, in which we compare our text entry method to a standard on-screen QWERTY keyboard and a physical desktop keyboard. Finally, we discuss the results and some design implications.

The associate editor coordinating the review of this manuscript and approving it for publication was Alessio Vecchio.

II. RELATED WORK

A. TEXT ENTRY METHODS FOR SMARTPHONES

The most usual text entry method for smartphones is a touchscreen QWERTY keyboard. This type of virtual keyboard only appears when it is needed, allowing the user a larger screen area, as well as permitting users to configure it, e.g. according to their preferred language [6]. However, several researchers have noted that due to their relatively small screen size, the QWERTY keyboard is not necessarily the most efficient one for smartphones. This, combined with the occlusion and fat finger problems inherent to direct-touch input [7], make it difficult for some users to efficiently and correctly type on touchscreen keyboards. The problems caused by the small sizes of keyboards in smartphones are even more pronounced as screens get smaller, e.g. smartwatches are more suitable for viewing, rather than inputting, content [8].

Therefore, many improvements to touchscreen QWERTY keyboards have been proposed, several of which are widespread today – e.g. word completion (suggestions) and predictive text to anticipate user input [6], [9]. Shape writing, e.g. Swype and Swiftkey, or similar keyboards in which the user draws a connecting line between letters in a keyboard [10], are also widely in use. Some researchers have proposed a different key layout, finding that introducing an alternative keyboard immediately is better than introducing it gradually [11].

Differently-shaped keyboards have also been introduced by reducing the number of keys, i.e. letters are grouped in a single button or area and word disambiguation techniques are employed (e.g., LetterWise, [12] 1Line keyboard [13]). Some of these techniques, besides reducing the number of keys pressed by grouping multiple letters, are designed to allow sight-free text entry. Two examples are ThumbStroke [14] and Escape-Keyboard [15]: the former is a virtual keyboard with a single round key in which each letter is located in one of eight areas, and the latter shows four areas where users can perform touches-and-flicks in eight different directions to select a letter.

Researchers have also proposed low-occlusion [16] or even invisible keyboards [17], leveraging the fact that most users have by now memorized the keyboard layout, finding that users were able to reach a practical level of performance after a few sessions [17].

Gestures, voice, and handwriting recognition have also been used as input modalities for smartphones. For example, the mobile devices' rear camera has been used to capture finger gestures [18], and drawn gestures have been used to input programming codes on mobile devices, with better results in speed, comfort and user satisfaction over a traditional keyboard [19]. Although voice is the preferred input method for smartwatches [8], it suffers from several issues: mistakes have to be corrected manually, and although improvements have been made, the algorithms have to take into consideration myriad accents and dialects [6].

Some studies have compared mobile input methods in different situations. A comparison of input methods for older and younger users found that voice was the fastest method overall, and that voice and physical QWERTY keyboards had a lower word error rate and were the most usable methods [20]. A comparison of a speech and touchscreen keyboard in English and Mandarin Chinese found that in both languages, speech was almost three times as fast as the keyboard and produced fewer errors during entry but slightly more errors in the final text [21].

B. MOBILE OCR

Several OCR applications are available for Android and iOS smartphones, e.g. scanner-type applications or narrowly specialized applications, such as those that only allow business card processing [22], receipt management [23], or translation of visual text [24]. A considerable amount of research has focused on mobile OCR for blind users, e.g. to allow them to read posters and flyers [25], since OCR applications designed for sighted users have to be adapted to help blind users achieve OCR-readable images [26]. Most previous research has focused on improving the obtained images, as in the case of blind users [26], or improving the OCR algorithms' accuracy and efficiency.

In most of the existing smartphone OCR apps, the text is typically extracted as a unique text file. Then users then need to (1) select, (2) copy and (3) paste the fragments of text of their interest in other applications. To do so, users need to go forward and back between the OCR app and the target app/website where the different fragments of text are required. Additionally, it is important to note that selecting a text fragment (from a longer text) on a smartphone is not straightforward as on desktop computers.

A commercial application for smartphones called Scanner Keyboard [1] integrates OCR in a traditional keyboard, allowing users to take a picture and choose an area of it to select the relevant text to be copied. This is the closest approach to CameraKeyboard, since it allows users to extract the text in the same app/website where it is needed, without requiring users to swap between apps.

However, Scanner Keyboard is designed to be used to fill out a single field. In fact, when it is needed to fill in different fields (e.g., name, surname, and email), users are required to go back and forth between the image (to select the text of interest) and the corresponding text fields. In contrast, CameraKeyboard extracts all the text at once, and users can select the fields of interest and enter the corresponding text just like writing on a normal keyboard. Moreover, the extracted text is automatically split into fragments (lines and words) so that user can simply tap on the fragments of interest rather than select them from a longer text.

This paper proposes CameraKeyboard, an OCR smartphone keyboard that allows an alternative input mode for

[1] https://play.google.com/store/apps/details?id=com.tecit.android.barcodekbd.demo - https://youtu.be/r3uzb1uFsP8

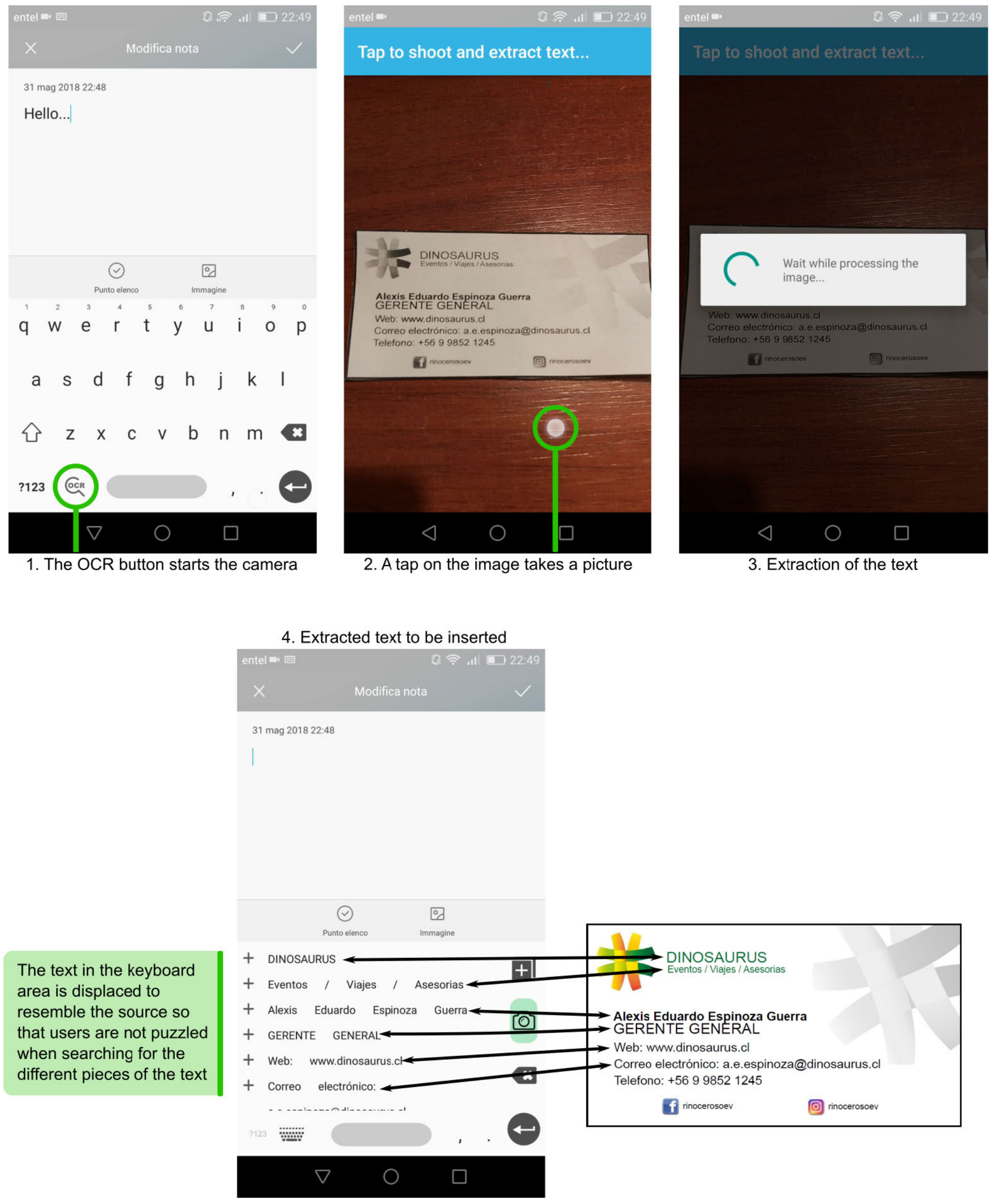

1. The OCR button starts the camera
2. A tap on the image takes a picture
3. Extraction of the text
4. Extracted text to be inserted
Modifica nota
31 mag 2018 22:48
Hello...
Tap to shoot and extract text...
Wait while processing the image...
DINOSAURUS
Eventos / Viajes / Asesorias
Alexis Eduardo Espinoza Guerra
GERENTE GENERAL
Web: www.dinosaurus.cl
Correo electrónico: a.e.espinoza@dinosaurus.cl
Telefono: +56 9 9852 1245
The text in the keyboard area is displaced to resemble the source so that users are not puzzled when searching for the different pieces of the text

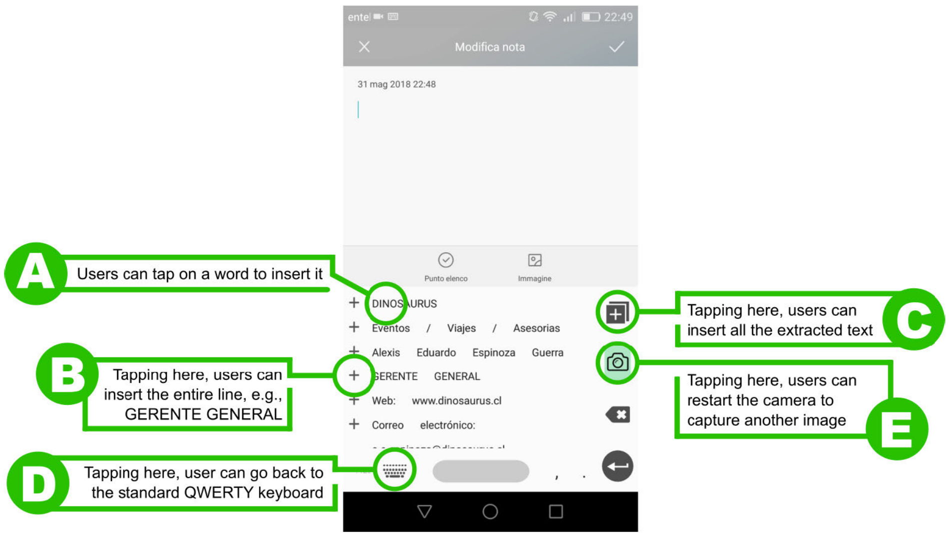
FIGURE 1. **CameraKeyboard use process. At the beginning, users can write through the QWERTY on-screen keyboard as usual. When users need to start the camera, they tap on the OCR button (1), then pointing the camera to the text of interest, user can take a picture (2). The image is sent to the Google Cloud Vision API and the text is extracted (3). Once the text is returned to the smartphone, it is displaced similarly to the source and user can insert the entire text, a single line, or a single word.**

users, especially when they need to share, store, or use textual information from the real world on their smartphones. Although CameraKeyboard integrates the use of well-known OCR technologies in a QWERTY keyboard, the design of the user interaction changes the way in which users use and consume OCR.

C. COGNITIVE FACTORS: RECALL VERSUS RECOGNITION

In psychology, the distinction between recall and recognition is well-known. Recognition deals with the ability to recognise a piece of information. In contrast, recall represents a person's ability to retrieve information details from memory [27]. Broadly speaking, recognition is much easier than recall; this aspect is well-known in user interface design [28]. For example, users generally find it easier to recognise a command in a graphical user interface rather than recall a command that must be written into a command-line interface.

Recall and recognition are involved when transcribing a text. When using a QWERTY keyboard to transcribe a text, users are required to *recall* the text of interest, and write it by



Modifica nota
31 mag 2018 22:48
Punto elenco
Immagine
DINOSAURUS
Eventos / Viajes / Asesorias
Alexis Eduardo Espinoza Guerra
GERENTE GENERAL
Web: www.dinosaurus.cl
Correo electrónico:
A
Users can tap on a word to insert it
B
Tapping here, users can insert the entire line, e.g., GERENTE GENERAL
C
Tapping here, users can insert all the extracted text
D
Tapping here, user can go back to the standard QWERTY keyboard
E
Tapping here, users can restart the camera to capture another image

FIGURE 2. CameraKeyboard options for inserting text, go back to the QWERTY keyboard and restart camera to take another picture.

using the keyboard. When using CameraKeyboard, instead, users are required to *recognise* the extracted text, and write it by selecting it. Considering the lower cognitive burden that recognition represents over recall, we posit that the use of CameraKeyboard should be less mentally demanding than the use of a QWERTY keyboard.

III. DESIGN OF CAMERAKEYBOARD

A. USER INTERFACE

CameraKeyboard initially looks like a regular QWERTY keyboard (Fig. 1-Top-Left), in which users can write as usual. However, when users need to transcribe a text from a physical source, they can select the OCR button on the bottom left of the keyboard, which starts the camera, point it at the text of interest, and capture the image (Fig. 1-Top-Center). Then, the text is extracted through OCR (Fig. 1-Top-Right) and displayed on the keyboard area resembling the displacement of the original source line by line (Fig. 1-Bottom).

Once the text is displayed, users have several options to insert the text in the application they are using. They can insert the extracted text all at once (Fig. 2-C), insert single lines (e.g., Fig. 2-B), or insert a single word (e.g., Fig. 2-A). Also, users can go back to the standard QWERTY keyboard (Fig. 2-D), or restart the camera to extract the text from another picture (Fig. 2-E).

B. TECHNICAL FEATURES

CameraKeyboard is developed in Java as any other alternative keyboard for Android. The QWERTY keyboard works as usual, but users have the option to enable the camera to take pictures and recognise text. In such a case, pictures are reduced at a resolution of 1280x960 and compressed in JPEG format. Then pictures are sent to the Google Cloud Vision API, which is used to extract the text through OCR. This web-service allows uploading an image (as input), returning a JSON (as output) with the recognised text organized line-by-line. The orientation of the image does not affect text recognition, so users can take landscape or portrait pictures according to their preferences and the specific situation.

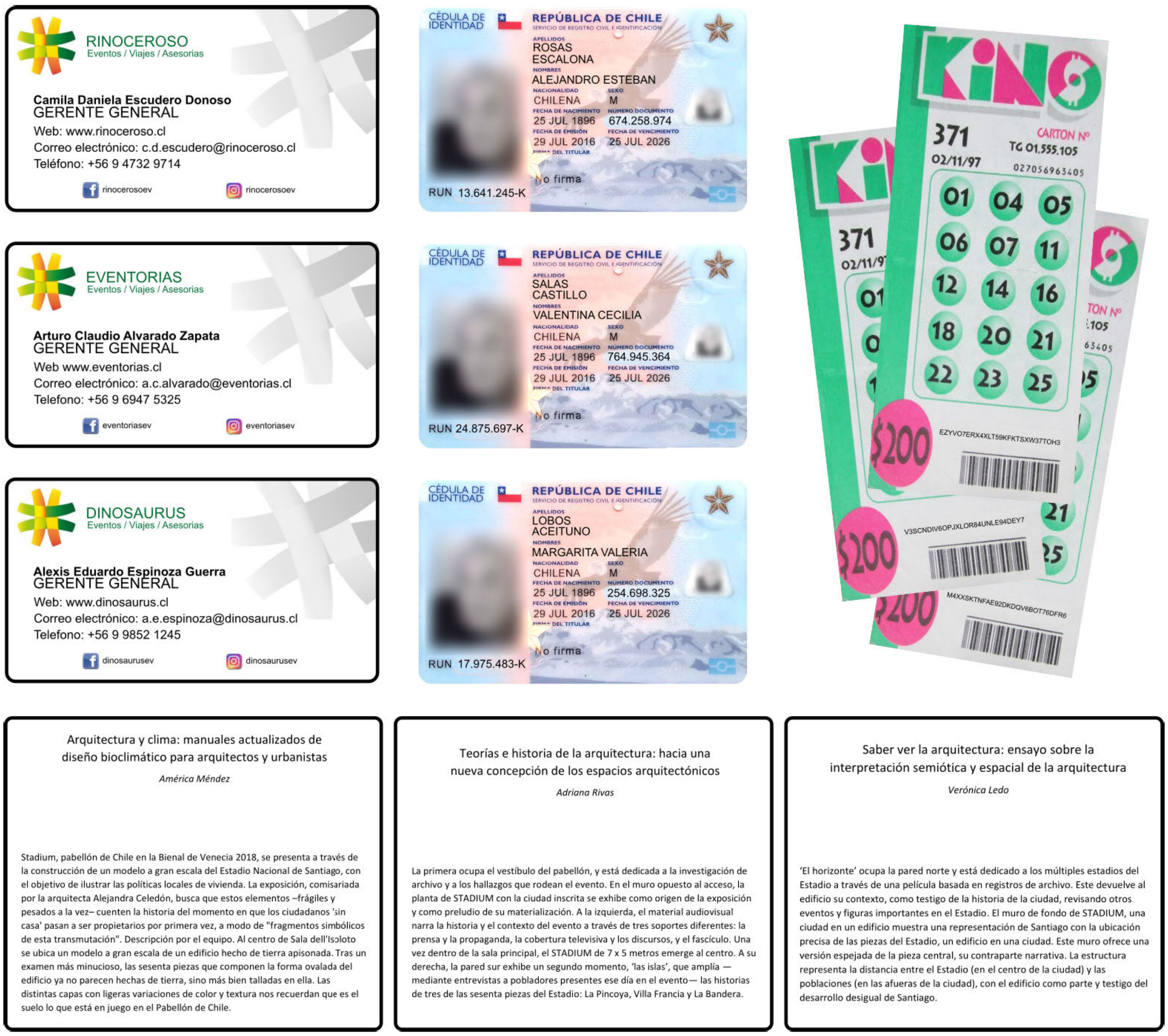
IV. STUDY SETUP

A. PARTICIPANTS AND APPARATUS

We selected eighteen participants (5F/13M) aged between 18 and 48 years (M: 26.4, SD: 9.2). None of them had previous knowledge about CameraKeyboard. The user study consisted of four tasks, carried out sequentially, with a total duration of around 30 minutes. Participants were compensated with a 7 USD gift card. The studies took place in a university office. We used a Huawei Shot X smartphone, which ran CameraKeyboard and Google Keyboard, and a laptop with a Spanish keyboard.

B. PROCEDURE

First, participants signed a consent form. Then, they were presented with an overview of CameraKeyboard. To show how CameraKeyboard works, we demonstrated how to capture data from a driving licence including examples of transcription of single words (e.g., Fig. 2-A), entire lines (e.g., Fig. 2-B), and all extracted texts (Fig. 2-C). Next, we asked participants to carry out four transcription tasks. For each task, participants had to transcribe one item using CameraKeyboard, one using Gooogle Keyboard, and one using a physical QWERTY keyboard. The first task was to transcribe the data from a business card (first name, last name, email, and phone number). The second task was the transcription of data from a Chilean ID (first name, last name, personal id, and document code). The third task was to transcribe a lottery code. The final task was the transcription of the title and author of a paper. All participants were native Spanish speakers, and all the texts to transcribe were also in Spanish. To avoid carryover effect, the order of the conditions for each participant - i.e., CameraKeyboard, Google Keyboard,



RINOCEROSO
Eventos / Viajes / Asesorias

Camila Daniela Escudero Donoso
GERENTE GENERAL
Web: www.rinoceroso.cl
Correo electrónico: c.d.escudero@rinoceroso.cl
Teléfono: +56 9 4732 9714
rinocerosoev rinocerosoev

EVENTORIAS
Eventos / Viajes / Asesorias

Arturo Claudio Alvarado Zapata
GERENTE GENERAL
Web www.eventorias.cl
Correo electrónico: a.c.alvarado@eventorias.cl
Telefono: +56 9 6947 5325
eventoriasev eventoriasev

DINOSAURUS
Eventos / Viajes / Asesorias

Alexis Eduardo Espinoza Guerra
GERENTE GENERAL
Web: www.dinosaurus.cl
Correo electrónico: a.e.espinoza@dinosaurus.cl
Telefono: +56 9 9852 1245
dinosaurusev dinosaurusev

CÉDULA DE IDENTIDAD
REPÚBLICA DE CHILE
SERVICIO DE REGISTRO CIVIL E IDENTIFICACIÓN
APELLIDOS
ROSAS
ESCALONA
NOMBRES
ALEJANDRO ESTEBAN
NACIONALIDAD SEXO
CHILENA M
FECHA DE NACIMIENTO NÚMERO DOCUMENTO
25 JUL 1896 674.258.974
FECHA DE EMISIÓN FECHA DE VENCIMIENTO
29 JUL 2016 25 JUL 2026
FIRMA DEL TITULAR
No firma
RUN 13.641.245-K

CÉDULA DE IDENTIDAD
REPÚBLICA DE CHILE
SERVICIO DE REGISTRO CIVIL E IDENTIFICACIÓN
APELLIDOS
SALAS
CASTILLO
NOMBRES
VALENTINA CECILIA
NACIONALIDAD SEXO
CHILENA M
FECHA DE NACIMIENTO NÚMERO DOCUMENTO
25 JUL 1896 764.945.364
FECHA DE EMISIÓN FECHA DE VENCIMIENTO
29 JUL 2016 25 JUL 2026
FIRMA DEL TITULAR
No firma
RUN 24.875.697-K

CÉDULA DE IDENTIDAD
REPÚBLICA DE CHILE
SERVICIO DE REGISTRO CIVIL E IDENTIFICACIÓN
APELLIDOS
LOBOS
ACEITUNO
NOMBRES
MARGARITA VALERIA
NACIONALIDAD SEXO
CHILENA M
FECHA DE NACIMIENTO NÚMERO DOCUMENTO
25 JUL 1896 254.698.325
FECHA DE EMISIÓN FECHA DE VENCIMIENTO
29 JUL 2016 25 JUL 2026
FIRMA DEL TITULAR
No firma
RUN 17.975.483-K

KINO
371
02/11/97
CARTON Nº
TG 01.555.105
027056963405
01 04 05
06 07 11
12 14 16
18 20 21
22 23 25
$200
EZYVO7ERX4XLT59KFKTSXW37TOH3

V3SCNDIV6OPJXLOR84UNLE94DEY7

M4XXSKTNFAE92DKDQV6BOT76DFR6

Arquitectura y clima: manuales actualizados de diseño bioclimático para arquitectos y urbanistas

América Méndez

Stadium, pabellón de Chile en la Bienal de Venecia 2018, se presenta a través de la construcción de un modelo a gran escala del Estadio Nacional de Santiago, con el objetivo de ilustrar las políticas locales de vivienda. La exposición, comisariada por la arquitecta Alejandra Celedón, busca que estos elementos –frágiles y pesados a la vez– cuenten la historia del momento en que los ciudadanos 'sin casa' pasan a ser propietarios por primera vez, a modo de "fragmentos simbólicos de esta transmutación". Descripción por el equipo. Al centro de Sala dell'Isoloto se ubica un modelo a gran escala de un edificio hecho de tierra apisonada. Tras un examen más minucioso, las sesenta piezas que componen la forma ovalada del edificio ya no parecen hechas de tierra, sino más bien talladas en ella. Las distintas capas con ligeras variaciones de color y textura nos recuerdan que es el suelo lo que está en juego en el Pabellón de Chile.

Teorías e historia de la arquitectura: hacia una nueva concepción de los espacios arquitectónicos

Adriana Rivas

La primera ocupa el vestíbulo del pabellón, y está dedicada a la investigación de archivo y a los hallazgos que rodean el evento. En el muro opuesto al acceso, la planta de STADIUM con la ciudad inscrita se exhibe como origen de la exposición y como preludio de su materialización. A la izquierda, el material audiovisual narra la historia y el contexto del evento a través de tres soportes diferentes: la prensa y la propaganda, la cobertura televisiva y los discursos, y el fascículo. Una vez dentro de la sala principal, el STADIUM de 7 x 5 metros emerge al centro. A su derecha, la pared sur exhibe un segundo momento, 'las islas', que amplía —mediante entrevistas a pobladores presentes ese día en el evento— las historias de tres de las sesenta piezas del Estadio: La Pincoya, Villa Francia y La Bandera.

Saber ver la arquitectura: ensayo sobre la interpretación semiótica y espacial de la arquitectura

Verónica Ledo

'El horizonte' ocupa la pared norte y está dedicado a los múltiples estadios del Estadio a través de una película basada en registros de archivo. Este devuelve al edificio su contexto, como testigo de la historia de la ciudad, revisando otros eventos y figuras importantes en el Estadio. El muro de fondo de STADIUM, una ciudad en un edificio muestra una representación de Santiago con la ubicación precisa de las piezas del Estadio, un edificio en una ciudad. Este muro ofrece una versión espejada de la pieza central, su contraparte narrativa. La estructura representa la distancia entre el Estadio (en el centro de la ciudad) y las poblaciones (en las afueras de la ciudad), con el edificio como parte y testigo del desarrollo desigual de Santiago.

FIGURE 3. The samples used for the user studies: business cards (top-left), document IDs (top-center), lottery cards (top-right), papers (bottom). Although texts are different among comparable samples, they have the same number of character and special character for any of the data to be transcribed.

and QWERTY Physical Keyboard - were established using a Latin Square [29].

Regarding the Google Keyboard condition, we disabled the personalized suggestions option so that the keyboard could not learn any suggestions. This prevented the last participants from having suggestions that the keyboard could have learned from the previous participants. All other settings were unchanged.

After each transcription activity, we asked participants to complete a post-task questionnaire using the first two questions of the After Scenario Questionnaire [30], which measures the participants' satisfaction with the ease of completing the task, and the satisfaction with the amount of time it took to complete the task. We left out the third, and last, question because it is out of the experiment scope, since it asks satisfaction with supporting information (e.g. on-line help) when completing the task. At the end of the experiment, we asked users to complete the System Usability Scale [31] questionnaire and write any comments that they had about CameraKeyboard.

C. MATERIAL DESIGN AND TASK RATIONALE

We designed three different business cards, document IDs, lottery cards, and papers, so that participants transcribed a different text in each of the conditions (Fig. 3). This prevented participants from remembering the text in the subsequent conditions. Although the sample documents were all different, they had the same number of characters and special characters so that they could be comparable.

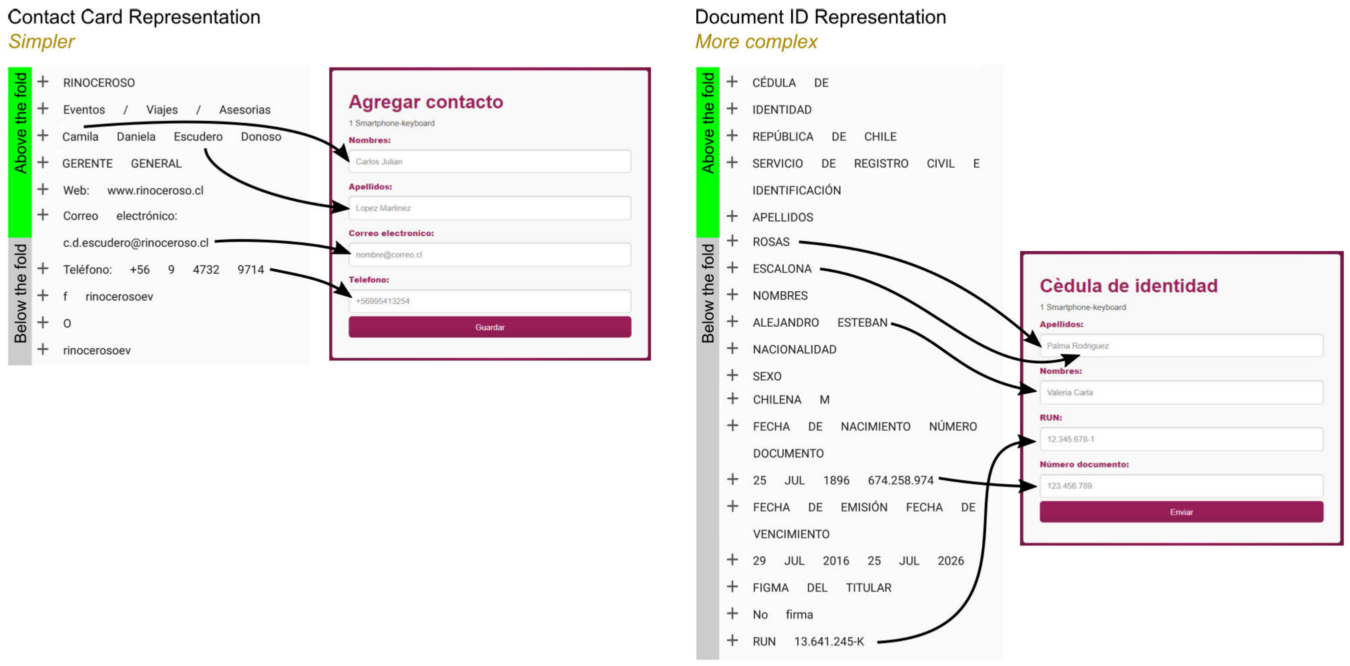


FIGURE 4. The Document ID has a more complex structure than the Business Card. First, the elements of the Contact Card are in the same order as the fields in the form. Instead, the elements of the Document ID are not in the same order of the fields in the form, e.g., the RUN, which in the document is the last element, is the next-to-last field in the form. On the contrary, the document number, which in the document is the next-to-last element, is the last field in the form. Second, in the Document ID there is much more information to search and this may slow down and puzzle users. Finally, in the document, the first elements to insert (i.e., the surnames ROSAS ESCALONA) is below the fold, namely, the users need to scroll down to find them, and this may make the interaction and the search more difficult.

We chose four tasks to represent different text-inputting scenarios.

Task 1 consists of the transcription of a simple structured text where the text must be transcribed in the same order in which it appears. In contrast, task 2 consists of the transcription of a more complex structured text where the text must be transcribed in a different order than the order in which it appears. We expected a significant difference between these two types of structured texts since we believe that it is harder to find a specific text within several pieces of information that are not of interest, and it is also harder to transcribe information when it is not in the same order as the form. Fig. 4 shows the detailed differences between the first two tasks.

Task 3 and 4 were chosen to compare users' performances regarding the transcription of alphanumeric texts versus natural texts. Alphanumeric texts are harder to recall due to their arbitrary nature [32], and we expect CameraKeyboard to be particularly helpful for alphanumeric texts.

D. APPLICATION FOR COLLECTING DATA

Participants transcribed the required text in a web-based application (Fig. 5) that recorded all user input (keyboard and click/tap events). Accordingly, we could extract errors and the time of execution for any task, including the exact timing for each of the characters or words that were transcribed. Regarding errors, they were considered for each field, namely, one or more errors on a field were counted as one error. Moreover, we did not consider differences between uppercase and lowercase characters.

The web application works both on smartphone browsers (used for CameraKeyboard and Google Keyboard) and desktop browsers (used for the physical keyboard). In each of the forms, the timer started when participants focused on a field, and ended when participants submitted the form.

E. DATA ANALYSIS

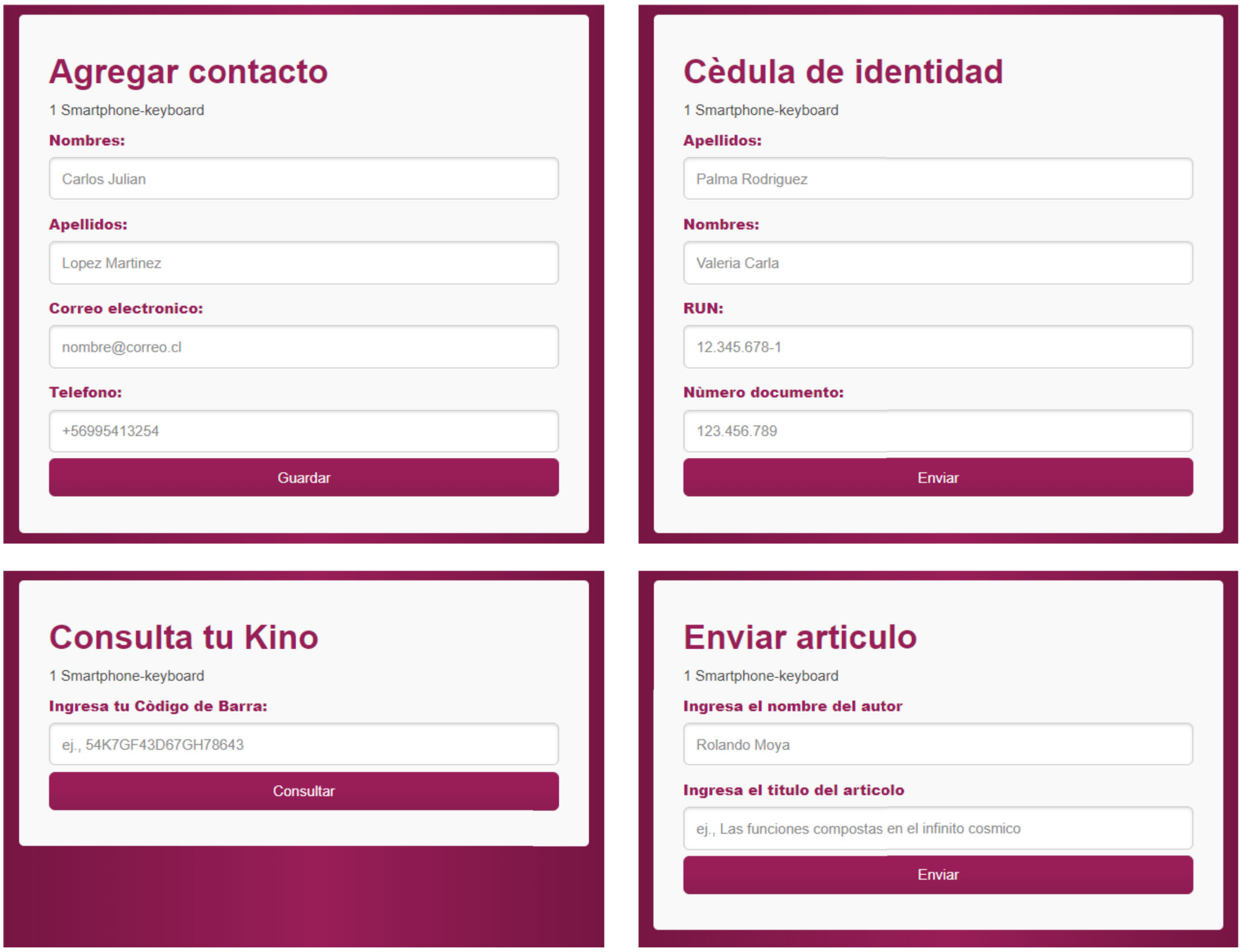
Since most of the data were not normal (Shapiro-Wilk test), we performed the Friedman test to detect significant differences between conditions both for time of execution and post-task questionnaires. When differences could be detected, we performed post-hoc analysis with Wilcoxon signed-rank tests conducted with a Bonferroni correction applied, resulting in a significance level set at $p < 0.017$.

Regarding the SUS questionnaire, as proposed by their authors, we summed up the score of the all responses and multiplied them by 2.5 so that we could obtain a score between 0 and 100 for any participant. Then, we computed the average of participants' scores, to have an overall score representing the usability of CameraKeyboard.

V. RESEARCH QUESTIONS

The four tasks of our user study are aimed at answering four different research questions, as follows.

1) Which (out of Google Keyboard, Physical QWERTY Keyboard, or CameraKeyboard) is the best (considering speed and error rate) way to transcribe data from a simple, structured source in which the fields must be



Agregar contacto

1 Smartphone-keyboard

Nombres: Carlos Julian

Apellidos: Lopez Martinez

Correo electronico: nombre@correo.cl

Telefono: +56995413254

Guardar

Cèdula de identidad

1 Smartphone-keyboard

Apellidos: Palma Rodriguez

Nombres: Valeria Carla

RUN: 12.345.678-1

Nùmero documento: 123.456.789

Enviar

Consulta tu Kino

1 Smartphone-keyboard

Ingresa tu Còdigo de Barra: ej., 54K7GF43D67GH78643

Consultar

Enviar articulo

1 Smartphone-keyboard

Ingresa el nombre del autor: Rolando Moya

Ingresa el titulo del articolo: ej., Las funciones compostas en el infinito cosmico

Enviar

FIGURE 5. **Forms of the application designed for collecting data.**

transcribed in order? We answer this question through Task 1 (contact card).

2) Which (out of Google Keyboard, Physical QWERTY Keyboard, or CameraKeyboard) is the best (considering speed and error rate) way to transcribe data from a more complex structured source in which the fields must be transcribed in a different order than how they appear on the source? We answer this question through Task 2 (document ID).
3) Which (out of Google Keyboard, Physical QWERTY Keyboard, or CameraKeyboard) is the best (considering speed and error rate) way to transcribe sequential alphanumeric codes? We answer this question through Task 3 (lottery code).
4) Which (out of Google Keyboard, Physical QWERTY Keyboard, or CameraKeyboard) is the best (considering speed and error rate) way to transcribe natural language text? We answer this question through Task 4 (paper title and author).

The results for each of the tasks are presented in Section VI and discussed in Section VII.

VI. RESULTS

A. TASK 1 - TRANSCRIBING BUSINESS CARD DATA

When transcribing contact data from a business card, the fastest keyboard was CameraKeyboard, followed by the physical keyboard and Google Keyboard, with the averages, respectively, of 42, 51 and 58 seconds to complete the form. The Friedman test detected significant differences among the conditions ($X2(2) = 24.029$, $p = 0.001$), and post hoc analysis with Wilcoxon signed-rank revealed significant differences between CameraKeyboard and the physical keyboard ($Z = -2.992$, $p = 0.003$), CameraKeyboard and Google Keyboard ($Z = -3.622$, $p = 0.001$), and Google Keyboard and the physical keyboard ($Z = -2.380$, $p = 0.017$).

The post-task questionnaire revealed that CameraKeyboard was also perceived to be the fastest, followed by the physical keyboard and Google Keyboard, with medians of 5, 4, and 3 respectively. The Friedman test detected significant differences among the conditions ($X2(2) = 28.441$, $p = 0.001$), and post hoc analysis with Wilcoxon signed-rank revealed significant differences between CameraKeyboard and the physical keyboard ($Z = -3.745$,

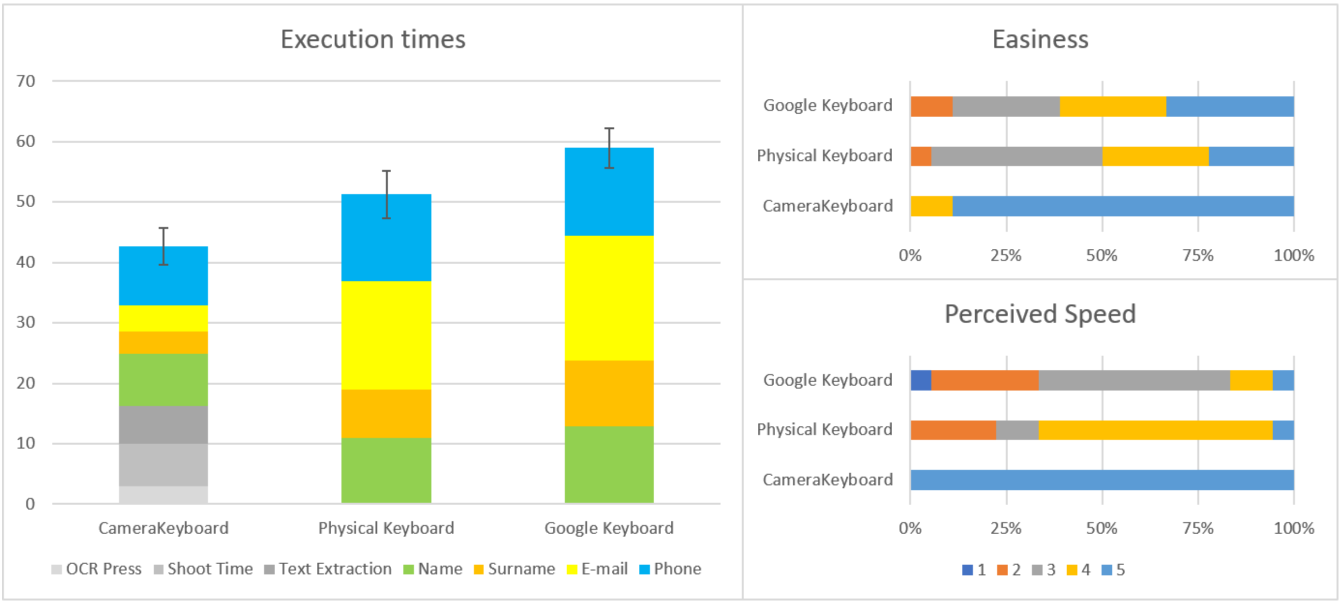


FIGURE 6. Execution times, perceived easiness and speed when transcribing data of a contact. Error bars in the left graph represent 80% CI. Values in the right graphs range between 1 (strong disagreement) and 5 (strong agreement).

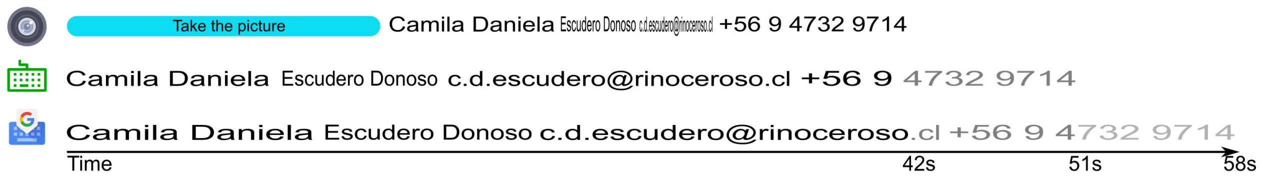

Take the picture
Camila Daniela Escudero Donoso c.d.escudero@rinoceroso.cl +56 9 4732 9714
Camila Daniela Escudero Donoso c.d.escudero@rinoceroso.cl +56 9 4732 9714
Camila Daniela Escudero Donoso c.d.escudero@rinoceroso.cl +56 9 4732 9714
Time
42s
51s
58s

FIGURE 7. The figure compares the time for text entry for all types of keyboards (CameraKeyboard, the physical keyboard and Google Keyboard from top to bottom) in the contact card task. Words are stretched to graphically represent the portion of time needed for their entry. The text is just an example: each contact card had the same length of text to be comparable.

p = 0.001), CameraKeyboard and Google Keyboard (Z = −3.695, p = 0.001), and Google Keyboard and the physical keyboard (Z = −2.556, p = 0.011).

The post-task questionnaire revealed that CameraKeyboard was perceived as the easiest to use, followed by Google Keyboard and the physical keyboard, with medians of 5, 4, and 3.5 respectively. The Friedman test detected significant differences among the conditions (X2(2) = 17.491, p = 0.001), and post hoc analysis with Wilcoxon signed-rank revealed significant differences between CameraKeyboard and the physical keyboard (Z = −3.397, p = 0.001), CameraKeyboard and Google Keyboard (Z = −3.002, p = 0.003), but not between Google Keyboard and the physical keyboard (Z = −0.632, p = 0.527).

All results are displayed in Fig. 6.

Regarding errors, the transcriptions using Google Keyboard and CameraKeyboard had 2 errors each (i.e., 0.11 errors per participant), while the physical keyboard had 1 error (i.e., 0.05 errors per participant). The low number of errors do not allow us to perform inferential tests to understand if differences are significant.

The length of the text to be transcribed may influence which keyboard is the fastest. To examine this, we found the point at which CameraKeyboard becomes more efficient than the other keyboards. CameraKeyboard surpasses the physical keyboard and Google Keyboard when more than 5 blocks of text have to be transcribed. Otherwise, the other keyboards - at least under the perspective of task completion time - are more convenient than CameraKeyboard (see Fig. 7). The fifth block corresponds to the email. Note that we consider a block of text a continuous sequence of characters between two blank spaces (e.g., c.d.escudero@rinoceroso.cl and +56 in Fig. 7 are two different blocks of text).

B. TASK 2 - TRANSCRIBING DOCUMENT ID DATA

When transcribing data from a document ID, the fastest keyboard was the physical keyboard, followed by CameraKeyboard and Google Keyboard, with respective averages of 41, 47 and 50 seconds to complete the form. The Friedman test detected significant differences among the conditions (X2(2) = 6.478, p = 0.039), and post hoc analysis with Wilcoxon signed-rank revealed significant differences between Google Keyboard and the physical keyboard (Z = −2.560, p = 0.010), but not between CameraKeyboard and Google Keyboard (Z = −1.042, p = 0.297), and CameraKeyboard and the physical keyboard (Z = −1.683, p = 0.092).

The post-task questionnaire revealed that CameraKeyboard was perceived as the fastest, followed by the physical keyboard and Google Keyboard, with medians of 5, 4, and

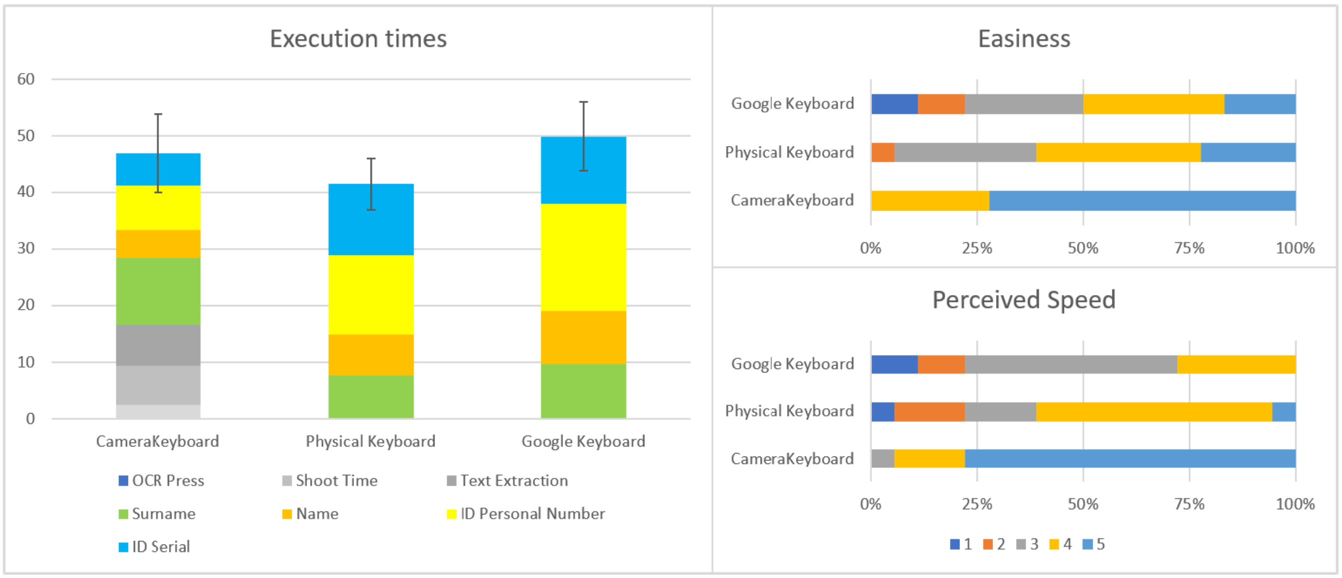


FIGURE 8. Execution times, perceived easiness and speed when transcribing data of a document id. Error bars in the left graph represent 80% CI. Values in the right graphs range between 1 (strong disagreement) and 5 (strong agreement).

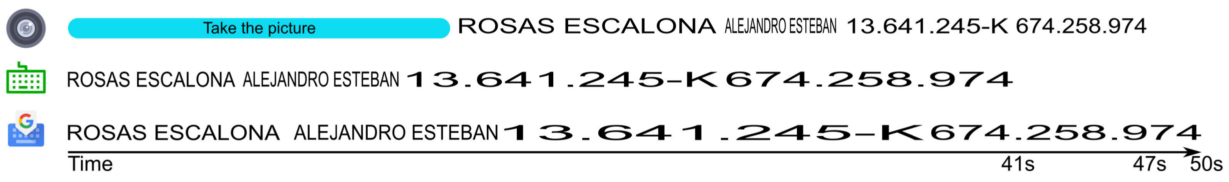

Take the picture
ROSAS ESCALONA ALEJANDRO ESTEBAN 13.641.245-K 674.258.974
ROSAS ESCALONA ALEJANDRO ESTEBAN 13.641.245-K 674.258.974
ROSAS ESCALONA ALEJANDRO ESTEBAN 13.641.245-K 674.258.974
Time
41s
47s
50s

FIGURE 9. The figure compares the time for text entry for all types of keyboards in the document ID task. Words are stretched to graphically represent the portion of time needed for their entry. The text is is just an example: each document ID had the same length of text to be comparable.

3 respectively. The Friedman test detected significant differences among the conditions (X2(2) = 24.366, p = 0.001), and post hoc analysis with Wilcoxon signed-rank revealed significant differences between CameraKeyboard and the physical keyboard (Z = −3.206, p = 0.001), CameraKeyboard and Google Keyboard (Z = −3.680, p = 0.001), but not between Google Keyboard and the physical keyboard (Z = −1.999, p = 0.046).

The post task questionnaire revealed that CameraKeyboard was perceived as the easiest to use, followed by the physical keyboard and Google Keyboard, with medians of 5, 4, and 3.5 respectively. The Friedman test detected significant differences among the conditions (X2(2) = 20.462, p = 0.001), and post hoc analysis with Wilcoxon signed-rank revealed significant differences between CameraKeyboard and the physical keyboard (Z = −3.139, p = 0.002), CameraKeyboard and Google Keyboard (Z = −3.370, p = 0.001), but not between Google Keyboard and the physical keyboard (Z = −1.456, p = 0.145).

All results are displayed in Fig. 8.

Regarding errors, the transcriptions using Google Keyboard and the physical keyboard had 3 errors each (i.e., 0.16 errors per participant), while CameraKeyboard had no errors. The low number of errors does not allow us to perform inferential tests to understand if differences are significant.

CameraKeyboard surpasses Google Keyboard only when more than 6 blocks of text (i.e., all the text in the task) have to be transcribed. Otherwise, the Google Keyboard - at least under the perspective of task completion time - is more convenient than CameraKeyboard (see Fig. 9). However, the physical keyboard is the most convenient in any case.

C. TASK 3 - TRANSCRIBING A LOTTERY CODE

When transcribing the code from a lottery card, the fastest was CameraKeyboard, followed by the physical keyboard and Google Keyboard, with the averages, respectively, of 21, 39 and 52 seconds to complete the form. The Friedman test detected significant differences among the conditions (X2(2) = 34.111, p = 0.001), and post hoc analysis with Wilcoxon signed-rank revealed significant differences between CameraKeyboard and the physical keyboard (Z = −3.728, p = 0.001), CameraKeyboard and Google Keyboard (Z = −3.732, p = 0.001), and Google Keyboard and the physical keyboard (Z = −3.641, p = 0.001).

The post task questionnaire revealed that CameraKeyboard was perceived as the fastest, followed by the physical keyboard and Google Keyboard, with medians of 5, 2, and 2 respectively. The Friedman test detected significant differences among the conditions (X2(2) = 27.169, p = 0.001), and post hoc analysis with Wilcoxon signed-rank revealed

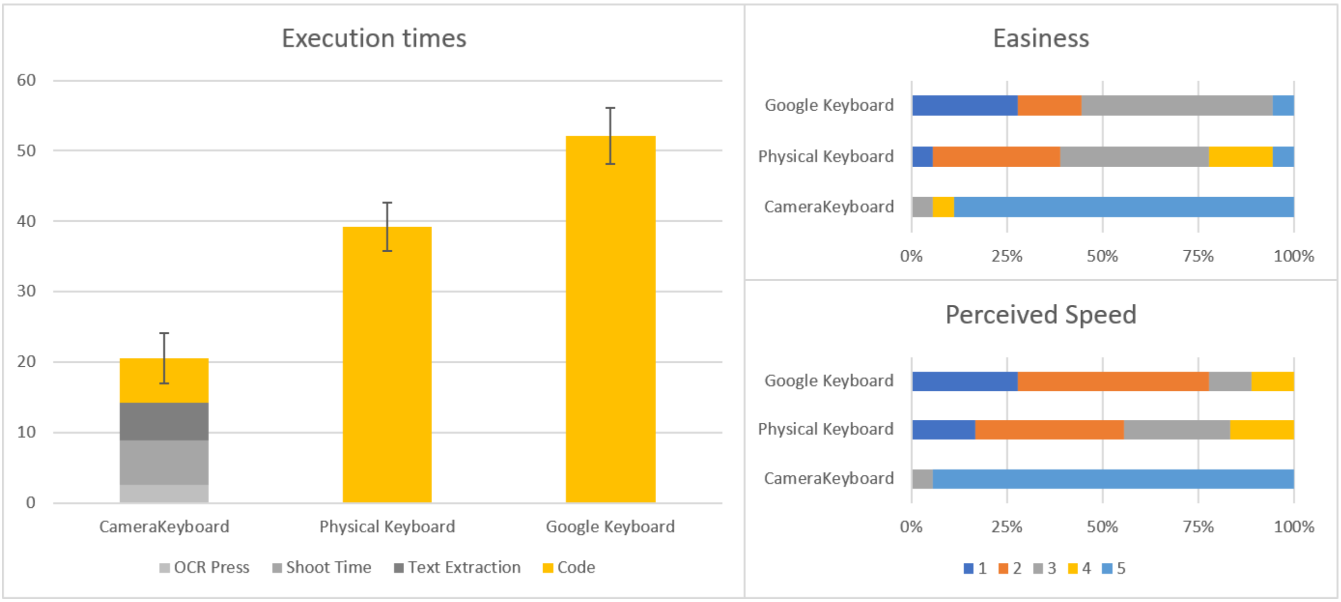


FIGURE 10. Execution times, perceived easiness and speed when transcribing the code of a lottery card. Error bars in the left graph represent 80% CI. Values in the right graphs range between 1 (strong disagreement) and 5 (strong agreement).

significant differences between CameraKeyboard and the physical keyboard (Z = −3.754, p = 0.001), CameraKeyboard and Google Keyboard (Z = −3.721, p = 0.001), but not between Google Keyboard and the physical keyboard (Z = −1.178, p = 0.178).

Moreover, the post task questionnaire revealed that CameraKeyboard was perceived as easiest to use, followed by the physical keyboard and Google Keyboard, with medians of 5, 3, and 3 respectively. The Friedman test detected significant differences among the conditions (X2(2) = 25.754, p = 0.001), and post hoc analysis with Wilcoxon signed-rank revealed significant differences between CameraKeyboard and the physical keyboard (Z = −3.660, p = 0.001), CameraKeyboard and Google Keyboard (Z = −3.564, p = 0.001), but not between Google Keyboard and the physical keyboard (Z = −1.370, p = 0.171).

All results are displayed in Fig. 10.

Regarding errors, the transcriptions using Google Keyboard had 3 errors (i.e., 0.16 errors per participant), the physical keyboard had 2 errors (i.e., 0.11 errors per participant), while CameraKeyboard had 6 errors (i.e., 0.33 errors per participant).

CameraKeyboard surpasses the physical keyboard when more than 16 alphanumeric characters have to be transcribed. Moreover, CameraKeyboard surpasses Google Keyboard only when more than 10 alphanumeric characters have to be transcribed. Otherwise, the other keyboards - at least under the perspective of task completion time - are more convenient than CameraKeyboard (see Fig. 11).

D. TASK 4 - TRANSCRIBING TITLE AND AUTHOR OF A PAPER

When transcribing the title and author of a paper, the fastest was CameraKeyboard, followed by the physical keyboard and Google Keyboard, with averages, respectively, of 38, 58 and 61 seconds to complete the form. The Friedman test detected significant differences among the conditions (X2(2) = 21.333, p = 0.001), and post hoc analysis with Wilcoxon signed-rank revealed significant differences between CameraKeyboard and the physical keyboard (Z = −3.162, p = 0.002), CameraKeyboard and Google Keyboard (Z = −3.724, p = 0.001), and Google Keyboard and the physical keyboard (Z = −0.588, p = 0.556).

The post task questionnaire revealed that CameraKeyboard was perceived as the fastest, followed by the physical keyboard and Google Keyboard, with the medians of 5, 2, and 2 respectively. The Friedman test detected significant differences among the conditions (X2(2) = 27.029, p = 0.001), and post hoc analysis with Wilcoxon signed-rank revealed significant differences between CameraKeyboard and the physical keyboard (Z = −3.684, p = 0.001), CameraKeyboard and Google Keyboard (Z = −3.767, p = 0.001) but not between Google Keyboard and the physical keyboard (Z = −0.362, p = 0.717).

Moreover, the post task questionnaire revealed that CameraKeyboard was perceived as easiest to use, followed by the physical keyboard and Google Keyboard, with medians of 5, 3, and 3 respectively. The Friedman test detected significant differences among the conditions (X2(2) = 19.069, p = 0.001), and post hoc analysis with Wilcoxon signed-rank revealed significant differences between CameraKeyboard and the physical keyboard (Z = −3.345, p = 0.001), CameraKeyboard and Google Keyboard (Z = −3.241, p = 0.001), but not between Google Keyboard and the physical keyboard (Z = −0.580, p = 0.553).

All results are displayed in Fig. 10.

Regarding errors, all input methods had 4 errors each (i.e., 0.22 errors per participant).

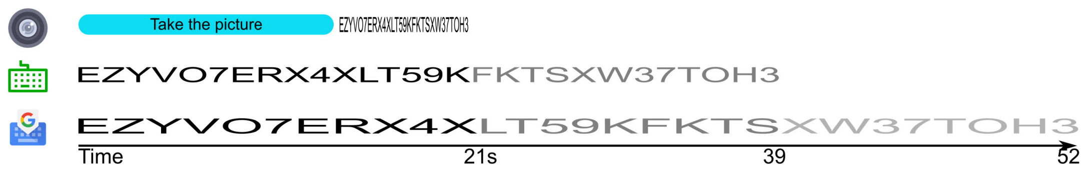

Take the picture
EZYVO7ERX4XLT59KFKTSXW37TOH3
EZYVO7ERX4XLT59KFKTSXW37TOH3
EZYVO7ERX4XLT59KFKTSXW37TOH3
Time
21s
39
52

FIGURE 11. The figure compares the time for text entry for all types of keyboards in the lottery code task. Words are stretched to graphically represent the portion of time needed for their entry. The text is is just an example: each lottery code had the same length of text to be comparable.

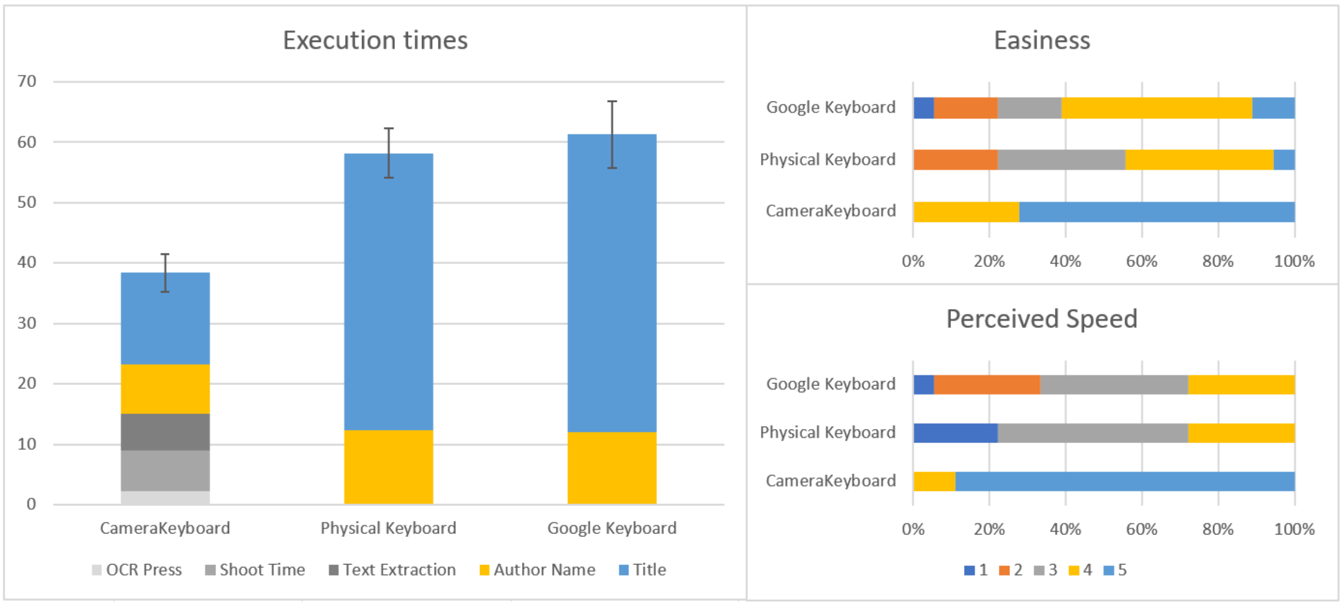


FIGURE 12. Execution times, perceived easiness and speed when transcribing the title and author of a paper. Error bars in the left graph represent 80% CI. Values in the right graphs range between 1 (strong disagreement) and 5 (strong agreement).

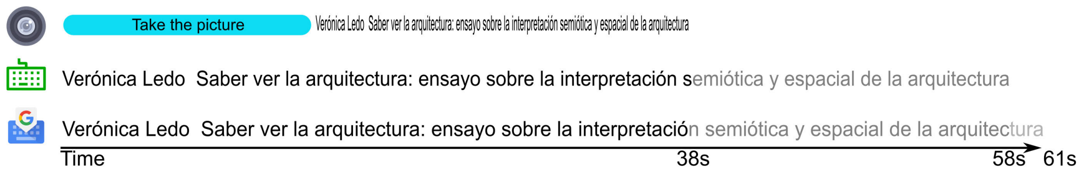

Take the picture
Verónica Ledo Saber ver la arquitectura: ensayo sobre la interpretación semiótica y espacial de la arquitectura
Verónica Ledo Saber ver la arquitectura: ensayo sobre la interpretación semiótica y espacial de la arquitectura
Verónica Ledo Saber ver la arquitectura: ensayo sobre la interpretación semiótica y espacial de la arquitectura
Time
38s
58s
61s

FIGURE 13. The figure shows the time for text entry for all types of keyboards in the natural language task. Words are stretched to graphically represent the portion of time needed for their entry. The text is is just an example: each paper title and author had the same length of text to be comparable.

CameraKeyboard surpasses the physical keyboard only when more than 72 characters of natural language (i.e., *"s"* of *"semiotica"* in Fig. 13) have to be transcribed. Moreover, CameraKeyboard surpasses Google Keyboard only when more than 69 characters of natural language (i.e., *"o"* of *"interpretacion"* in Fig. 13) have to be transcribed. Otherwise, the other keyboards - at least under the perspective of task completion time - are more convenient than CameraKeyboard (see Fig. 13).

E. SUS QUESTIONNAIRE RESULTS

The average score of the SUS questionnaire for CameraKeyboard was 90. The distribution of the scores is displayed in Fig 14. Scores above 85 are associated with the adjective "excellent". In terms of acceptability, scores above 72 are considered "acceptable" [33]. A strong correlation between SUS and the Net Promoter Score has been found - scores above 78 are given by users who would recommend the system to others [34].

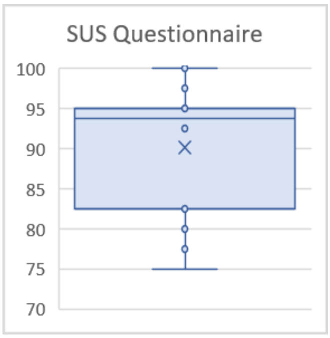


FIGURE 14. SUS questionnaire results.

F. USER FEEDBACK

Out of the 18 participants, 17 left written feedback about CameraKeyboard. We categorized these comments as: positive comments, suggestions for improvement, and problems the participants faced when interacting with CameraKeyboard. We discuss this feedback in this section, identifying participants by a number (P1 to P18). Most users had positive comments: they highlighted CameraKeyboard's speed, especially when transcribing long information (P1, P2, P3, P4, P7, P9, P10, P14) in error-prone data (P1, P2, P3). They thought it was useful (P2, P4, P5, P13), easy to use (P2, P13, P16) and learn (P3), comfortable (P10), and would like to have it on their own smartphone (P16). One user highlighted the possibility of using the transcribed data, e.g. to share it through e-mail or WhatsApp (P1), and another liked not having to switch between alphabetic and numerical keyboards when inputting mixed-type data (P2).

The participants also discussed some problems that they had when trying out the application, although these were not widespread and only mentioned by one or two users each. Two users found that in the business card, the number was not recognised as a single line (P5, P9). Other issues were that one user used a double tap to focus the picture, finding that this closed the camera (P6), that the keyboard requires an internet connection (P10), that occasionally there were precision problems (P11, P15), and one user found it difficult to switch between the two keyboards to correct small errors (P18).

Regarding improvements, users gave us some feedback on how to improve CameraKeyboard: allow adjusting picture brightness to improve recognition (P3), make the CameraKeyboard button more noticeable (P7), improve spacing (P12), link fields with data when there is a direct match (P16), and allow taking more than one photo (P16, P17). One user also would like to use CameraKeyboard to recognise and digitalise handwritten text (P7).

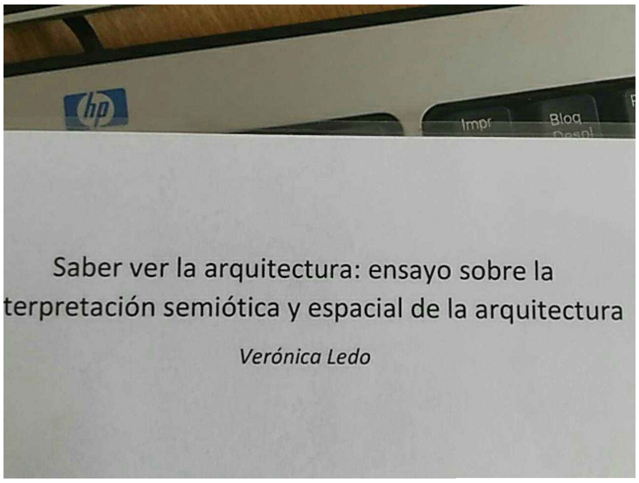
G. OBSERVATION

We explained to users that they could be free to take pictures in landscape or portrait modes. However, we observed that in most of the cases, they used the landscape orientation. Regarding this issue, our data on each task is the following:

- Business Card: 6 Portrait / 12 Landscape
- Document ID: 6 Portrait / 12 Landscape
- Lottery code: 12 Portrait / 6 Landscape
- Paper data: 6 Portrait / 12 Landscape

Some of the errors in the use of CameraKeyboard were caused by users who put the camera too close to the text to be recognised. In this way, some characters were not in the frame and could not be recognised. In Fig 15, we show an example of a picture taken by one of our participants that caused a recognition error.

Regarding Google Keyboard, almost all users used the suggestion bar on the top of the keyboard. Moreover, 16 users transcribed words using keyboard letters as usual, whereas the remaining two users mostly used Glide Typing (sequence of continuous swipes over the letters that compose a word) to transcribe words.



Saber ver la arquitectura: ensayo sobre la
terpretación semiótica y espacial de la arquitectura
Verónica Ledo

FIGURE 15. **In this image, the first two characters of the second line (i.e., "in") were not recognised.**

Although P18 stated that they didn't know how to get back to the normal keyboard, we observed that another user realized there was an error and corrected it by using the normal keyboard (see button in Fig. 2-D).

VII. DISCUSSION

A. CAMERAKEYBOARD SPEED

1) OVERALL SPEED

First of all, we must point out that CameraKeyboard makes sense only when the text to transcribe is not too short. CameraKeyboard is quite fast for text entry tasks (see Figures 7, 9, 11 and 13), but a significant amount of time is required to take the picture. Therefore, the amount of time required to write short texts is usually lower than the time required to (1) enable the camera, (2) take a picture and (3) wait for text extraction; and this (long) process causes a small advantage for the other keyboards in terms of time.

Moreover, it must be noted that execution times do not depend too much on the length of the text to be transcribed. Rather, execution time depends on:

1) the ability of the user to recognise the different extracted blocks of texts and;
2) and the number of blocks of text to be transcribed.

Regarding block text recognition (point one noted above), users waste a significant amount of time searching for the first block of text (see the time needed to enter the first block of text in Figures 6, 8, 10, and 12). However, after looking for the first block of text, the remaining ones are recognised faster because users take the first block of text as an initial starting point to search for the remaining blocks.

Regarding the number of blocks (point two noted above), we note that - leaving out the time to search for such blocks - the entry of a block requires just a tap, namely, it is equivalent to writing a single character.

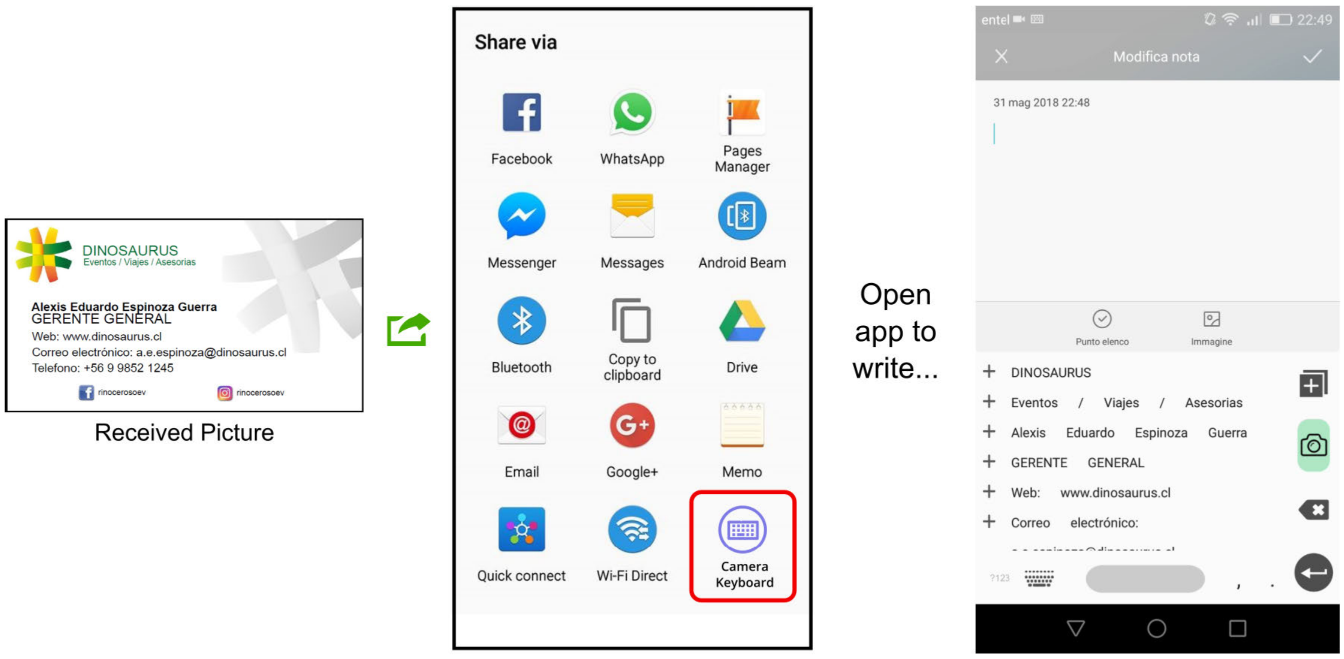

DINOSAURUS
Eventos / Viajes / Asesorias
Alexis Eduardo Espinoza Guerra
GERENTE GENERAL
Web: www.dinosaurus.cl
Correo electrónico: a.e.espinoza@dinosaurus.cl
Telefono: +56 9 9852 1245

FIGURE 16. **A user may share a picture to CameraKeyboard, open the app where the text will be used, and enter the extracted text.**

2) CAMERAKEYBOARD SPEED FOR SPECIFIC TASKS

We had notable differences between task 1 and task 2 due to the different levels of complexity of the sources. In the document ID task (task 2), the position of the surnames - below the fold, see Fig. 4 - somehow puzzled the users, who had to scroll down to find them. Overall, the users spent more time searching for those surnames on the CameraKeyboard interface rather than writing them using the other two keyboards (see green areas in the execution time graph in Fig. 8). In contrast, in the contact card task (task 1), searching for the names was faster than transcribing them (see green areas in the execution time graph in Fig. 6) - probably because names were above the fold. The order of the field also impacted execution times. The time required to transcribe the email of the contact card (task 1) was very short in comparison to the other keyboards since it was in the same order as the form fields and the structure was not complex (see yellow areas in the execution time graph in Fig. 6). This notable difference is not appreciable in the document ID task (task 2, see yellow areas in the execution time graph in Fig. 8), probably because the fields were not in the same order as the source.

Overall, searching for a text that is displayed in the same order as the original source is generally easier than transcribing a text with traditional keyboards. In fact, according to post-task questionnaires, CameraKeyboard was preferred also for transcribing data from complex structures, e.g. the document ID (see Fig. 8). Accordingly, CameraKeyboard was cognitively less demanding than the other keyboards. CameraKeyboard was slower than the physical keyboard (see Fig. 8), but it was perceived as faster by users. These contrasting results - perceived speed does not correspond with actual speed - are, from our point of view, justified by the ease of use of CameraKeyboard, which is related to the ease of recognition over recall [28].

The transcription of alphanumeric characters requires other considerations. Recalling alphanumeric characters (task 3) is cognitively more demanding than recalling natural text (task 4) because of their arbitrary nature [32]. The higher cognitive demand can be inferred from the users' responses on easiness and perceived speed, where differences are notably in favour of CameraKeyboard (see Fig. 10). As a matter of fact, using CameraKeyboard, the transcription of the alphanumeric code required just a tap since users just need to recognise the code. Instead, with the QWERTY keyboards, users needed to continuously switch their attention between the lottery code and the keyboard and were able to transcribe and recall 4-6 letters at a time.

Using CameraKeyboard, users were faster at transcribing natural texts (Task 4), but the difference was not as large as when transcribing alphanumeric codes (see Fig. 12). In fact, using the normal QWERTY keyboard, users were able to recall more text, and the switching of attention between the article and the keyboard was not so frequent as in the alphanumeric code.

Finally, we can conclude that CameraKeyboard was faster, but the benefits are overall greater when transcribing alphanumeric codes.

B. CAMERAKEYBOARD ERROR RATES

Considering the low number of errors, we could not state that an OCR-based automatic solution such as CameraKeyboard is less error-prone than manual text entry through traditional mobile or physical keyboards. Rather, both techniques have

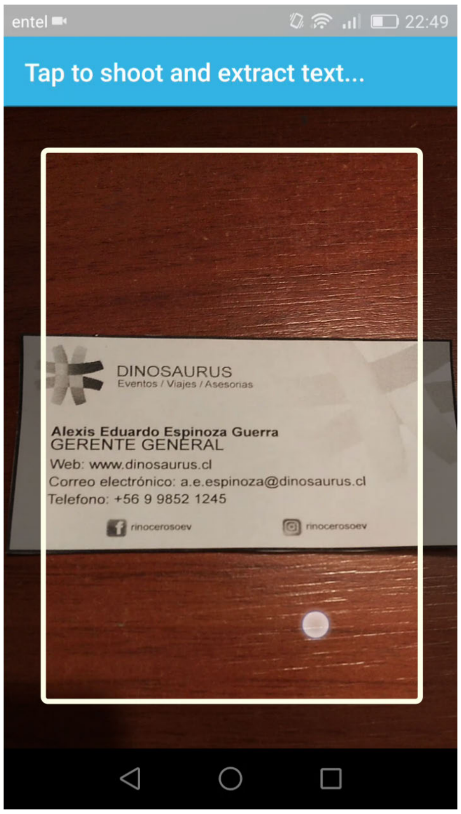

entel
22:49
Tap to shoot and extract text...
DINOSAURUS
Eventos / Viajes / Asesorias
Alexis Eduardo Espinoza Guerra
GERENTE GENERAL
Web: www.dinosaurus.cl
Correo electrónico: a.e.espinoza@dinosaurus.cl
Telefono: +56 9 9852 1245
rinocerosoev
rinocerosoev

FIGURE 17. **An overlay frame on the camera image can help users select the text. This would prevent the text from being too close to the camera, causing recognition errors.**

a similar (and small) error rate. Most errors were caused by a single erroneous character. For example, in the contact card, CameraKeyboard did not identify a dot in the email address; in the physical keyboard, a user mistook one character in the email address; and in Google Keyboard, a user forgot one digit in the phone number field. One participant in particular stated out loud that they had mild dyslexia, and e.g. in the lottery code they wrote 25 characters with several mistakes, while the code was actually 28 characters long. In the case of conditions such as dyslexia, in which users may be more prone to errors, especially when transcribing a long sequence of characters with no semantic meaning, a tool like CameraKeyboard may be especially useful. Previous research has found that dyslexic users may benefit from specialised word processing software [35].

C. DESIGN IMPROVEMENTS

This section discusses possible design improvements that emerged after our design experience and user evaluation.

1) SHARING ALTERNATIVE

CameraKeyboard could also be used when a users receives a picture that contains text and they want to use the text without transcribing it manually. In this situation, the received picture could be shared to CameraKeyboard (Fig. 16-Left-Center) and, after opening the app where the text will be used, CameraKeyboard could display the extracted text (Fig. 16-Right).

2) OVERLAY FRAME

To avoid recognition errors caused by texts positioned out of the frame, such as the case discussed in Fig 15, an overlay frame like the one in Fig. 17 may be added to guide users to select the required text. This would prevent the text from being too close to the camera, causing recognition errors.

VIII. CONCLUSION

This paper presented CameraKeyboard, a text entry technique for smartphones that uses the smartphone camera as a keyboard, allowing users to digitalise text from physical sources.

We found that CameraKeyboard is generally faster than Google Keyboard even if its use only makes sense when the text to transcribe is not extremely short, since users take some time to open the camera, take the picture, and extract the needed text. However, when transcribing more than 5-6 words, CameraKeyboard was the best alternative. Regarding alphanumeric codes, CameraKeyboard resulted to be the best alternative only when transcribing more than 11 characters.

Errors with CameraKeyboard and traditional QWERTY alternatives were infrequent, except when users did not point the camera at the source text correctly (see Fig. 15).

SUS questionnaire results show that CameraKeyboard was well-received by users. Moreover, a post-task questionnaire also revealed that CameraKeyboard is preferred even when it is slower than Google Keyboard. This suggests that CameraKeyboard is cognitively less demanding than QWERTY alternatives.

Although Camerakeyboard is designed to be used with unstructured and structured data, there are considerable improvements that may be made when interacting with structured data. In this case, CameraKeyboard could be used to train an online crowd-powered machine learning system (e.g., [36], [37]) to automatically match source (the picture taken) and destination (the form to fill). For example, when faced with a new form to fill out, CameraKeyboard would make the first user to enter the data manually. However, for subsequent uses of the same form, CameraKeyboard could use progressive learning [38] to automatically fill in the form based on the procedure learned from the first user. In this way, the system could learn progressively directly from experience, potentially becoming able to match any new kind of structured source data with new kinds of structured destinations.

REFERENCES

[1] *Swiftkey*. Accessed: Jun. 17, 2019. [Online]. Available: https://play.google.com/store/apps/details?id=com.touchtype.swiftkey

[2] M. Romano, L. Paolino, G. Tortora, and G. Vitiello, "The tap and slide keyboard: A new interaction method for mobile device text entry," *Int. J. Hum.-Comput. Interact.*, vol. 30, no. 12, pp. 935–945, 2014, doi: 10.1080/10447318.2014.924349.

[3] M. Laine and O. S. Nevalainen, ''A standalone OCR system for mobile cameraphones,'' in *Proc. IEEE 17th Int. Symp. Pers., Indoor Mobile Radio Commun.*, Sep. 2006, pp. 1–5.
[4] J. Duffy. *The Best Mobile Scanning and OCR Apps*. Accessed: Jun. 17, 2019. [Online]. Available: https://zapier.com/blog/best-mobile-scanning-ocr-apps/
[5] X.-P. Luo, J. Li, and L.-X. Zhen, ''Design and implementation of a card reader based on build-in camera,'' in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, vol. 1, Aug. 2004, pp. 417–420.
[6] D. Pogue, ''Unpredictive text,'' *Sci. Amer.*, vol. 311, no. 6, p. 38, 2014.
[7] D. Wigdor, C. Forlines, P. Baudisch, J. Barnwell, and C. Shen, ''Lucid touch: A see-through mobile device,'' in *Proc. 20th Annu. ACM Symp. Interface Softw. Technol.*, Oct. 2007, pp. 269–278.
[8] M. M. Luna, F. A. A. de Melo Nunes Soares, H. A. D. do Nascimento, J. Siqueira, E. F. de Souza, T. H. Nascimento, and R. M. da Costa, ''Text entry on smartwatches: A systematic review of literature,'' in *Proc. IEEE 42nd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, vol. 2, Jul. 2018, pp. 272–277.
[9] P. Quinn and S. Zhai, ''A cost-benefit study of text entry suggestion interaction,'' in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2016, pp. 83–88.
[10] I. A. Sonaike, T. A. Bewaji, P. Ritchey, and S. C. Peres, ''The ergonomic impact of swype,'' in *Proc. Hum. Factors Ergonom. Soc. Annu. Meeting*, 2016, vol. 60, no. 1, pp. 1374–1378, doi: 10.1177/1541931213601317.
[11] M. S. Geary and M. Lind, ''Smart phone keyboard layout usability,'' *Int. J. Technol. Hum. Interact.*, vol. 14, no. 4, pp. 110–135, Oct. 2018.
[12] I. S. MacKenzie, H. Kober, D. Smith, T. Jones, and E. Skepner, ''LetterWise: Prefix-based disambiguation for mobile text input,'' in *Proc. 14th Annu. ACM Symp. Interface Softw. Technol.*, 2001, pp. 111–120.
[13] F. C. Y. Li, R. T. Guy, K. Yatani, and K. N. Truong, ''The 1line keyboard: A QWERTY layout in a single line,'' in *Proc. 24th Annu. ACM Symp. Interface Softw. Technol.*, 2011, pp. 461–470.
[14] J. Lai, D. Zhang, S. Wang, I. Y. Kilic, and L. Zhou, ''A thumb stroke-based virtual keyboard for sight-free text entry on touch-screen mobile phones,'' in *Proc. 51st Hawaii Int. Conf. Syst. Sci.*, 2018, pp. 1–10.
[15] N. Banovic, K. Yatani, and K. N. Truong, ''Escape-keyboard: A sight-free one-handed text entry method for mobile touch-screen devices,'' *Int. J. Mobile Hum. Comput. Interact.*, vol. 5, no. 3, pp. 42–61, 2013.
[16] K. Sun, C. Yu, and Y. Shi, ''Exploring low-occlusion qwerty soft keyboard using spatial landmarks,'' *ACM Trans. Comput.-Hum. Interact.*, vol. 26, no. 4, 2019, Art. no. 20.
[17] S. Zhu, T. Luo, X. Bi, and S. Zhai, ''Typing on an invisible keyboard,'' in *Proc. CHI Conf. Hum. Factors Comput. Syst. (CHI)*, New York, NY, USA, 2018, pp. 439:1–439:13, doi: 10.1145/3173574.3174013.
[18] S. Lee, T. Ha, and S.-J. Lim, ''Finger gesture input utilizing the rear camera of a mobile phone: A perspective of mobile cad,'' *Hum. Factors Ergonom. Manuf. Service Ind.*, vol. 28, no. 2, pp. 69–80, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/hfm.20724
[19] G. Costagliola, V. Fuccella, A. Leo, L. Lomasto, and S. Romano, ''The design and evaluation of a gestural keyboard for entering programming code on mobile devices,'' in *Proc. IEEE Symp. Vis. Lang. Hum.-Centric Comput. (VL/HCC)*, Oct. 2018, pp. 49–56.
[20] A. L. Smith and B. S. Chaparro, ''Smartphone text input method performance, usability, and preference with younger and older adults,'' *Hum. Factors, J. Hum. Factors Ergonom. Soc.*, vol. 57, no. 6, pp. 1015–1028, 2015, doi: 10.1177/0018720815575644.
[21] S. Ruan, J. O. Wobbrock, K. Liou, A. Ng, and J. A. Landay, ''Comparing speech and keyboard text entry for short messages in two languages on touchscreen phones,'' *Proc. Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 4, 2018, Art. no. 159.
[22] O. Krejcar, ''Smart implementation of text recognition (OCR) for smart mobile devices,'' in *Proc. 1st Int. Conf. Intell. Syst. Appl. (INTELLI)*, vol. 19, 2012, p. 24.
[23] M. Zhang, A. Joshi, R. Kadmawala, K. Dantu, S. Poduri, and G. S. Sukhatme, ''OCRdroid: A framework to digitize text using mobile phones,'' in *Proc. Int. Conf. Mobile Comput., Appl., Services*. San Diego, CA, USA: Springer, 2009, pp. 273–292.
[24] V. Fragoso, S. Gauglitz, S. Zamora, J. Kleban, and M. Turk, ''TranslatAR: A mobile augmented reality translator,'' in *Proc. IEEE Workshop Appl. Comput. Vis. (WACV)*, Jan. 2011, pp. 497–502.
[25] L. Neat, R. Peng, S. Qin, and R. Manduchi, ''Scene text access: A comparison of mobile OCR modalities for blind users,'' in *Proc. 24th Int. Conf. Intell. Interfaces (IUI)*, New York, NY, USA, 2019, pp. 197–207, doi: 10.1145/3301275.3302271.
[26] M. Cutter and R. Manduchi, ''Improving the accessibility of mobile OCR apps via interactive modalities,'' *ACM Trans. Accessible Comput.*, vol. 10, no. 4, pp. 11:1–11:27, Aug. 2017, doi: 10.1145/3075300.
[27] J. Johnson, *Designing With the Mind in Mind: Simple Guide to Understanding User Interface Design Rules*. Amsterdam, The Netherlands: Elsevier, 2013.
[28] J. Nielsen, ''Enhancing the explanatory power of usability heuristics,'' in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 1994, pp. 152–158.
[29] E. J. Williams, ''Experimental designs balanced for the estimation of residual effects of treatments,'' *Austral. J. Sci. Res.*, vol. 2, no. 2, pp. 149–168, Jun. 1949.
[30] J. R. Lewis, ''Psychometric evaluation of an after-scenario questionnaire for computer usability studies: The ASQ,'' *ACM SIGCHI Bull.*, vol. 23, no. 1, pp. 78–81, Jan. 1991., doi: 10.1145/122672.122692.
[31] J. Brooke, ''SUS—A quick and dirty usability scale,'' in *Usability Evaluation in Industry*, vol. 189, no. 194. London, U.K.: Taylor & Francis, 1996, pp. 4–7.
[32] J. Light, P. Lindsay, L. Siegel, and P. Parnes, ''The effects of message encoding techniques on recall by literate adults using AAC systems,'' *Augmentative Alternative Commun.*, vol. 6, no. 3, pp. 184–201, 1990.
[33] A. Bangor, P. Kortum, and J. Miller, ''Determining what individual SUS scores mean: Adding an adjective rating scale,'' *J. Usability Stud.*, vol. 4, no. 3, pp. 114–123, May 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=2835587.2835589
[34] J. Sauro, *A Practical Guide to the System Usability Scale: Background, Benchmarks & Best Practices Paperback*. Denver, CO, USA: Measuring Usability LLC, 2011.
[35] P. Gregor, A. Dickinson, A. Macaffer, and P. Andreasen, ''SeeWord—A personal word processing environment for dyslexic computer users,'' *Brit. J. Educ. Technol.*, vol. 34, no. 3, pp. 341–355, 2003. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-8535.00331
[36] Y. Yan, R. Rosales, G. Fung, and J. G. Dy, ''Active learning from crowds,'' in *Proc. 28th Int. Conf. Int. Conf. Mach. Learn.* Madison, WI, USA: Omnipress, 2011, pp. 1161–1168.
[37] B. Settles, ''Active learning literature survey,'' Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, WI, USA, Tech. Rep. 1648, 2009.
[38] R. Venkatesan and M. J. Er, ''A novel progressive learning technique for multi-class classification,'' *Neurocomputing*, vol. 207, pp. 310–321, Sep. 2016.

ALESSIO BELLINO received the Ph.D. degree in computer science from the University of Milano-Bicocca. He is currently an Assistant Professor with the School of Informatics and Telecommunications, Universidad Diego Portales, Chile. His research interests include interaction design, human–computer interaction, and ubiquitous computing.

VALERIA HERSKOVIC received the Engineering and Ph.D. degrees in computer science from the Universidad de Chile, Santiago, Chile. She is currently an Associate Professor with the Department of Computer Science, Pontificia Universidad Católica de Chile. She is also the Co-Founder of Chilewic, an ACM celebration of women in computing in Chile. Her research interests include human–computer interaction, ubiquitous and pervasive computing, and e-health.