User Performance and Experience in Camera-Based Scanning

Ruiqi Kong
HCI-E MSc Final Project Report 2019
UCL Interaction Centre, University College London
Supervisor: Enrico Costanza

ABSTRACT

Camera-based scanning applications are often used to access more information in public places, and the design of such Apps may influence both user performance and user experience. Camera-based scanning is a pointing and clicking task, while it differs from scanning using other devices. It involves a trade-off between image resolution and lag, which might potentially influence usability and user experience. This study focuses on the influence brought by delay and clicking method. Delay might be related to high image resolution; the clicking method can either be explicit click or auto click. A laboratory experiment was conducted to measure human performance in the form of throughput and error rate in the process of scanning. Subjective impressions were also collected via questionnaire from participants. Results showed a significant effect of a one-second latency on throughput, and a higher error rate with auto scan. Resolution did not influence the recognition of code reading system, and had little influence on user experience. Users showed preference to auto scan over explicit scan, while explicit scan was considered to be useful in specific situations. Three implications are proposed based on the quantitative and qualitative data, calling for lower resolution of viewfinder, more uses of explicit click in certain situations, and more consideration about the sense of control during the scan process. The study might help enhance user experience of camera-based mobile scanning applications.

Author Keywords

Camera-based scanning; resolution and lag; clicking methods; mobile devices; mobile HCI; experiment.

ACM Classification Keywords

H.5.m. *Information interfaces and presentation* (e.g., HCI): Miscellaneous

MSc Contribution Type

Empirical

MSC HCI-E FINAL PROJECT REPORT

Project report submitted in part fulfilment of the requirements for the degree of Master of Science (Human-Computer Interaction with Ergonomics) in the Faculty of Brain Sciences, University College London, 2019.

1. INTRODUCTION

Camera-based scanning applications are widely used in daily life. The scanning function enables people to access various mobile services and information about the target [22,32]. Users often find the camera-based scan technology targeted at visual tags in mobile phones is easy to use [32], and the adoption of such technology has gained global popularity, such as QR (quick response) code [11]. Scannable visual markers are applied in various scenarios: customers can compare the price of Amazon products with QR or barcode scanner applications [28]; museums and exhibitions provide scannable materials for visitors to enhance their experience [1,25,29]; restaurants integrate scannable visual markers into the eating experience to encourage interaction [19].

While early non-smart phones require a manual click to confirm a scan [32], nowadays the scanning function is mostly realised in an automatic way. A result will pop out as soon as the system successfully recognises a target and receives results from the backend database. The recognition process might take hundreds of milliseconds or a few seconds, with little visual feedback during the time. If the lag is longer than a certain length of time, it might lead to a sense of uncertainty among users. The mode of explicit click, on the contrary, might bring a sense of control through a manual click. However, explicit click is only possible when the system latency is relatively low, since this mode requires real-time feedback on the viewfinder. The latency can be attributed to the performance of devices, the complexity of processing algorithm, and the quality of network if remote transmission is involved [18,32].

Since it is impossible to completely eliminate latency, developers try to apply different methods to control the lag in an acceptable range. Image resolution is one factor that influences the processing time of the picture captured by camera, and thus the trade-off between high resolution and low latency becomes interesting when the computing power of device is limited. Higher resolution may raise the probability of a successful recognition, but increases latency; lower resolution means less data to be processed, and hence decreased processing time and latency, but it might incur a failed scan in poor light conditions. At the same time, it remains unclear that when the latency is relatively low, whether explicit click with a button could outperform auto click from a comprehensive view, taking possible visual feedback and a sense of control into consideration. This paper focuses on the trade-off of latency and resolution, as well as the comparison of two clicking methods, investigating user performance and experience in the scenario of scanning visual markers using camera phones.

Camera-based scanning is a kind of target acquisition, which is a classic topic in Human-Computer Interaction. Literature has investigated the impact of lag in various conditions, and evidence shows its negative influence on human performance [14,18,23,34]. It is recommended that lag should be cut down to less than 0.1 second so that users feel the system gives real-time feedback, and 1.0 second is the limit that users feel their thoughts are not interrupted [2,20]. With the restriction of hardware performance and network quality, how to balance the improvement of speed and the sacrifice of factors related to system performance, such as image resolution, becomes a challenging problem. Few papers explore the relation between resolution and latency in the scenario of camera-based code scanning, which becomes one main concern of this study.

Many interaction techniques based on camera phones have been evaluated in terms of human performance and experience. Factors such as target width and distance, display size, and jitter of the cursor on screen have been studied [10,24,33]. A modified model of Fitts' Law was proposed and validated in real world through a camera-based interaction technique named "Magic Lenses" [26,27]. In a recent study, visual feedback is suggested to be favourable to users in a computer vision processing system [12]. Despite the universal appearance of explicit click in early non-smart phones and the positive effect of visual feedback, the comparison of two clicking methods in code scanning applications has not been conducted.

Inspired by the work conducted by Rohs and Oulasvirta [26,27], this study evaluates user performance and experience in a target acquisition task using a camera-based scan application. The scan time and the scanned targets are recorded for performance analysis; data relevant to user experience is gathered through an online questionnaire accomplished after the target acquisition tasks; user behaviours are observed across the tasks. The trade-off of latency and resolution is discussed in this paper, based on participants' performance in the scan tasks and their subjective impressions explained in questionnaire. Two clicking methods are also compared. The results provide evidence for making decisions about the priority of speed and resolution in designing scan applications, as well as the choice of clicking methods. Users might benefit from a better design of code scanning applications that meet their needs in different scenarios.

2. RELATED WORK

2.1 Influence of Lag on Human Performance and Effort to Reduce It

Fitts' law is a model of human movement, which is widely applied in HCI to evaluate devices or interaction techniques for pointing. Its application scenario is restricted to "rapid, aimed movements" [17]. The model can be applied to measure human performance in the form of throughput, or predict movement time of subjects as a prediction model. Since one dimension of this paper is human performance, Fitts' Law is applied to evaluate subjects' performance of scan tasks.

The measurement involves *index of difficulty* (*ID*) and *movement time* (*MT*). *ID* is calculated as $\log_2(\frac{A}{W}+1)$, where *A* refers to the distance between the original position of the pointing device and the target, and *W* is the width of the target. *MT* is the movement time to complete a motor task. According to Fitts' Law, the ratio of *ID* and *MT* stays relatively constant across a range of *ID* settings, and the ratio is defined as *throughput*, which represents "human rate of information processing" in bits per second [17]. A larger throughput means better human performance of using the pointing device to perform target selection tasks in a specific setting. At the same time, *error rate* in participants' behaviour is considered an important measure of human performance by some researchers, although it is usually relatively low [17].

Lag is a detrimental factor to subjects' performance in target pointing experiment. MacKenzie and Ware modified the classic model of Fitts' Law based on an experiment with four lag conditions. They proposed that lag has a multiplicative effect on Fitts' index of difficulty [18]. Lag is common in computer systems, and it may come from input device, software or output device if network is not involved. If the computation load is heavy, the length of lag might exceed 100ms, which is the perception threshold for human [2,20]. The existence of lag often has negative effects on human performance and experience. It is reported that even relatively small lags could dramatically decrease human performance, if the targets to be acquired are small [34]. Compared with high-quality graphics, a short latency is more favourable to users in Virtual Reality (VR) applications [23].

System latency in the applications that require a high refresh rate usually contains two types, temporal delay and reduced display update rates. Although both of them might influence user performance and user experience, users tend to be more sensitive to temporal delay [14]. Therefore, it is recommended that emphasis should be put on the reduction of temporal delay [14,18,34]. Since lag derived from input or output device is often determined by the hardware itself, developers tend to cut down delay through improvement of software design.

However, it is sometimes difficult to reduce lag in applications that require highly complicated computation. Augmented reality (AR) applications are such systems that often have a perceivable delay [16]. Similar to AR which asks for recognition of specific markers, camera-based code scanning also involves image processing and decoding. Aside from the complexity of algorithms to recognise the codes, the resolution of images taken by the mobile camera also has a direct impact on the processing time. With the increasing quality of photos taken by mobile phones, the processing load for scanning applications increases.

Previous study shows a shorter temporal delay is preferred than better quality of graphics or higher update rate of display in both VR and AR application scenarios [14,18], while a camera-based scanning scenario has not been investigated. One aim of this study is to figure out whether it is necessary to keep the resolution of images high in the scenario of scanning with camera phones. If resolution is important to users, a range of image resolution could be reported where both the rate of successful recognition and the length of delay are within an acceptable range. If it turns out resolution is irrelevant to either user experience or the probability of successful scan, the resolution of scanning applications could be set to be the lowest level.

2.2 Interaction Techniques in Camera-Based Target Acquisition

Target acquisition is a common operation in computer-human interaction [9]. As discussed previously, Fitts' Law is a classic model to predict movement time of target acquisition or pointing gestures. The model is applicable to one-, two- and three-dimensional movements [8,17,21], and it is widely applied to evaluate interaction techniques of target acquisition. Rohs and Oulasvirta proposed a two-parts Fitts' Law model to fit camera-based selections [26]. In camera-based target acquisition, the movement of the screen has two stages: physical and virtual. People could only see the target in screen in the virtual pointing stage, while the real target is directly visible in the physical pointing stage. The model was also validated in a real-world environment, where a "markerless recognition of objects" was realised through AR [27].

This modified model is often used in the evaluation of an interaction technique named "peephole", which is a popular technique to navigate large information spaces [24]. Despite various methods to implement the interaction technique [10], the usage generally stays the same. Users hold a presenter device and move it in front of a public display. Information will appear on the public display screen, and the location of the area matches the position of the handheld device. Moving the device around is similar to navigate the complete display through a peephole. A cursor is shown in the centre of the peephole, helping users locate the target.

To evaluate the influence of factors such as peephole size [24] and target size [10] on human performance, target acquisition tasks were designed. Participants usually have to click a button on the handheld device to confirm a target, which is similar to the operation of scanning a target in a non-smart phone. Although the action of marker recognition and code scanning is similar to "peephole", the design of scanning applications applies auto scan by default. Auto scan provides result fast, which is usually a favourable trait. However, when more than one scannable target is available, auto scan might make mistakes easily. By contrast, explicit click gives users more control, potentially resulting in a stable performance. The assumption leads to the other concern of this study: whether clicking with a button is better than clicking automatically.

2.3 Evaluation of Selection Technique and Design Feature

A typical method to evaluate human performance in pointing tasks is cyclical multi-direction pointing task paradigm [30]. Nine targets are fixed on a plane in the form of circle, and participants use a specific pointing device to select the targets in a pre-set sequence. It is preferred over the one-directional task because it is closer to a real-world situation [26]. This task paradigm does not meet the expectation of this experiment, however, since the direct transition from one target to another cannot produce a situation where users may scan a marker accidentally when passing by. The existence of such condition can test whether auto scan is error-prone compared with manual scan.

An untraditional evaluation method was proposed in an experiment to validate a modified Fitts' Law Model in real world, which is a reciprocal pointing task [27]. Participants have to "move back and forth" between a pair of targets, of which the properties such as size and shape vary [27]. The design of the task is to meet the research question that how participants change their behaviour in this situation. The design of reciprocal pointing task in a real-world setting inspired the task design of this experiment, where participants scan targets in pairs. The first target within a pair is fixed, functioning as a starting marker. The existence of starting marker is similar to people's behaviour of putting their mobile phone to their pocket.

Based on the two points gained from previous paper, the task design is established. Participants have to scan three pairs of targets for a certain number of times, and the three pairs share the same starting marker. The layout of scannable markers deliberately creates the situation that users might scan a wrong target if no strategy is applied in the auto click mode.

When calculating throughput, the two-stage modified Fitts' Law model proposed by Rohs and Oulasvirta [26] will not be applied, although this experiment involves camera-based selections. The "physical pointing stage" becomes so short that can be ignored, since the target markers are close to each other, which makes it easy for users to locate the target. It can be claimed that scans happen in the virtual pointing stage, therefore, the original single-part Fitts' Law Model fits this experiment most.

Throughput and *error rates* are two measurements of human performance in this experiment. To a certain degree, the two measures represent "speed" and "accuracy" of users in performing scanning tasks. *Throughput* means human capacity of information processing [17], which is similar to the concept of "speed". *Error rate* shows how accurate the user scans the expected target, as a wrong scan is defined as a scan of a target that is not assigned.

Aside from human performance, studies that focus on interaction techniques may also evaluate the ease of use [14]. When it comes to the effect of specific design feature in a system, subjective impressions of different conditions are often gathered to understand users' opinions [12]. Since this study has one concern about the experience of two modes of scan, a questionnaire is designed to collect participants' preferences.

3. EXPERIMENT DESIGN

3.1 Design

The experiment is a within-subject design of two independent variables. One independent variable is *resolution* (R), along with simulated latency. It has three levels: Full resolution (F) with a latency of 1000 milliseconds (ms), half resolution (H) with a latency of 500 ms, and quarter resolution (Q)with a latency of 0 ms. Resolution determines how clear the image in the viewfinder looks. The latency is the time before a visual feedback shows up in the viewfinder, which indicates a successful recognition of the scannable target.

The maximum latency of 1000 ms is chosen because it is the limit that people feel their thoughts are not interrupted [2,20]. The latency of 0 ms is chosen to represent the shortest overall latency. From tests, the researcher found the actual processing time for recognising a target in low resolution is so short that cannot be perceived, and therefore the lowest latency was decided to be 0 ms. Also, since visual feedback will only appear when the system has recognised the marker, actual processing time has already been included in the overall latency, in which sense the simulated latency becomes an extra latency aside from the actual processing time.

The other independent variable is *clicking method* (CM), including two modes: auto scan (A) and explicit click (E). Auto scan means a scan happens automatically when the viewfinder is pointed to a target; explicit click means users have to click a button to trigger a scan. In the mode of explicit click, visual feedback will appear on the viewfinder in a relatively low latency that is close to a real-time setting, and the result will only pop up when the user clicks the button. In the mode of auto scan, a result will automatically pop up when the system has recognised the marker.

The design is factorised into two separate blocks, studying one independent variable in each block. To mitigate possible learning effects, the conditions for each independent variable are fully counterbalanced. Block 1 studies resolution, therefore six combinations of the three conditions are applied with explicit-click mode. Block 2 studies clicking method, and two combinations of the two conditions are applied with quarter resolution (latency of 0 ms). Table 1 shows the detailed combinations.

Block	Resolution			Clicking method	
1	F	H	Q	E	
	F	Q	H	E	
	H	F	Q	E	
	H	Q	F	E	
	Q	F	H	E	
	Q	H	F	E	
2	Q			E	A
	Q			A	E

Table 1. Combinations of conditions in two blocks.

To evaluate human performance, *throughput* and *error rate* are calculated. *Throughput* is calculated by the ratio of *ID* and *MT*. Three markers yield different *ID*s due to the varied distance between them and a starting marker. *MT* is defined as the time from successfully scanning the starting marker to scanning the target marker. *Effective Movement Time (EMT)* is further defined to remove the fake delay in the scan process. Specifically, for trials under the condition of full resolution, EMT is $MT - 1000$, and for trials of half resolution, *EMT* is $MT - 500$. *Error rate* is calculated as the number of wrong scans and failed scans divided by the total number of scanning operation. User experience about the resolution-latency conditions and clicking methods is investigated through an online questionnaire after finishing the pointing tasks.

Aside from human performance and user experience, human behaviour is also investigated. How close the user holds the camera phone to the marker is studied, in the form of marker size within the screen. Size is calculated as $width^2 + height^2$, following the common method of describing screen sizes using the length of the diagonal It is not rooted due to a consideration of decreasing computation complexity. How users hold the phone is also observed, such as whether they use both hands or a single hand.

3.1.1 Hypotheses

Since a latency of one second might influence people's thoughts, it might also influence their scan behaviour significantly. Auto scan leaves out the operation of clicking, which might yield better performance than explicit click. In particular, the completion time in auto scan might be shorter than explicit click, while a higher error rate is expected. In terms of user experience, people might find latency of 1000 ms as well as quarter resolution is uncomfortable in the process of scanning. They might prefer auto scan to explicit click, because the former is widely applied in daily life.

Hypothesis 1: Participants' performance under the condition of Full resolution with a latency of 1000 ms is significantly worse than that under other conditions in Block 1.

Hypothesis 2: Participants' performance under the condition of Auto scan is significantly better than that under the condition of Explicit click, while auto scan yields a higher error rate.

Hypothesis 3: Participants feel uncomfortable when the latency reaches 1000 ms or when the resolution is very low (quarter).

Hypothesis 4: Participants prefer auto scan to explicit click.

3.2 Materials

The experiment was conducted using a smart phone (SM-G9350, Samsung Galaxy S7 Edge) with Android version 8.0.0. Two mobile applications were developed for two blocks respectively in Qt Creator. Four visual markers from d-touch system [3] were applied in this study. The markers were printed on papers and stuck on a white board to be fixed on the wall. Each marker is 4.0 cm wide and 4.5 cm high. One of the markers was set to be the starting marker (SM), while the other three were numbered 1, 2 and 3. The markers were distributed in an upside-down "L" shape, with a 10-centimetre distance between each other. The layout is demonstrated in Figure 1. The *IDs* of three markers are shown in Table 2.

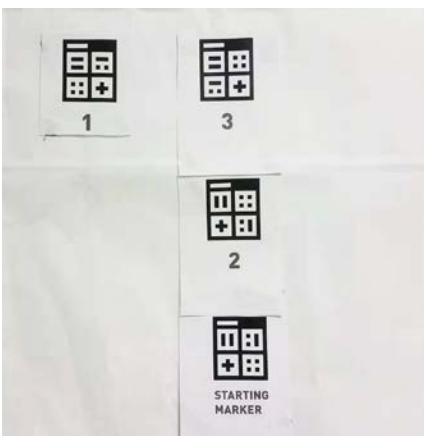

Figure 1. Layout of scannable targets.

Marker No.	Distance to the SM (cm)	*ID*
1	$10\sqrt{5}$	2.72032
2	10	1.80735
3	20	2.58496

Table 2. *ID* of three markers.

3.2.1 Application

The two mobile applications were developed in Qt Creator, mainly written in QML. The design of the App follows the idea of Model-View-Control (MVC) structure [5]. Java Script deals with the main logic part, and QML widgets are applied to implement the Model and View part. The programme depends on two libraries written in C++: the d-touch library [3] to read scannable markers; a File I/O library to read and write files in physical devices, which is also written by the author of this report. The structure following an MVC framework is illustrated in Figure 2. Interactions between users and the application are presented in Figure 3, and are explained in the following paragraphs.

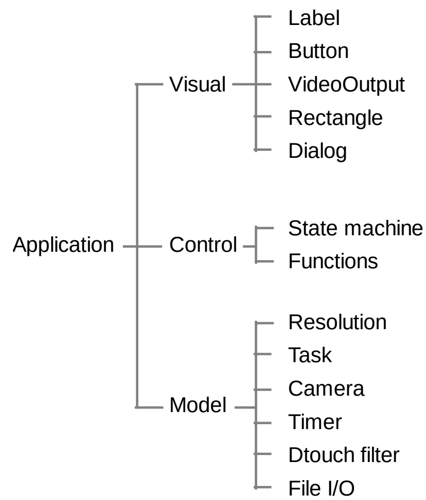

Figure 2. Structure of the application.

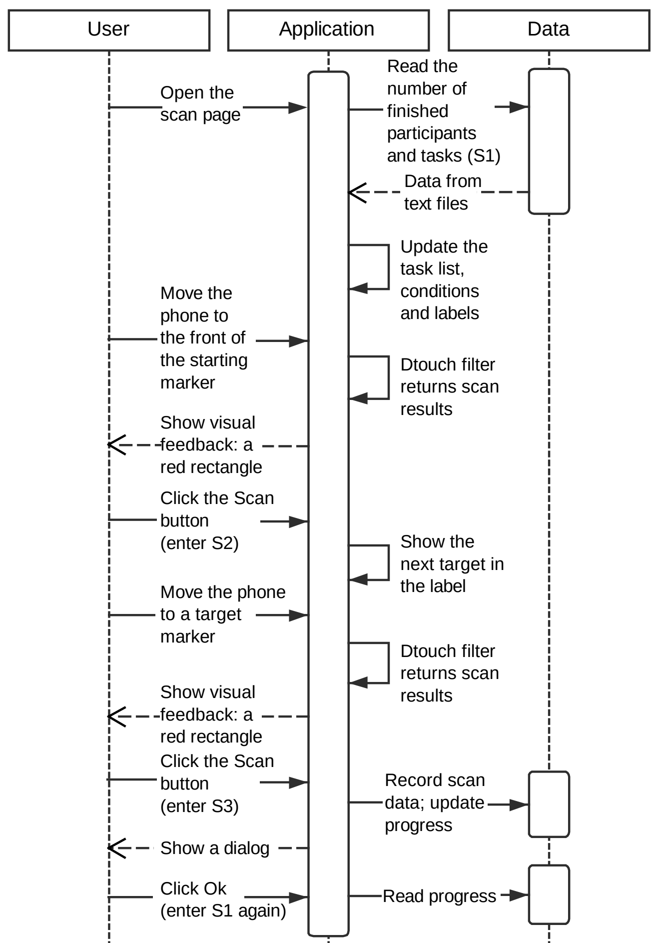

Figure 3. UML sequence diagram under the condition of explicit scan.

The logic of the App is realised through a state machine framework [31], which is composed of three states:

State 1: Prepare for next scan

State 2: Ready to scan the starting marker

State 3: Ready to scan the target marker

In the first state, the programme reads the number of participants that have completed all the tasks and the number of finished tasks for the current participant. The data is stored in a text file in the physical device. The programme then updates the task list if necessary. For Block1, a task in the task list includes the condition of *resolution* and the target number; for Block 2, a task includes the condition of *clicking method* and the target number. In each condition, the three markers are set to be scanned for eight times respectively.

The sequence of targets is randomised, while the sequence within one condition for each participant stays the same. For example, in the full *resolution* condition for the first participant, the sequence of target markers is "233122133222113331121321", and it is the same for the following half and quarter *resolution* conditions. The third participant is expected to scan the targets in the same sequence, while the sequence of resolution conditions is full-quarter-half.

There are 3 conditions × 3 markers ×8 times × 6 combinations = 432 tasks in the list of Block 1, and 2 conditions × 3 markers ×8 times × 2 combinations = 96 tasks in the list of Block 2. For each participant, 3 conditions × 3 markers ×8 times = 72 tasks are fetched from the task list in Block 1, and every two participants fetch a new segment from the task list. In Block 2, one participant is assigned 2 conditions × 3 markers ×8 times = 48 tasks.

Condition	Resolution	Latency
Full	640×480	0 ms
Half	320×240	500 ms
Quarter	176×144	1000 ms

Table 3. Resolution and latency settings in Block 1.

After reading the progress, the condition will be updated according to the current task. In Block 1, the resolution and latency before visual feedback will be updated if the current setting differs from the setting of the task. There are three levels of resolution with latency, which are specified in Table 3. The choice of resolution is based on supported resolution options of the device. 176×144 is the lowest resolution supported by the device, and 640×480 brings perceivable latency in the viewfinder. In Block 2, the scan mode will be switched if the task brings a different value of "clicking method". In Explicit-click mode, a button is fixed in the bottom of the screen, and it can only be clicked when a visual feedback appears on the target. In Auto-scan mode, there is no button on the screen, and a successful scan will be recorded by the programme once the d-touch system recognises a marker.

When the "Scan" button is clicked (Block 1) or a new marker is recognised (Block 2), the state is transited into State 2. In this state, it is expected that the starting marker would be scanned. If the starting marker is actually scanned, the programme will enter State 3; otherwise it will stay in State 2. If a target marker is scanned in State 2, a dialogue will appear, indicating the user that he or she should scan the starting marker (see Figure 4).

Figure 4. Error Dialogue.

State 3 waits for the user to scan a target marker. After a target marker is scanned, a dialogue will appear to guide users to scan back to the starting marker. Once the dialogue is accepted by the user, the programme jumps to State 1. If the starting marker is scanned in this state, no response will be triggered.

Scan results are dealt by one function. If the result is a starting marker, the function will update a label that shows the next target number. If the result is not a starting marker, normally data about the last scan will be recorded and written into a local file. At the same time, the local file to record finished number of participants and tasks will be updated.

One piece of scan record includes 11 columns, which are specified in Table 4. *Expected/Actual target number* is the coding of markers defined by the d-touch system. *Completion time* is the time between a starting marker is scanned and a target marker is scanned. In Explicit-click mode, it is the time between the two times that "scan" button is clicked; in auto-scan mode, it is the two times that a new marker is recognised by the d-touch system. The format of time is defined by the java script function applied in this programme. The width and height of the marker when it is scanned is also recorded, and its value is the ratio of the marker size to the viewfinder.

Column	Value
Participant number	1-12
Block number	0, 1

Column	Value
Resolution	Full, half, quarter
Clicking method	Explicit, auto
Expected target number	2341, 2413, 2431
Actual scanned target number	2341, 2413, 2431
Starting marker scan time	the number of milliseconds since 1 January 1970 00:00:00
Target scan time	the number of milliseconds since 1 January 1970 00:00:00
Completion time	*Target scan time – starting marker scan time*
Marker width	0.0-1.0
Marker height	0.0-1.0

Table 4. Format of a piece of scan record.

The view of the App mainly consists of a viewfinder, a text label in the centre, and a button in the bottom. The resolution of viewfinder changes according to the current task in Block 1, while it stays constant at 176×144 in Block 2. A red rectangle will appear as visual feedback if there is one and only one marker within the viewfinder (see Figure 5 left). The central text label in red indicates next target once the starting marker is scanned (see Figure 5 right). The overlay design of the dynamic label is suggested by Toye et al.[32], which allows users to notice the change easily. When the next target remains unknown, the label appears as three points. The scan button has a border in yellow, and the border width becomes wider when it is clicked, to indicate an effective click. The button is disabled when no marker could be recognised from the viewfinder. In auto-scan mode, there is no button on the screen (see Figure 6).

Figure 5. Visual feedback and text labels.

Figure 6. Auto scan and explicit scan.

3.2.2 Questionnaire

The questionnaire includes three closed questions and three open questions, and is published on Microsoft Forms. The three closed questions are centred around participants' attitudes towards resolution, lag and clicking methods; open questions are designed to understand participants' subjective impressions of the relationship between resolution and lag, as well as auto-scan and explicit-click. The order of questions follows the design of the experiment, from the topic of resolution and lag to the topic of clicking methods. The content is specified in Table 5.

	Question	Options
Q1	*Which level of resolution becomes unbearable in terms of lag before the red rectangle appears?*	*1. Full resolution (lag: 1000ms)* *2. Half resolution (lag:500ms)* *3. Quarter resolution (lag: 0s)* *4. Does not matter*
Q2	*Which level of resolution becomes unbearable in terms of image quality shown on the screen?*	*1. Full resolution (640×480)* *2. Half resolution (320×240)* *3. Quarter resolution (176×144)* *4. Does not matter*

	Question	Options
Q3	*On what occasions do you think the image resolution is more important than the lag when scanning markers?*	*(Open question)*
Q4	*Which clicking method do you prefer?*	*1. Explicit click (with button)* *2. Auto click (without button)* *3. Hard to say*
Q5	*What positive or negative impression do you have on the two kinds of clicking methods?*	*(open question)*
Q6	*On what occasions do you think the two clicking methods are respectively suitable for?*	*(open question)*

Table 5. The content of the online questionnaire.

3.3 Participants

Twelve participants (11 female, 1 male, age between 22 and 27) took part in this experiment, and £10 was transferred to each of them as a reward. Participants are recruited through online advertisements spread within classmates and friends, and consists of graduate students and researchers from universities in England. Participants are all right-handed and have experience in using mobile scan applications.

3.4 Procedure

Consent forms were collected after the researcher explained the research to the participants. The usage of the scan application and the layout of the markers were then introduced to participants. Afterwards, participants were guided to stand in front of the markers and started to practice. They were told to practice until they could explain the correct process correctly. After they were ready, the researcher switched the mode from "practice" to "experiment" through a hidden button in the application, and participants started to scan the markers. The only requirement was to scan as fast as possible during the scan of each pair of targets. Since the application records the time between scanning the starting marker and scanning the target marker, which is a complete scan of a pair of targets, participants were allowed to take a short break between the pairs.

The scan experiment includes two blocks, with a focus on *resolution/lag* and *clicking methods* respectively. In Block 1, each participant finished scans for three conditions. Under each condition, the three pairs of markers were scanned for eight times in a shuffled order which was set in the application. The participant clicked the button to scan the starting marker first, and then attempted to scan the target marker of which the number matched the one displayed on the screen. Afterwards, a dialogue reminded the participant to scan the starting marker as a new scan. When the participants finished all the tasks in three conditions, the application quitted automatically. In Block 2, participants had to finish the tasks of two conditions. Block 1 took approximately eight minutes to accomplish, while Block 2 took around 4 minutes. During the experiment, if they felt the application did not function normally, they would report to the researcher to check.

After the two blocks, the participants were asked to complete an online questionnaire using a laptop. They could ask the researcher questions if they felt unsure about the meaning of the questions. The survey took about 10 minutes to complete.

4. RESULTS

4.1 Human Performance

The experiment yielded 12 subjects× 3 conditions × 3 markers × 8 selections = 864 data points in Block 1 and 12×2×3×8=576 data points in Block 2. Thirty outliers in Block 1 and thirty-two outliers in Block 2 were removed, which were more than 2 standard deviation from the mean.

Repeated measures ANOVA tests were run on the results in Block 1. Human performance tends to be better when the lag is shorter and resolution is lower. As shown in Figure 7, with the decrease of *resolution* and *lag*, *throughput* increases. A significant main effect of *Resolution/lag* on *throughput* is found ($F(2,22) = 18.37$, $p < .001$). A post hoc was further run, and there is a significant difference between full resolution and half resolution/quarter resolution, while the variance between half resolution and quarter resolution is not significant. The details are shown in Table 6. No significant difference is found in terms of *error rate* ($F(2,22) = 1.000$, $p = 0.384$), and the range is illustrated in Figure 8.

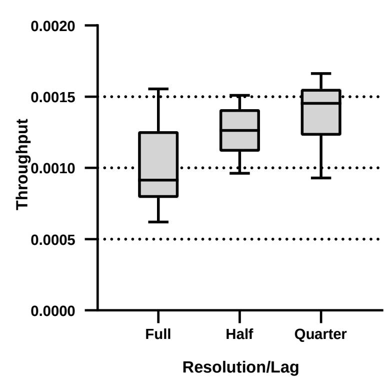

Figure 7. Boxplot showing throughput for the three levels of resolution/lag. Bars indicate range, lines within the box indicate mean.

		Mean Difference	SE	t	p_{bonf}
Full	Half	-2.219e -4	6.106e -5	-3.634	0.012
	Quarter	-3.408e -4	5.731e -5	-5.947	< .001
Half	Quarter	-1.189e -4	5.253e -5	-2.264	0.134

Table 6. Post hoc results for throughput in Block 1.

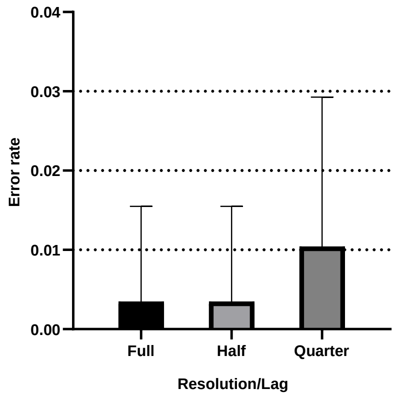

Figure 8. Bar chart of error rate for the three levels of resolution/lag. Error bar stands for standard deviation (SD).

Paired-samples *t*-tests were used to analyse the quantitative data collected from Block 2. There is no significant effect of *clicking methods* on *throughput* ($t(11) = 1.932$, $p = 0.079$), while the *error rate* under the condition of auto scan is significantly higher than that of explicit click ($t(11) = 5.162$, $p < .001$). Mean and standard deviation are displayed in Table 7, and Figure 9 shows the difference between two conditions.

Dependent Variable	Condition	Mean	Standard Deviation
Throughput	Auto	0.003	0.003
	Explicit	0.001	2.222e -4
Error rate	Auto	0.142	0.091
	Explicit	0.010	0.019

Table 7. M and SD of throughput and error rate in Block 2.

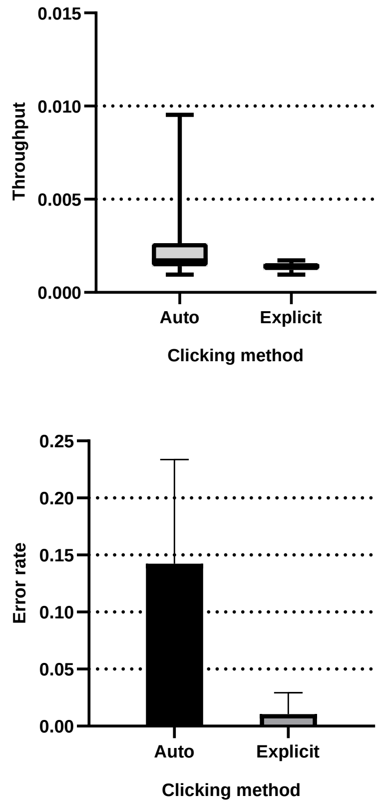

Figure 9. Plot of throughput and error rate for the two conditions of clicking methods.

4.2 Subjective Impressions

Most participants (8 out of 12) think the lag in the condition of full resolution becomes unbearable, and image quality does not matter in the process of scan. Three participants chose that lag does not matter during scan and one chose half resolution, which was later confirmed that she thought the second condition she met was half resolution, but actually was full resolution.

In terms of occasions where resolution might be more important than latency, most subjects did not have any idea. Two participants answered that “accuracy” might be a factor, and one answered “when the QR code is very small”.

Eight participants prefer auto click and the other four prefer to clicking with a button. Explicit click seems not have obvious advantages, as only one participant mentions “*it can be used to scan several items continuously and clearly*”. It is also mentioned by one subject that she “*sometimes forget to press the button to scan*”. On the other hand, auto click is considered to be fast and convenient, and it can also “*free one of the hands*”. Some disadvantages are also mentioned by participants, which are centred around a sense of uncertainty and confusion:

P4: *“When I scan different items in a short period it made me confused.”*

P8: *"The auto click method makes me feel unsure about whether it is working well."*

P12: *"The auto mode is a little bit too fast to show the results."*

Participants listed the situations that they felt suitable for the two kinds of clicking methods. The situations can be generally categorised into three types: material, task, and scenario. From the perspective of materials, explicit click is commented suitable when there are multiple markers in a limited space, while auto scan is considered suitable when only one marker is available in front of the camera.

P1: *"(Explicit) when many codes on one page"*

P3: *"(Explicit) there are 2+ QR codes near each other"*

P1: *"(Auto) when only one large size code on the page"*

P3: *"(Auto) … there's only one targeted QR code. Or other unwanted QR codes might be mistakenly scanned, which would be annoying"*

From the perspective of tasks, explicit scan is considered suitable when the scan task needs to be conducted repetitively. Auto scan is deemed more efficient when a faster scan is preferred.

P4: *"(Explicit) have to scan several items continuously"*

P6: *"(Explicit) work that requires a large number of repetitive actions"*

P3: *"(Auto) in the rush of scanning QR code…"*

P8: *"(Auto) when the task has really high time pressure…"*

From the perspective of scenarios, a consensus seems to be reached that explicit click is necessary when the scan activity involves payment and identification. The operation of clicking gives users a sense of control. Auto click is referred to be usable in common daily scenarios.

P3: *"(Explicit) users won't make mistake by clicking the button as a way of checking and confirming it"*

P5: *"(Explicit) for specific accounts with financial transactions"*

P8: *"(Explicit) when the feedback of the scanning may be not clear…"*

P9: *"(Explicit) large payment and identity confirmation"*

P10: *"(Explicit) pay or receive money"*

P12: *"(Explicit) may be suitable for activities like payment"*

P5: *"(Auto) for specific accounts with social aim"*

P9: *"(Auto) information collecting in museums"*

P11: *"(Auto) normal use"*

The availability of hands is also mentioned in both sides. Auto scan is still usable when there is only one hand available while explicit scan requires the user is able to touch the screen.

P8: *"(Explicit) …and when it is convenient for the user to touch the screen"*

P8: *"(Auto) when… it is inconvenient for me to touch the screen when holding"*

P11: *"(Auto) for the disabled perhaps"*

4.3 Observation of Behaviour

The size of targets shown in the viewfinder was also recorded, which reflects how close the participant moved the phone to the markers. For Block 1, a repeated measures ANOVA test was run on the recorded sizes. Resolution and lag do not have a significant effect on the size of markers ($F(2,22) = 1.700$, $p = 0.206$), as illustrated in Figure 8. For Block 2, a paired samples t-test was applied. There is a significant main effect of clicking method on the marker size ($t(11) = 3.342$, $p = 0.007$), and auto scan (M = 1.232, SD = 0.398) brings bigger size than explicit click (M = 0.800, SD = 0.261). Figure 10 shows the differences.

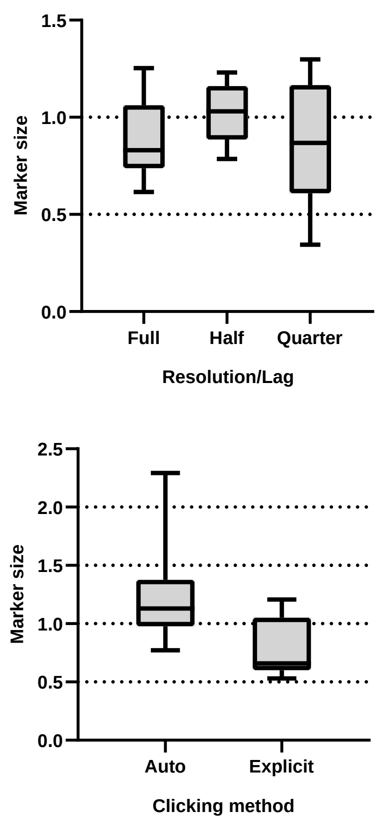

Figure 10. Resolution/lag and size (top); Clicking method and size (bottom).

Regarding how users hold the phone, it is found that most participants held the phone with one hand first, and then changed to both hands after a few scans in the condition of explicit click. When using both hands, participants tended to hold the phone with left hand and click the button using right hand. Some participants scanned using both hands from the very beginning. One participant held the phone in a landscape view when the condition was auto scan.

Participants showed different strategies when scanning the marker 3 in the condition of auto scan. One major strategy is to avoid the marker 2 by following a curved trajectory; the other strategy is to move the phone very quickly over the marker 2. However, the second strategy often led to a wrong scan of the marker 2. One participant also scanned marker 1 in a curved way.

When the red rectangle did not appear, one participant touched the viewfinder to “help” the application to recognise the marker. One participant mentioned that the red rectangle changed its size sharply in an abnormal way when the resolution was low, which made her feel a bit strange.

After the scan tasks, many participants commented that they did not notice the change of resolution. Some participants said the auto scan is “too sensitive”. One participant mentioned auto scan makes him “feel a bit dizzy” due to the continuous jitter during the tasks.

5. DISCUSSION

5.1 Findings

A one-second latency in the condition of high resolution slows down users’ scan operations but does not influence the accuracy of scanning. Auto scan brings less accurate scan results than explicit click, while no significant difference of scan speed is found between the two clicking methods. It is also found that participants tend to move the phone closer to the marker in auto scan mode. From the questionnaire, it is revealed that users tend to pay little attention to resolution but have negative impressions on the one-second latency. Most participants prefer auto scan to explicit click, while some disadvantages of auto scan are also put forward. Despite their preference of auto click, participants list various situations that the two clicking methods are suitable for respectively.

The results partly support the hypotheses, but show a more complicated view than what was expected. Since human performance involves both *throughput* of scanning and *error rate*, neither of the two independent variables is related to a significantly better performance. The trade-off of “speed and accuracy” [35] seems to appear in the experiment, but the sacrifice of speed does not bring improved accuracy, neither does a higher error rate relate to significantly improved *throughput*.

Another trade-off of “speed and accuracy”, which is not of the conventional meaning, was thought to exist between resolution and lag, but no evidence is found to support it. The classic trade-off problem is discussed from the view of a person; people have to sacrifice accuracy for speed or sacrifice speed for accuracy in many kinds of tasks [35]. In the context of resolution and lag, a lower resolution might increase the possibility of a failed scan, which is a decrease of accuracy; since lower resolution means the frames that need to be processed are relatively smaller, the time needed to recognise a marker might decrease, which can be explained as faster speed.

Based on the results in this experiment, the decrease of resolution does not bring lower recognition rate of markers. Actually, with the lowest resolution supported by the mobile phone, the code reading system was able to recognise the marker quickly. It can be explained by the robust design of markers and code reading systems. From the perspective of designers, scannable markers are specially designed to be recognisable in poor conditions such as low resolution and poor light [3,36]. As a result, it can be assumed that low resolution can provide stable recognition results.

It is not surprising that users care more about latency than resolution when scanning codes. The aim of scanning decides the limited attention that users pay to the resolution of viewfinder. Users scan the code for further information, which means scanning itself is not the core concern. The action of scan only requires a successful recognition of a marker by the code reading system, rather than taking a high-resolution picture. As long as the system is able to decode the marker, users tend not to notice the viewfinder resolution. Although one participant mentioned the unstable size of the visual feedback in low resolution condition, which might be caused by the drawback of the decoding system, the problem can be deemed as a minor one. On the contrary, users appear less tolerant to a longer latency, which is in accordance with the findings in previous research that people put more emphasis on delay reduction [14,18].

As most scan functions applied in daily life are in auto scan mode, it is natural for users to prefer auto click to explicit click. However, the performance data collected in the experiment and the answers from the questionnaire also suggest there are reasonable usage scenarios for explicit click, and it does not harm human performance significantly in scanning tasks. One typical scenario is when multiple markers stay close to each other, which may lead to recognition of a wrong target easily in auto mode. A button enables users to control which target to scan.

From the data of marker size within the viewfinder, it is found users tend to move the phone closer to the target in auto mode. It might mean users lack a sense of control over the scan function, and moving closer helps add the sense of control. Explicit control, on the other hand, in itself gives users a sense of control. The visual feedback, the red rectangle appearing over the target again enhances the sense of control, emitting a clear signal that the marker has been recognised. Although the time it takes to recognise the target is the same for the two clicking methods, the form of which a result pops up is different. In auto scan, the result jumps out suddenly without a “signal”, which means it might happen immediately or after a few seconds. It completely depends on the system to show the result. Explicit click, however, hands over the power to users, letting users decide the time to show the result. The result will appear immediately after the user clicks the button, despite the fact that it actually takes longer time to return a result than auto click. The “extra” operation might assure users that they are in control of the scanning process.

The simple action of click might bring a sense of control to users, which is favoured on some occasions. Several participants mentioned financial transactions and identity authentication when asked what occasions they feel explicit click is suitable for. This reflects the sense of control that users can gain through the operation of click by intuition. In daily life, automatic scan of QR codes are widely applied in authentication processes, and schemes are carefully designed to ensure the security during communication [4,15]. If the scannable target is clear, the action of auto scan is also controlled by the user, as the user has to actively move the camera phone to the front of the marker. If there are multiple markers and the user has difficulty locating the only one that he or she would like to scan, an explicit way of clicking might be preferred.

Taking cases of attacks based on QR codes into consideration, explicit click can function as a shield and actually improve security of activities of scanning codes. According to Krombholz et al. [13], QR codes in public places might be replaced completely or partly by malicious attackers, which exposes users at risk. While users may hold concern about the trustworthiness of the scannable targets, a scan application using auto scan mode might recognise the marker directly without a buffer, as code reading systems are generally sensitive and efficient. To avoid being attacked easily, explicit click could be combined with design directions suggested by Krombholz et al., such as displaying the expected content on the screen, or pre-process the content [13].

If there is a series of scan tasks to be completed, just like this experiment, explicit click might also function as a point of separation between each scan task. The action of scanning divides the whole process into multiple “chunks”, which is a common term used in Psychology and Cognitive Science. Chunking has been investigated in various fields [7], and it is often an unconscious motor action. However, studies also reveal that it might help acquire skills effectively by intentionally dividing a sequence of actions into several chunks [6]. Scanning cannot be considered as a kind of skill acquisition, yet the chunking brought by explicit clicks might help users gain a sense of stability and a sense of control. As one participant said, the continuous movement of the phone in auto scan mode makes him feel “dizzy”. The negative experience might come from the minor yet persistent shake of the camera phone, and the lack of a “break” point enhances the feeling of instability.

5.2 Implications

Low resolution is adequate for code scanning. A relatively low-quality frame captured by the camera can be successfully decoded by the code reader system, which can be attributed to the robust design of such coding and decoding systems. The low-quality image displayed in the viewfinder would not bring negative experience to users, because users tend to ignore the viewfinder resolution and expect a shorter latency before the scan result pops out. Although the cut-down of resolution might not lead to a greatly shorter lag, it is recommended to apply available means to shorten the processing time of the scan process. In other words, it is unnecessary to apply high resolution settings for the viewfinder in a code scanning scenario.

Explicit click seems to be a better choice than auto click, especially when multiple targets are close to each other. The two clicking methods did not vary significantly in terms of *throughput*, while a lower *error rate* was found in the mode of explicit click. In other words, explicit click did not make the scanning process slow, which is contrary to participants’ impressions, and it caused few mistakes during scanning, especially when the targets were close to each other. When multiple scannable markers stay adjacent to each other, it becomes challenging to point a target in auto scan mode. In spite of various strategies that users could apply to avoid scanning other markers by accident, the probability of a wrong scan is rather high. On the other hand, although users might have the impression that auto click is faster than explicit click, which might not be true according to the experiment result, the potential risk of a wrong scan can make it less efficient than explicit scan.

A typical scenario where multiple scannable targets exist is in galleries. In a gallery that collects abundant artworks, the collections are often hang on the wall adjacent to each other. Users are able to access more information about a piece of artwork by using applications with a scanning function. Smartify is such an application that supports recognition of some works in National Gallery in London [37]. As it applies an auto scan mode, the application might negatively influence user experience due to wrong scans at times. Explicit click is more user-friendly in such scenarios. However, if users are not able to use their hands to touch the screen, explicit click becomes inconvenient compared with auto click.

The sense of control is a factor worth consideration when designing scan-related applications. A sense of control and security is preferred in financial and identification situations. If the system does not return a result immediately, users are likely to feel unsure and worried. Explicit click might provide a sense of control, although it does not actually influence the scan of the code reading system. In a scenario that the scannable marker might be modified maliciously, explicit click could increase controllability and thus improve security, if combined with other design features. Explicit click might also assure users by chunking the scan process into multiple small units, which might be helpful when there are a series of scan tasks to be accomplished.

5.3 Limitation

One limitation is that the correspondence between resolution and lag may be far from the reality. In the design of this experiment, a 640×480 resolution is matched with a 1-second lag, and the lag is shortened by half with a 320×240 resolution. Due to the high computation power of current smart devices, a one-second latency might not be found in the scanning process. Furthermore, the time spent on decoding the marker from the frame captured by the camera might compose a very small part of the whole latency, and thus the cutting down of resolution would not shorten the latency significantly. This makes the contribution of the first block of the experiment limited, considering the generally strong computation power of the devices today.

Furthermore, the range of resolution and lag in this experiment is relatively narrow, with only three options. It makes the analysis of the trade-off between resolution and lag limited. The gap between the settings of latency could be narrowed, for example, from 500 ms to 100 ms. A larger range with more combinations of resolution and lag could be considered.

The successful scans under the condition of the lowest resolution indicates a limitation that the tested markers are very robust. One of the reasons why the researcher was interested in the trade-off between resolution and lag is that a low resolution might incur a scan failure. However, in the experiment, no scan failure happened. This can be explained by the general robust performance of scannable markers, but there might exist markers that are not so robust as the tested ones. It would be interesting to investigate the trade-off when recognition of markers fails under the condition of low resolution.

Another limitation lies in the method to calculate index of difficulty, which is involved in the calculation of *throughput*. In the equation that $ID = \log_2(\frac{A}{W} + 1)$, A is the distance of movement, and W is the target width. Since the scan task is designed to start with the same starting marker, A of the three markers is calculated as the distance between each marker and the starting marker. W stays the same value due to the identical marker size. However, the movement distance of the camera phone does not necessarily equal the distance between markers. Although most participants tended to point the target with it being in the centre of the viewfinder, there were still minor differences in terms of movement distance due to the differences of holding posture and scanning habits. Some participants moved a minimum distance to scan the targets, while others attempted to let the marker appear in the centre of the viewfinder.

The task design of this experiment also has its limitation. The starting point of applying a non-conventional task design is to approach the setting of a real scenario, such as in a gallery. In a gallery, multiple scannable targets can be found on the same plane, and visitors are likely to scan continuously to access the introduction of this piece of artwork, which takes a short period of time to read. In the experiment, however, the continuous scanning does not include the short "reading" time. It makes the task rather unrealistic compared with the gallery setting, while the design does not follow a traditional cyclical task in the laboratory setting either. Therefore, the ecological validity of the results may not be very high.

6. CONCLUSION

This paper presents a study about human performance and user experience of using camera-based scanning applications. With a non-conventional design of scan tasks, it is found a relatively long latency strongly influences users' throughput capacity in a negative way, while resolution of viewfinder has little effect on users' performance or experience. Two modes of clicking methods for scan applications vary in their effects on the error rate of scanning, as more mistakes were made in auto mode. Users prefer auto scan over explicit click, but there are certain scenarios that explicit click is suitable for. A lack of senses of control under the condition of auto scan is also revealed through users' behaviours and subjective impressions.

Implications that can be seen from the results involve three perspectives. Regarding the physical setting of the viewfinder, a low image quality is adequate for code scanning, since scannable codes are intentionally designed to be robust against poor scan conditions. For the scenarios where multiple scannable targets are close to each other, explicit click is recommended, because it avoids unexpected recognition of other markers. Finally, a sense of control could be gained through combining the explicit click into the scan process, which is worth considering when designing applications for scanning codes.

The setting of pictures in a gallery points out a direction for further research about scanning. Not all the pictures in a gallery is scannable, and this might make visitors confused if there are no clear signs showing whether a picture can be scanned. It could be investigated what forms of signs is needed when the scannable marker is not as obvious as QR codes. The experience brought by scanning markers of different design styles can also be studied, such as barcode, QR code, designable visual markers [3] and hidden markers within a picture. The former ones make it clear that it is scannable, while the last two kinds of markers might bring uncertainty to users.

How the sense of control may influence user experience when using scanning applications can also be researched. People's need for the sense of control varies according to the usage scenario, for example, the need might be urgent for scenarios of transactions or identification, while scanning a picture in a gallery does not yield a need of security. Aside from the existence of a button, what design features might help build a sense of control can also be explored.

ACKNOWLEDGMENTS

I would like to thank my supervisor, Enrico Costanza, for his patient guidance and feedback during the process of experiment design and application development. I also feel thankful to the twelve participants who came to take part in this experiment. Finally, I would like to express my gratitude to my family and friends for their emotional support.

REFERENCES

1. Dawn M. Armfield, Ann Hill Duin, and Isabel Pedersen. 2018. Experiencing Content: Heuristics for Human-Centered Design for Augmented Reality. In *2018 IEEE International Professional Communication Conference (ProComm)*, 241–247. https://doi.org/10.1109/ProComm.2018.00057
2. Stuart K. Card, George G. Robertson, and Jock D. Mackinlay. 1991. The information visualizer, an information workspace. In *Proceedings of the SIGCHI conference on Human factors in computing systems Reaching through technology - CHI '91*, 181–186. https://doi.org/10.1145/108844.108874
3. Enrico Costanza and Jeffrey Huang. 2009. Designable visual markers. In *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09*, 1879. https://doi.org/10.1145/1518701.1518990
4. Daniel B. DeSoto and Peskin Mark Andrew. 2015. Login using QR code, U.S. Patent 8,935,777, issued January 13, 2015.
5. Alan Ezust and Paul Ezust. 2011. *Introduction to Design Patterns in C++ with Qt: Models and Views*. Prentice Hall, New Jersey, United States.
6. Fabio E. Fontana, Ovande Furtado, Oldemar Mazzardo, and Jere D. Gallagher. 2009. Whole and Part Practice: A Meta-Analysis. *Perceptual and Motor Skills* 109, 2: 517–530. https://doi.org/10.2466/pms.109.2.517-530
7. Fernand Gobet, Martyn Lloyd-Kelly, and Peter C. R. Lane. 2016. What's in a Name? The Multiple Meanings of "Chunk" and "Chunking." *Frontiers in Psychology* 7. https://doi.org/10.3389/fpsyg.2016.00102
8. Tovi Grossman and Ravin Balakrishnan. 2004. Pointing at trivariate targets in 3D environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 447–454.
9. Yves Guiard and Michel Beaudouin-Lafon. 2004. Fitts' law 50 years later: Applications and contributions from human-computer interaction. *International Journal of Human-Computer Studies*.
10. Bonifaz Kaufmann and David Ahlström. 2012. Revisiting peephole pointing: a study of target acquisition with a handheld projector. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services - MobileHCI '12*, 211. https://doi.org/10.1145/2371574.2371607
11. Peter Kieseberg, Manuel Leithner, Martin Mulazzani, Lindsay Munroe, Sebastian Schrittwieser, Mayank Sinha, and Edgar Weippl. 2010. QR code security. 430–435.
12. Jacob Kittley-Davies, Ahmed Alqaraawi, Rayoung Yang, Enrico Costanza, Alex Rogers, and Sebastian Stein. 2019. Evaluating the Effect of Feedback from Different Computer Vision Processing Stages: A Comparative Lab Study. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*, 1–12. https://doi.org/10.1145/3290605.3300273
13. Katharina Krombholz, Peter Frühwirt, Peter Kieseberg, Ioannis Kapsalis, Markus Huber, and Edgar Weippl. 2014. QR Code Security: A Survey of Attacks and Challenges for Usable Security. In *Human Aspects of Information Security, Privacy, and Trust*, Theo Tryfonas and Ioannis Askoxylakis (eds.). Springer International Publishing, Cham, 79–90. https://doi.org/10.1007/978-3-319-07620-1_8
14. Ming Li, Katrin Arning, Luisa Vervier, Martina Ziefle, and Leif Kobbelt. 2015. Influence of temporal delay and display update rate in an augmented reality application scenario. In *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia - MUM '15*, 278–286. https://doi.org/10.1145/2836041.2836070
15. Kuan-Chieh Liao and Wei-Hsun Lee. 2010. A Novel User Authentication Scheme Based on QR-Code. *Journal of Networks* 5, 8: 937–941. https://doi.org/10.4304/jnw.5.8.937-941
16. Yaqiong Liu, Jing Ling, Guochu Shou, Hock Soon Seah, and Yihong Hu. Augmented Reality Based on the Integration of Mobile Edge Computing and Fiber-Wireless Access Networks. 6.
17. I. Scott MacKenzie. 1992. Fitts' law as a research and design tool in human-computer interaction. *Human-computer interaction* 7, 1: 91–139.
18. I. Scott MacKenzie and Colin Ware. 1993. Lag as a Determinant of Human Performance in Interactive Systems. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, 488–493.
19. Rupert Meese, Shakir Ali, Emily-Clare Thorne, Steve D. Benford, Anthony Quinn, Richard Mortier, Boriana N. Koleva, Tony Pridmore, and Sharon L. Baurley. 2013. From codes to patterns: designing interactive decoration for tableware. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*, 931. https://doi.org/10.1145/2470654.2466119
20. Robert B. Miller. 1968. Response time in man-computer conversational transactions. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I on - AFIPS '68 (Fall, part I)*, 267. https://doi.org/10.1145/1476589.1476628
21. Atsuo Murata and Iwase Hirokazu. 2001. Extending Fitts' law to a three-dimensional pointing task. *Human movement science* 20, 6: 791–805.

22. Eamonn O’Neill, Peter Thompson, Stavros Garzonis, and Andrew Warr. 2007. Reach out and touch: using NFC and 2D barcodes for service discovery and interaction with mobile devices. In *International Conference on Pervasive Computing*, 19–36.
23. Randy Pausch. 1991. Virtual reality on five dollars a day. In *Proceedings of the CHI ’91 Conference on Human Factors in Computing Systems*, 265–269.
24. Roman Rädle, Hans-Christian Jetter, Jens Müller, and Harald Reiterer. 2014. Bigger is not always better: display size, performance, and task load during peephole map navigation. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI ’14*, 4127–4136. https://doi.org/10.1145/2556288.2557071
25. Antonio M Rinaldi and Cristiano Russo. 2018. User-centered Information Retrieval using Semantic Multimedia Big Data. In *2018 IEEE International Conference on Big Data (Big Data)*, 2304–2313.
26. Michael Rohs and Antti Oulasvirta. 2008. Target acquisition with camera phones when used as magic lenses. In *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI ’08*, 1409. https://doi.org/10.1145/1357054.1357275
27. Michael Rohs, Antti Oulasvirta, and Tiia Suomalainen. 2011. Interaction with magic lenses: real-world validation of a Fitts’ Law model. In *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI ’11*, 2725. https://doi.org/10.1145/1978942.1979343
28. Rucksack Mobile App Development. 2018. Barcode Scanner for Amazon - United States. *Barcode Scanner for Amazon - United States*. Retrieved May 10, 2019 from https://play.google.com/store/apps/details?id=com.cannotbeundone.amazonbarcodescanner&hl=en_GB
29. Natalya Skoryukina, Dmitry P. Nikolaev, and Vladimir V. Arlazarov. 2019. 2D art recognition in uncontrolled conditions using one-shot learning. In *Eleventh International Conference on Machine Vision (ICMV 2018)*, 76. https://doi.org/10.1117/12.2523017
30. R. William Soukoreff and I. Scott MacKenzie. 2004. Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts’ law research in HCI. *International Journal of Human-Computer Studies* 61, 6: 751–789. https://doi.org/10.1016/j.ijhcs.2004.09.001
31. The Qt Company Ltd. 2019. The Declarative State Machine Framework. Retrieved July 18, 2019 from https://doc.qt.io/qt-5/qmlstatemachine.html
32. Eleanor Toye, Richard Sharp, Anil Madhavapeddy, David Scott, Eben Upton, and Alan Blackwell. 2007. Interacting with mobile services: an evaluation of camera-phones and visual tags. *Personal and Ubiquitous Computing* 11, 2: 97–106.
33. 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 ICME conference francophone on l’Interaction Homme-Machine - IHM ’13*, 1–6. https://doi.org/10.1145/2534903.2534905
34. Colin Ware and Ravin Balakrishnan. 1994. Reaching for objects in VR displays: lag and frame rate. *ACM Transactions on Computer-Human Interaction (TOCHI)* 1, 4: 331–356. https://doi.org/10.1145/198425.198426
35. Wayne A. Wickelgren. 1977. Speed-accuracy tradeoff and information processing dynamics. *Acta Psychologica* 41, 1: 67–85. https://doi.org/10.1016/0001-6918(77)90012-9
36. Yue Liu, Ju Yang, and Mingjun Liu. 2008. Recognition of QR Code with mobile phones. In *2008 Chinese Control and Decision Conference*, 203–206. https://doi.org/10.1109/CCDC.2008.4597299
37. Smartify. Retrieved May 8, 2019 from https://smartify.org/