Contents lists available at ScienceDirect

Int. J. Human–Computer Studies

journal homepage: www.elsevier.com/locate/ijhcs

Understanding mobile document capture and correcting orientation errors

Jeungmin Oh[a], Joohyun Kim[a], Myungjoon Kim[b], Woohyeok Choi[a],
SangJeong Lee[c], Uichin Lee[a,*]

[a] Graduate School of Knowledge Service Engineering, KAIST, Daejeon, Republic of Korea
[b] Department of Materials Science and Engineering, KAIST, Daejeon, Republic of Korea
[c] Developer Experience Lab, Software R & D Center, Samsung Electronics, Seoul, Republic of Korea

ARTICLE INFO

Keywords:
Device orientation
Automatic rotation
Document capture
Smartphone camera.

ABSTRACT

Smartphone cameras are increasingly used for document capture in daily life. To understand user behaviors, we performed two studies: (1) an online survey (n=106) to understand general smartphone camera usage behaviors related to information capture, as well as participants' experiences of orientation errors, and (2) a controlled lab study (n=16) to understand detailed document capture behaviors and to identify patterns in orientation errors. According to our online survey, 79.30% of the respondents reported experiencing orientation errors during document capture. In addition, our lab study showed that more than 90% of landscape capture tasks result in incorrect orientation. To solve this problem, we systematically analyzed the user behavior during document capture (e.g., video sequences and photographs taken or hand grip used) and propose a novel solution called ScanShot, which detects document capture time to help users correct orientation errors. ScanShot tracks the direction of gravity during document capture and monitors the users rotational or tilting movements of to update changes in orientation automatically. Our results confirm that document capture with 93.44% accuracy; in addition, our orientation update mechanism can reduce orientation errors by 92.85% using a gyroscope (for rotation) and 81.60% using an accelerometer (for micro-tilts).

1. Introduction

People use the built-in camera on their smartphone for various reasons, ranging from personal reflection to social experiences and functional tasks (e.g., Okabe, 2006; Gye, 2007; Lux et al., 2010). Research has shown that camera phones are often used for capturing functional images such as printed images or writing for later reference (Kindberg et al., 2004, 2005). Our work considers this type of document capture which is increasingly occurring in our daily life (e.g., capturing magazine or newspaper articles) (Brown and Sellen, 2000). While fixed scanners are still widely used in office settings, smartphone-based document capture allows users to instantly capture documents at anytime and in any location, which has dramatically influenced our document capture behaviors and the management of personal information (Doermann et al., 2003).

Document capture is typically done by configuring the angle of the smartphone camera into a top-down (or bird's-eye) view. However, as some readers may have experienced, orientation errors are often found in the captured images. We discovered that this type of problem originates from the inferred orientation of the phone being different from the capturing orientation of the user (hand posture). Recent smartphones have four orientation modes in a 2D space (just as in a picture frame on a wall), namely, portrait, upside down, landscape left (rotating the device to the left), and landscape right (rotating the device to the right), as shown in Fig. 1. For a given capturing orientation, there are three incorrect modes resulting in orientation errors during document capture from a top-down angle.

We began our study by conducting preliminary studies including online survey (n=106) and an in-lab experiment (n=16) with the goal to understand extensively the document capture behavior as well as the orientation errors. The research questions for each studies are 1) to investigate of real-world user experiences of capturing information using a smartphone camera from the online survey, and 2) to understand how people captures information using smartphone camera with analysis of the recorded video and sensor data from the in-lab experiment. The online study provides overall picture of contexts in which smartphone users captures information including documents. The findings are differentiated from prior works in that it studies real-world cases where smartphones are ubiquitous and is specific to information capturing. The in-lab experiment systematically shows

[*] Corresponding author.
E-mail addresses: jminoh@kaist.ac.kr (J. Oh), joohyun.kim@kaist.ac.kr (J. Kim), vcxzx@kaist.ac.kr (M. Kim), woohyeok.choi@kaist.ac.kr (W. Choi), j94.lee@samsung.com (S. Lee), uclee@kaist.ac.kr (U. Lee).

http://dx.doi.org/10.1016/j.ijhcs.2017.03.004
Received 6 August 2015; Received in revised form 2 March 2017; Accepted 10 March 2017
Available online 12 March 2017

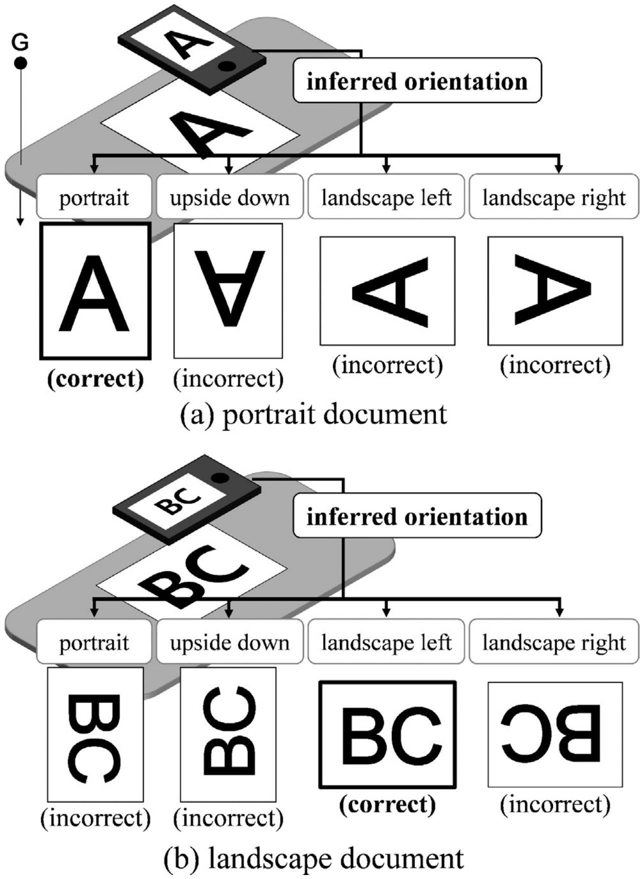


Fig. 1. Erroneous orientation problems when capturing a document using a smartphone.

how people captures the information in terms of hand grips, behavioral sequences with the details of orientation errors which turned out to be more severe in landscape mode.

Our video analysis of document capture during the in-lab experiment helped us propose ScanShot, a novel method for detecting document capture. Our approach is composed of two steps. First, ScanShot detects a document capture (when the body of phone is placed parallel to the ground) by monitoring the gravity direction using an accelerometer. For the second step, if a document capture is detected, ScanShot then attempts to update the orientation changes automatically. To achieve this, we proposed two different approaches. One approach is to detect rotation events by analyzing the recorded gyroscope data. When a phone is turned on (or a camera app is launched), it continuously records the gyroscope data. As soon as a document capture is detected, it examines the recorded gyroscope data to check whether any significant rotation changes previously occurred. Another approach is to infer the current orientation by observing the users micro-tilting behavior of the device while capturing a document. As the user holds their phone parallel to the ground, it is likely that they will tend to tilt inwards slightly (because the user will try to see the screen), a phenomenon we call micro-tilting. We previously showed that micro-titling behavior can be captured by monitoring the accelerometer and applying a machine learning model. To validate the efficacy of ScanShot, we carefully designed our algorithms and configured their parameters. Our evaluation showed that document capture moments can be detected with accuracy of 92.5% and automatic correction achieves accuracy of automatic rotation achieves accuracy of 92.85% (gyroscope) and 81.60% (accelerometer).

The key contributions of this paper are summarized as follows:

- We conducted an online survey (n=106) in order to investigate real-world user behaviors for capturing information using a smartphone camera and their awareness on the orientation issues.
- We performed an in-lab experiment (n=16) so as to study the detailed user interactions such as hand grips, behavioral sequences including video and sensor recordings.
- Based on the previous in-lab experiment, we describe a technique devised for inferring the user's document capture intention using an accelerometer. Our method can fairly precisely identify a document capture with an accuracy of 92.50%.
- We propose two methods for correcting orientation errors that occur during document capture when using a smartphone camera. The first method fixes the orientation errors by tracking the user's rotational movement using a gyroscope. The second method monitors the user's tilting behavior at the time of document capture to infer the correct orientation mode.
- We discuss several important issues including the generalizability of our algorithm and its integration into existing systems. In addition, we discuss practical design implications such as the design of context-aware services for document capture, investigating diverse hand grip positions for mobile interaction design, and increasing the awareness of viewfinder UI indicators.

The remainder of this paper is organized as follows. First, we start with a thorough review of related studies in Section 2. In Section 3, we then describe preliminary studies conducted to understand user behaviors regarding document capture using a mobile phone. Based on the insights acquired from the previous user study, in Section 4, we describe the design of ScanShot, which corrects erroneous orientations by monitoring the gyroscope and accelerometer sensors. In Section 5, we describe the performances of the two proposed methods. In Section 6, we discuss the generalizability and integration issues and suggest several practical design implications. Finally, we provide some concluding remark in Section 7.

2. Related works

2.1. Document capture using a camera

As the quality, accessibility, and functionality of digital devices improve, such tools are increasingly being used for information capture and storage. Brown and Sellen (2000) investigated the use of information capture in work settings to understand the motive and reasons behind the use of digital cameras for alternative purposes. They asked two groups of people in a workspace environment to capture any kinds of information and document-based information respectively, and interviewed them regarding what was captured and why the information was captured. They drew capture taxonomy which consists of 10 categories in order to explore design possibilities of new information capture devices. This work proposes the idea that cameras are taking on more functions beyond their original intended use in photography, especially for document capture and storage. However, their work was confined to an office environment; therefore, it is necessary to expand the study of information capture behavior to more naturalistic setting.

Most commonly, paper documents and marks on paper, such as hand-written notes are digitally saved using cameras. Kindberg et al. conducted an in-depth study to describe the intention and pattern of use of camera phones. In addition to typical photographs of people, cameras integrated with cell phones are often used to capture images of pages from books, screens, and writing on paper by interviewing actual users of camera phones (Kindberg et al., 2005, 2004). They also provided general statistics of camera phone use, and taxonomy of reasons for capture; however, their focus was on general photos, not photos for information capture. Ahmed et al. (2013) proposed a method to automatically generate camera captured document images because the dataset of document captured using a camera is expensive to collect.

2.2. Device orientation inference methods

Recent smartphones typically use gravity-based screen rotation algorithms (e.g., using an accelerometer), which assume that the users are standing or sitting upright while interacting with the device in their hand. That is, for inferring the orientation, the algorithms mainly consider only two axes (X and Y axes) of the gravity vector. However, the orientation inference fails to work properly if the smartphone is close to the plane parallel to the ground (e.g., taking a top-down shot, or placing the phone on a table), i.e., when the X and Y axes of gravity are close to zero. If we rotate the phone parallel to the ground while taking a top-down shot, the inferred orientation of the phone will remain the same, and thus, an erroneously inferred orientation will be obtained. According to a recent survey of user experiences regarding smartphone orientation, users reported three types of failures in an orientation inference; “lying down on one side,” “placing the device on a flat surface,” and “lying down while facing up” (Cheng et al., 2012). The orientation errors occurring during document capture are very similar to the “placing the device on a flat surface,” but occur while the phone is in the user's hands (not on a surface).

To the best of our knowledge, none of the earlier studies investigated the erroneous orientation problem in document capture systematically. Instead, researchers have mainly focused on studying how to prevent unwanted screen rotations when users change their body posture (e.g., lying down). We classified techniques related to automatic orientation inference into the following categories and discuss their limitations herein.

2.2.1. Gravity-based device orientation inference

Hinckley et al. (2000) proposed portrait/landscape display mode detection using a tilt sensor or two-axis accelerometer. The underlying ideas here are that users naturally tilt their device for device rotation changes (left/right, forward/backward), and sensing this behavior allows for automatic orientation inference. In their visual illustration of device orientation, the authors introduced a *Flat* range in which both tilt angles of the device fall within ±3 ° where no orientation changes are triggered. When the user places the device on a flat surface, a “put-down problem” arises as the device's tilting angle falls into the *Flat* range. In this case, the authors simply set the mode as the most stable recent orientation when the phone entered the *Flat* range. As shown later, this problem is similarly observed in document capture tasks, e.g., posing a top-down shot by moving the camera lens attached on the back of the device toward a document; the only difference here is that the user lifts the phone into the air to capture a document. In this case, users can freely rotate their phones to capture the documents correctly (for example, in landscape or portrait modes). As with the earlier work described above (Hinckley et al., 2000), the use of the previous orientation mode may cause orientation mismatch problems. Consequently, traditional gravity-based orientation systems do not work well for document capture.

Kunze et al. (2009) presented a method to infer the horizontal orientation of a mobile device carried in a pocket by processing the accelerometer signals when the user is walking. This method extends Mizel's approach (Mizell, 2003) in inferring horizontal and vertical directions. Accelerometer signals are projected onto the plane perpendicular to the gravity direction, and the first principal component values of the projected points are integrated to infer the horizontal orientation. This approach is not applicable to our document capture scenario because our work requires orientation sensing while the device is near the horizontal plane (angles toward earth's gravity).

2.2.2. Grasp-based device orientation inference

Cheng et al. (2013) explored how well a grasp can be used to infer the screen orientation by implementing a phone-sized grasp sensing prototype using 44 capacitive sensors attached to the back of a mobile device. The authors demonstrated its usefulness by prototyping a touch-sensor based phone case (achieving an inference accuracy of 80.9%), proving that the way in which the device is grasped is a good indicator of its orientation sensing. Lee and Ju (2013) developed a similar approach using three small, thin sensors). Wimmer and Boring (2009) proposed HandSense which employs capacitive sensors for detection when touched, squeezed, or held against a body part. Their machine learning algorithm is also capable of detecting which hand is holding the device. However, these methods are limited in that they require external sensors and grip postures for document capture are much more diverse (according to our user behavior study). Goel et al. (2012) built GripSense to infer hand postures (e.g., one- or two-handed interaction, and the use of the thumb or index finger) and the amount of pressure using built-in smartphone sensors such as screen-touch sensing, motion sensing, and built-in smartphone actuators such as vibration motors. Although inferring hand postures and pressure levels can provide valuable information regarding the inference of document capture and possibly correcting orientation errors, it is difficult to consider diverse capturing postures and infer the user's intention and correct orientation.

2.2.3. Computer vision based device orientation inference

Cheng et al. (2012) proposed iRotate which takes advantage of the front camera to detect the face orientation. This method is not applicable to our problem because the user's device is generally parallel to the document on a desk (flat plane), and thus the user's face cannot be captured consistently at this camera angle. In addition, their method is not applicable for devices without a front camera such as a digital camera. Alternatively, we can use optical character recognition (OCR) to automatically detect a document's orientation (Kwag et al., 2002; Lu et al., 2007; Le et al., 1994; Hinds and Fisher, 1990). However, its performance is heavily influenced by the context (e.g., font, handwriting, language, and light conditions), and it poses significant processing overhead when compared to motion data processing. Furthermore, our algorithm can cover broader situations such as capturing documents without characters in which correct orientations cannot be identified with computer vision based approaches.

3. Preliminary studies

3.1. Methodology

To understand user behaviors, we performed two user studies: (1) an online survey to understand smartphone cameras usage for information capture and participants' experiences of orientation errors, and (2) a controlled lab study to understand detailed document capture behaviors and to identify patterns of orientation errors.

3.1.1. Online survey design

We designed an online survey in order to investigate real-world user experiences of capturing information using a smartphone camera in addition to users' awareness of orientation issues. To encourage survey participation, we randomly selected a portion of the participants and compensated them with a gift certificate equivalent to 9 USD. The respondents were recruited from an online community of university students.

The main purpose of this survey study is to understand the frequency of smartphone camera use for information capture and the information that is captured in real-world situation.

We began by asking general questions on smartphone camera use for information capture purposes including experienced orientation errors while taking document photos (see Table 1). We clarify the questions that are difficult to understand by running the survey internally within our research team prior to the actual distribution of

Table 1
Survey results.

Question	Response (%)	
	Yes	No
Have you used the smartphone camera for the purpose of information capture? (i.e., not a portrait or landscape, used for information storage, sharing, collection, reference purposes)	98.11	1.89
Have you taken any document using the smartphone camera?	99.06	0.94
Have you experienced the orientation error issue while capturing a document using the smartphone camera?	79.30	20.70

the online survey. In addition, to help users understand the meaning of erroneous orientation issue, we also showed the participants photos of correct case and incorrect case. To investigate the information that people captured, we additionally asked respondents to report the photos they had taken for information capture in the last three months by reviewing their photo gallery app. They used text boxes to provide brief context on the information captured for up to 5 photos. We applied affinity diagramming which is commonly used technique to find categories on the collected data (Beyer and Holtzblatt, 1997). Two authors printed out all the answers in a small piece of paper and iteratively classified it until clear themes appeared according to what kind of information the answers were trying to capture (Fig. 2). In grouping the cases, we focused more on capture context than the captured content itself because the user's behavioral characteristic while photo taking, which is the purpose of this study, is more closely related to context than captured object. For example, although a presentation slide could be printed on paper or projected onto a screen, we differentiated between the two cases, as user behavior, and the resulting photos may differ according to the form of the target.

3.1.2. Lab study design

We conducted an in-lab experiment to understand user behaviors and find patterns of orientation errors. We also collected sensor data to use in developing a correction method during the study. The participants were asked to take photographs of 20 documents using a smartphone (Nexus 5) camera. We chose 20 sections from the pages of a National Geographic magazine and highlighted these sections using colored post-it flags. Half of these sections were vertically long documents (portrait orientation) and the rest were horizontally long (landscape orientation). Landscape and portrait documents appeared alternately. Before conducting the tasks, the participants engaged in a training session during which they took trial photographs, in order to become familiar with the experimental settings.

We designed the experiment by conducting a series of pilot tests. To mimic natural photo-taking behavior, we instructed the participants to take photographs of documents as usual. The participants were allowed to be seated or stand. The size of the target area to capture varied from a small section to an entire page, as earlier in this paper. As the only constraint, the users were asked to place the device on the desk between photographs in order to treat each one as an individual task. The phone's position on the desk was not specified, therefore some variation naturally occurred. The reason for including various factors in our experimental design is to conduct our user experiment in realistic environment, which eventually will allow us to derive more generalized results and solutions.

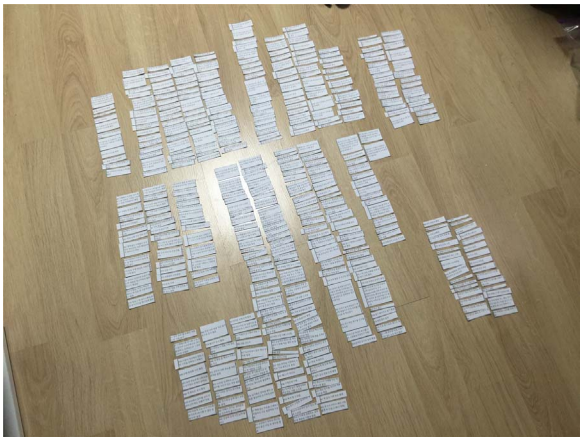


Fig. 2. How affinity diagramming were performed.

An Android reference phone, or a Nexus 5, and its pre-installed camera app were used in this experiment. All portions of the experiment were videotaped while the users interacted with the devices. In addition, the sensor data were recorded during the experiment using a sensor recording app.

Sixteen people with experience in smartphone document capture using a smartphone were recruited from a large university. Their ages ranged from 20 to 32 years (mean: 23.1, SD: 3.7), and seven were female. None was left-handed. All participants were compensated with 5 USD in cash.

3.2. Online survey results

3.2.1. General response

The survey results showed that photos for information capture including those of documents, are quite prevalent among our respondents (Table 1). The number of respondents was 106 with a mean age of 24.71, with std. 4.01 (max: 41, min: 28); 56.25% participants were male, and 43.75% were female. 10 participants were rewarded with a gift certificate equivalent to 9 USD by lottery. Almost all respondents (98.11%, 104/106) utilized a smartphone camera for information capture purposes other than portrait or landscape photos. Document capture using smartphone cameras was also popular, 99.06% of the users responded that they have taken document photos using a smartphone camera. On average, the users took 7.29 document photos per month (Fig. 3). Orientation errors were quite common among our respondents with 79.30% having experienced this issue.

3.2.2. Category of captured information

To determine what information is captured, we analyzed 410 answers collected from 106 respondents. We removed 31 answers because those were unclear or difficult to interpret. We classified the remaining 379 answers using affinity diagramming, and classified the answers into 7 categories as shown in Table 2.

Overall, our analysis resulted in two high-level themes, namely, document and non-document groups, depending on whether the target object of the photo is a document. The document group divides into printed documents and hand written documents. Among printed documents, we further found more specific themes depending on whether the target object of the photo is attached to any fixture (e.g., walls, and objects), namely *unattached printed documents*, and *printed documents attached to a fixture*. The remaining documents

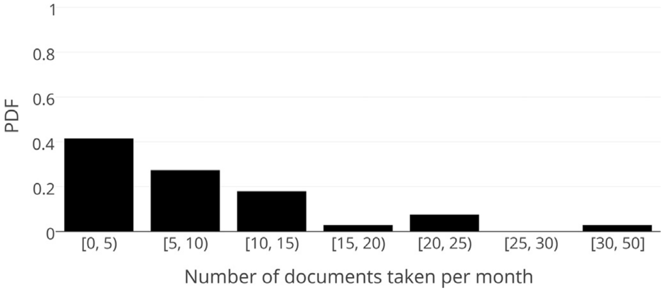


Fig. 3. PDF for number document taken per month.

Table 2
Categories of information captured using smartphone cameras.

Category	Ratio (%)	Examples
Unattached printed documents	26.39	Book, magazine, brochure
Printed documents attached to a fixture	24.54	Poster on a wall, price tag on clothes
Hand written documents	13.72	Lecture note, idea sketch
Projected slides	11.87	Lecture or seminar slide
Black/White boards	7.92	Writing on a chalkboard
Computer screens	5.01	Error screen, e-mail
Miscellaneous	10.55	Goods to buy, food before eating

were classified as *hand written documents*. In our data set, no instance of a hand written document attached to a fixture was found. For non-document groups, we found three categories: *projected slides*, *black/white boards*, and *computer screens*. The photos that do not belong to any of above categories were grouped as *miscellaneous*: for example, goods to buy, or food before eating (for journaling purpose). The details for each category are as follows.

The *unattached printed documents* are general documents that is not attached to any fixture. This category includes books, magazines, brochures, booklets, receipts, transcripts, and identification cards. People also captured part of documents: for example, a sentence in a page of a book, or a picture in a magazine. The majority were in place of digitizing the documents using a scanner for online submission or archiving.

The *printed documents attached to a fixture* group consists of documents that are attached to something, such as a wall and object. The majority of this group is wall posters: for example, posters for events, announcements, or recruitment attached to a wall; research posters at an academic conference, and menus in restaurant. Additionally, small documents, such as price tags on clothes, nutrition facts on a bottle, WiFi passwords in a cafe, bank account information on a utility bill, or the model number of an electronic device. These photos are taken mainly for further reference or sharing. It seems that information short enough to be typed in manually is often captured by taking a photo. For instance, WiFi passwords, and bank accounts were reported even though they might be easy to type in.

The photos of *hand written documents* are images containing hand writing, for example, an idea sketch and a solution to math question. Capturing note taking from a lecture and archiving assignments were the most common cases for this category because our respondents were primarily from the university community.

In addition to tangible documents, many photos of *projected slides* taken during seminars or lectures were found. Respondents captured these for two reasons. First, the slide is not publicly accessible. Second, people want to keep bookmark-like information for a specific slide; in this case, access to the slide files does not matter.

The *black/white boards* are photos of boards with writings from seminars or lectures. This category is different from the *hand written documents* in that it is written on a large wall surface such as a blackboard or whiteboard. Many took photos of black/white boards during lectures instead of taking notes for convenience. We found two motivations for doing this. First, people did it in order not to miss any content for the note taking because the lecturer often erases the board before the audience takes notes. Second, some reported that they are too lazy to take notes. It seems that note taking is being replaced by smartphone camera photos due to the convenience of taking photos and the laziness of people.

The *computer screens* are photos of PC or tablet screens as the name suggests. Although computers themselves have screenshot capabilities, many people reported that they share a screen by taking photos of it. It is assumed that it is more convenient to do so for sharing the scene with other people because of the prevalence of mobile messengers. In addition, there were situations in which screenshots cannot be captured such as a Windows error screen (generally called a blue screen) or console environment (for installing an operating system).

The *miscellaneous* are photos which did not belong to any category above. Example in this category includes photos of goods to buy (cosmetics, books, groceries), records of foods eaten (for food journaling purposes), research experiment progress, workout posture, and so on. We also observed that in a large portion of these instances, photos are used as to-do list (e.g., something to buy, something to send, somewhere to visit).

3.2.3. Summary

Our online survey showed that almost all respondents captured information including documents using a smartphone camera. Almost 80% of the users experienced the orientation errors while capturing documents, indicating that the problem is widespread. Among captured information, 64.65% were various types of documents: *unattached printed documents*, *printed documents attached to a fixture*, and *hand written documents*. We assume that the *unattached printed documents*, and *hand written documents* suffer from orientation issues most often because the target documents are generally placed on a flat surface such as a desk, which is highly relevant to the root cause of the problem (Section 3.3.2).

Compared to the paper of Brown and Sellen (2000), the captured information differs from our result, especially in terms of the distribution of items. In their work, the marks on papers were the most frequent for multimedia groups, whereas groups of printed documents comprised the majority of information captured in our study. This might be attributed to our data set, which was collected from natural settings with actual users' data. Our result also provides a more detailed view for categories which are described simply as a ‘specific item’ or an ‘image of screen, writing and so on’ in the paper of Kindberg et al. (2005).

3.3. Lab study results

3.3.1. Orientation errors

After the experiment, we counted how many photographs were taken with the wrong orientation. The error rates for the landscape and portrait modes were 0.93 (SD, 0.18) and 0.04 (SD, 0.17), respectively. The landscape capture tasks showed much higher error rates because the Android device used portrait mode as the default orientation (Table 3).

Errors mainly occurred because the user held the device perpendicular to the gravity, where a gravity-based orientation system does not work properly. Most of the participants except P11 showed consistent patterns, i.e., almost all of the landscape documents were captured with an orientation error, whereas the portrait documents were captured correctly (Table 4). P11 tended to tilt the device by approximately up to 45 ° before posing a top-down shot, which changed the orientation to the intended mode before the gravity-based orientation system began working incorrectly.

3.3.2. Video analysis

We investigated the user behaviors by carefully reviewing the recorded videos, thereby gaining insight into fixing orientation errors. Our analysis showed that the overall process consists of four steps: 1)

Table 3
Summary of orientation error.

Document	Correct	Rotated to left	Rotated to right	Upside-down	Total
Portrait	153	7	0	0	160
Landscape	11	147	1	1	160

Table 4
Error rate of 16 participants.

Document	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16
Portrait	0	0	0	0	0	0	0	0	0	0	0	0	0	0.7	0	0
Landscape	1	0.7	1	1	1	1	1	1	1	1	0.3	1	1	1	0.9	1
Overall	0.5	0.35	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.15	0.5	0.5	0.85	0.45	0.5

launching the camera app, 2) moving toward the target object and optionally adjusting the orientation by rotating the device, 3) composing a top-down shot (including zooming, tilting, and panning), and 4) touching the device to focus and press the shutter button.

First, the participants turned the screen on and launched the camera app when the phone was placed on the desk (55%) or in their hand (45%). This behavior should be carefully considered because sensor readings vary widely depending on the starting positions. Second, in the case of the landscape tasks, the participants adjusted the phone's orientation. The portrait capturing tasks do not require an orientation adjustment because the phone's default orientation is portrait mode (as is the smartphone's home screen orientation). Among the four orientations, with the exception of one session, we observed only two orientations in the experiment. This pattern may be consistent with left-handed users, which is very important because we can reduce the number of possible states when designing an automatic algorithm. Note that this step generally comes after launching the camera app, but we also observed that in a few landscape sessions, app launching was followed by device rotation. The one exception during which the participant rotated the phone to the right during the landscape tasks occurred because the participant accidentally dropped the phone during the session. Because the participant answered in our post-questionnaire that they never rotated the phone to the right for landscape use, we were able to ignore this case. Third, the participants then held their phone parallel to the plane where the document was placed and attempted to zoom/pan to capture the target area. Finally, once the target area was determined, the participants touched the device's screen to focus (occasionally) and pressed the shutter button to take the photograph.

In addition, other relevant behaviors were observed. Two participants (P12 and P16) took all of their photographs while standing, but their moving and rotation timing and grip types were not significantly different from those of other target objects. Three of the participants (P7, P8, and P16) continuously touched the screen to focus before pressing the shutter button. For this reason, the capture states of these participants were longer than the states of the other participants.

We further found that an uncaught rotation was the root cause of the problem. We observed that the system does not trigger an orientation change event when the user physically rotates the device. After the user moves the device toward the target object with an uncaught rotation, the outcome falls into one of following cases: 1) the initial orientation mode remains unchanged (portrait mode in our experiment setting), 2) the orientation mode changes to an unintended mode owing to the orientation error sensing in existing gravity-based orientation systems. We also noted that the errors occurring during landscape tasks were mainly from an uncaught rotation, and the errors occurring during portrait tasks were due to unintended transitions into an arbitrary orientation.

3.3.3. Analysis of captured photographs

We analyzed the captured photographs to further investigate the characteristics of document capture. First, we found that the margin around the target area varied widely among the different participants. Furthermore, the target areas within the photographs were captured at diverse sizes in the photos. Interestingly, we found that the content captured in the photographs showed some skewness in both the left-right and forward-backward directions. This indicates that the user did not perfectly set their phone parallel to the ground, and the device was tilted when the shutter button was touched. To calculate the skewness, we cropped two edges of the content within the captured photographs (upper and right edges) and calculated the border slopes. As shown in Table 5, the skewness varied across the orientation modes. For example, in portrait mode, an average of 2.29 ° of forward tilting occurred. Although the observed tilt was less than 3 °, we hypothesize that titling may be a good indicator for orientation sensing Table 6.

Table 5
Skewness of content in the photographs.

Document	Horizontal (Up)	Vertical (Right)
Portrait	2.29°	0.57°
Landscape	0.51°	2.86°

3.3.4. Hand grip analysis

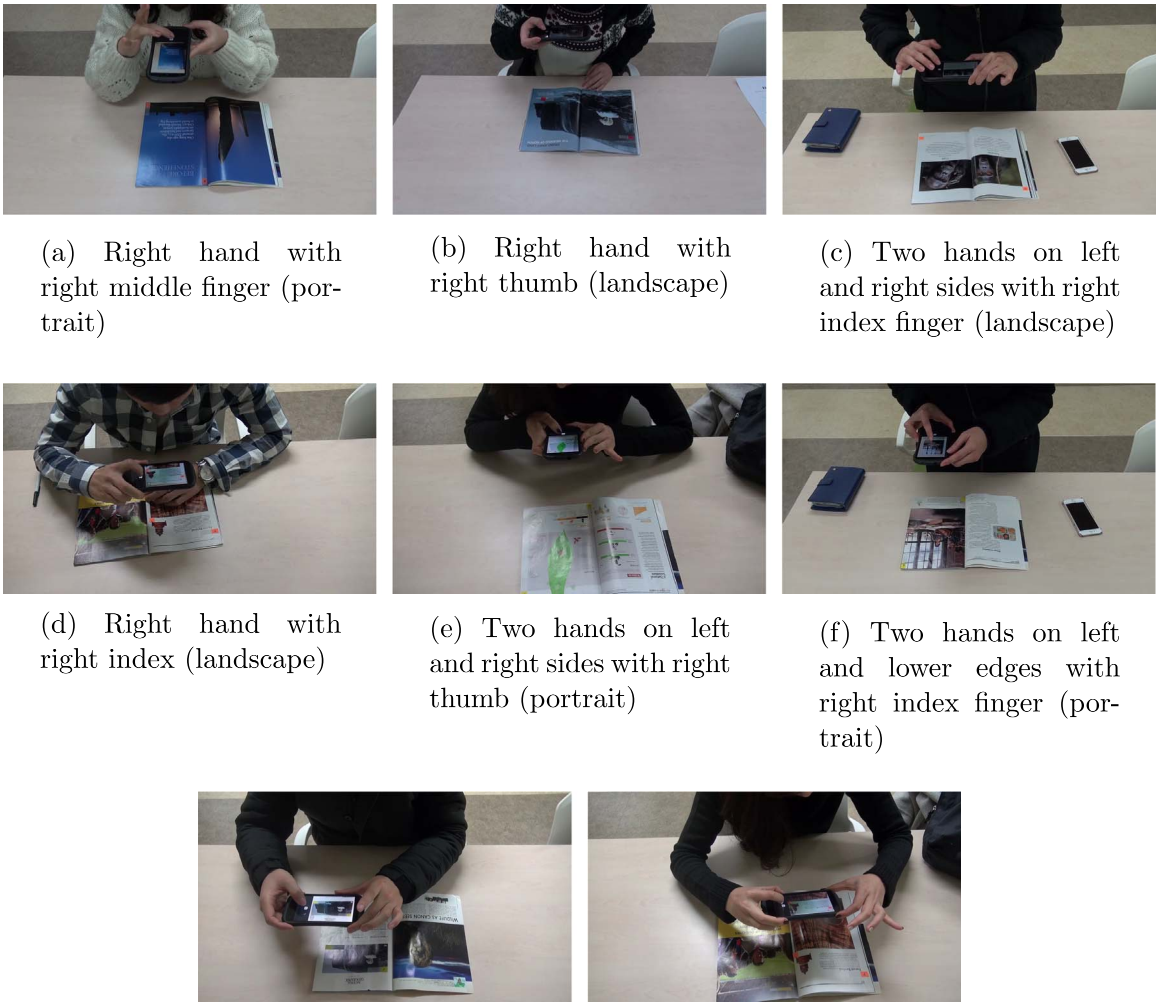
The grip type varied from person to person, and could be classified based on the number of hands used and the preferred grip. In 308 out of 320 sessions, the participants used two hands. Only 12 of the sessions showed one hand use for document capture. Two of the participants tended to use a one-handed grip (8 out of 12 sessions). The others applied a one-handed grip when using the other hand to unfold and hold the book flat or to touch the screen for focusing. It was also noted that some of the participants placed one hand under the device for support, or picked the device up with their fingers.

The grips used for document capture fall into eight types, detailed photographs of which are shown in Fig. 4. When capturing a landscape document, with the exception of four sessions, all of the participants picked up the smartphone with two hands (Fig. 4(a)). In the other sessions, the participants supported the smartphone using their right hand and picked it up using their left. For portrait mode, four grip types were observed for two-handed usage, depending on which hand the participants used for picking up the smartphone and which hand they used for support when capturing the image. This included picking up the phone with two hands, two-handed support, picking up the phone with their left-hand and supporting it with their right, and picking up the phone with their right hand and supporting it with their left.

Most of the participants tended to use their preferred grip types consistently during the entire study. However, some of the participants

Table 6
Summary of behavioral codes.

Tiers	Annotation	Definition
MCR state	move	Moving the device toward a target object
	capture	Document capture mode (top-down shot)
	return	Returning back to original position
Rotation state	rotate-left	Rotating the device to left
	rotate-right	Rotating the device to right
Event	grab	Grabbing a device
	app	Launching a camera app
	focus	Making a focus on target object by touching the screen
	shutter	Pressing a shutter button



(a) Right hand with right middle finger (portrait)

(b) Right hand with right thumb (landscape)

(c) Two hands on left and right sides with right index finger (landscape)

(d) Right hand with right index (landscape)

(e) Two hands on left and right sides with right thumb (portrait)

(f) Two hands on left and lower edges with right index finger (portrait)

(g) Two hands on left and right sides with right thumb (landscape)

(h) Two hands on left and right sides with right index (landscape)

Fig. 4. Grip types used for camera apps.

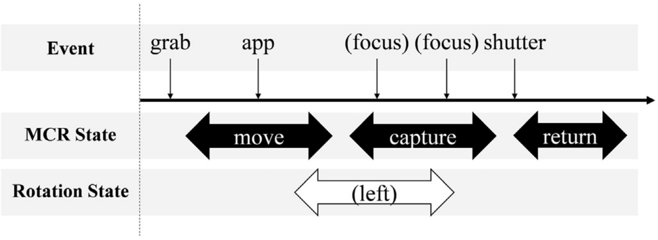


Fig. 5. Overview of the coding scheme.

changed their grip type when they needed to use one hand for another purpose such as focusing and unfolding the book. Some of the participants varied their grip postured depending on the size of the image to be captured.

3.3.5. Summary

From our preliminary lab study, we found some insightful user behaviors for solving an orientation error. The key results and their design implications are summarized as follows.

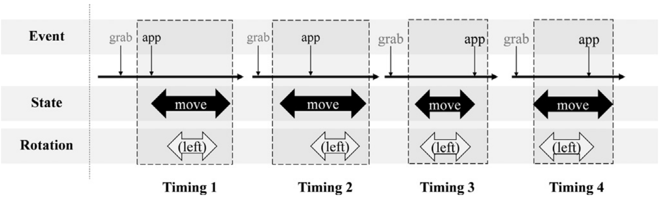
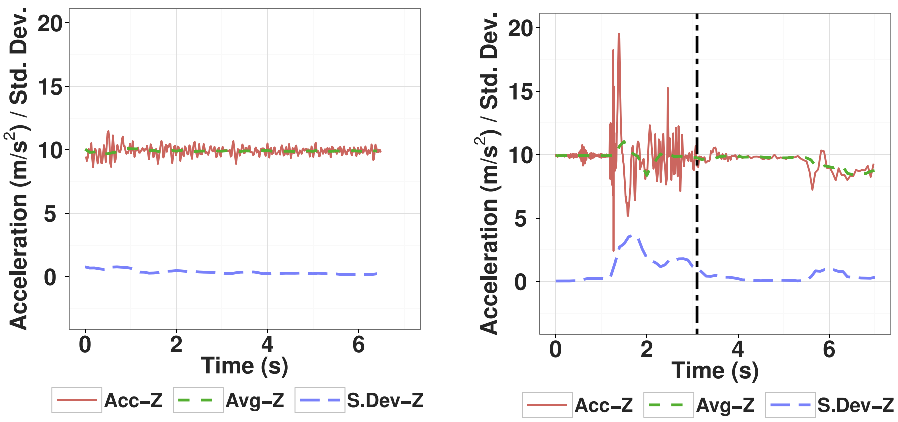


Fig. 6. Movement and rotation timing in terms of app event.

- While capturing a document, the users are likely to keep their device parallel to the document which is generally located on a nearly flat plane (e.g., a desk). This causes a gravity-based orientation system to operate improperly, thereby generating many errors such as ignoring the changes in orientation after rotating the device or switching to an unintended orientation. Therefore, detecting a user's document capture intention, which can be characterized as the device being in a parallel plane, is useful for solving the problematic occurrence of an orientation error.
- We found that changes in orientation resulting from uncaught rotations are the root cause of this problem. Most of the errors



(a) Acceleration value traces without a change in orientation

(b) Acceleration value traces with a change in orientation

Fig. 7. Accelerometer used in a capturing task: (a) portrait shot, and (b) landscape shot.

occurred during landscape tasks, and when the users rotated the device when it was placed on a flat plane. Thus, we can track the rotation to infer orientation errors, particularly when a user is in the document capture mode.

- According to our captured photo analysis, the captured photographs were likely to be skewed by up to approximately 3°. This means that the users tend to tilt the device despite their effort to keep it parallel to the document. In addition, we found that the degrees of tilting for landscape and portrait tasks differed from each other. This showed that the orientation mode can be inferred by monitoring tilting the degree of tilt when a user is in document capture mode.
- The types of hand grips used are more diverse for document capture when compared to normal use. The use of hand grip would be a less effective means for orientation correction. Furthermore, inferring diverse grip postures would be quite a challenging task.

4. Scanshot design

Our preliminary user study guided us to design a novel solution called ScanShot for detecting document capture and help the device correct any orientation errors. The key concept underlying ScanShot is that we can reliably detect whether a user is taking a top-down shot by tracking an accelerometer. Once a top-down shot is detected, we then analyze multiple sensor data to infer the current orientation mode. We propose two methods for correcting an orientation error, one using a gyroscope to track the rotation, and the other using an accelerometer to calculate the orientation mode by sensing any slight tilting of the device.

4.1. Capturing motion analysis

Based on our in-lab experiment, we analyzed the collected sensor data to identify the behavioral patterns during capturing task. Three of the authors annotated the sensor data by watching recorded videos. We used ELAN software,[1] which aids in video coding. The detailed coding scheme is as follows.

An *MCR state* tier represents the movement of a device. This starts from a *move* state and ends with a *return* state. A *capture* state means adjusting the angle subtly when composing a top-down angle. We added this state because an orientation error only occurs in context of a top-down shot. The device seems to stay still without no subtle hand motions in this state. This includes tilting and panning. A *return* state indicates when the device is moved back to its original position. These three states occur in sequence within a particular session.

A *rotation state* tier indicates the rotation of the device, which is related to the change in orientation. A rotation is separated because it is observed concurrently with an MCR state tier. Although there are two possible directions of rotation, left and right, we only found left rotations during our experiment. This is due to two reasons: 1) all of the participants held the device with their right hand, which is their dominant hand, which made a left rotation much more convenient than a right rotation, and 2) the software and hardware are favorable for using landscape left orientation mode, which even causes a left-handed person to rotate the device to the left, e.g., most built-in camera lenses are located at the top.

An *event* tier refers to an instantaneous event triggered at a specific point in time. *Focus* refers to an event in which a user touches the screen to focus the camera at the document. Events are optional and can be observed multiple times. Note that each annotation can overlap within other annotations in different tiers (Fig. 5). For example, the user may move the device toward the target object while rotating the device and launching the camera app with their finger.

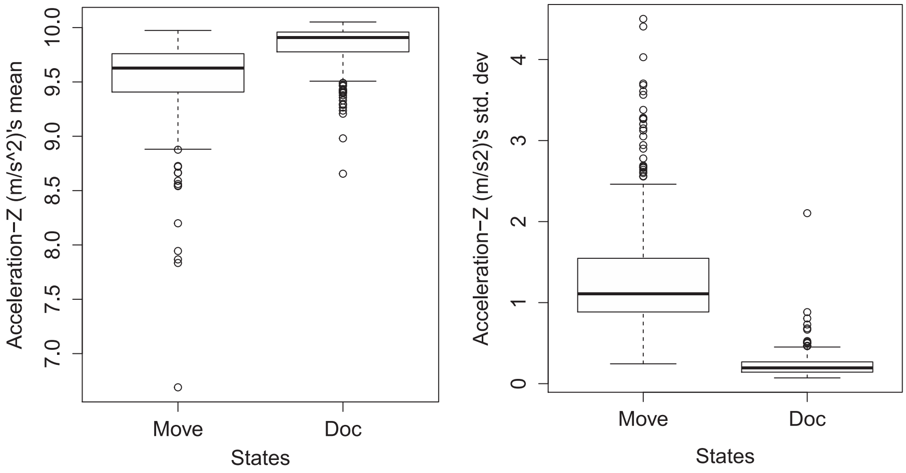
We also analyzed the sequential steps of *move state* and *rotation state* with respect to the *app* events. Four different timings were found (Fig. 6). In Timing 1, the user first launches the camera app, and then moves the device toward the target object. A rotation is found during this movement. In Timing 2, the user starts to move the device toward the target object and then launches the app: here the rotation occurs during the movement. In Timing 3, the user movement and rotation come first, and the camera app is then launched. In Timing 4, the user moves and rotates the device simultaneously. The user launches the camera app before the movement ends.

This finding was useful for determining when to enable sensor monitoring for automatic detection of the device rotation. The users rotated the device either before (Timings 3, 4) or after launching the app (Timings 1, 2). For this reason, the system was expected to monitor the sensor values prior to launching the camera app because a physical rotation was occasionally observed before the participants launched the app.

4.2. Document capture detection

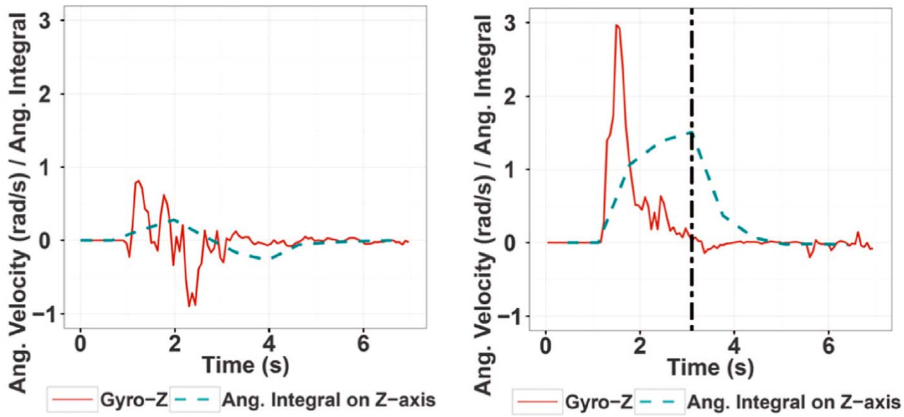
To detect document capture corresponding to the capture state, we propose a gravity-based detection method. When a camera app is launched, the accelerometer values fluctuate because of the movements (switching to a top-down shot with zooming/panning). After a few seconds, the accelerometer values are stabilized because the user must touch the screen to focus and then press the shutter button. When the

[1] ELAN: https://tla.mpi.nl/tools/tla-tools/elan/.



(a) Mean (b) Standard deviation

Fig. 8. Comparison between move and capture states.



(a) Angular value traces without orientation change (b) Angular value traces with orientation change

Fig. 9. Gyroscope traces during a capturing task: (a) portrait shot, and (b) landscape shot.

photograph is about to be taken, the Z-axis of the accelerometer is close to the gravitational acceleration, or 9.8 m/s^2. When taking a top-down shot, the participant's hands may shake slightly causing small fluctuations, as shown in Fig. 7. The sampling rate for the accelerometer was 125 Hz.

To distinguish the capture states from the move states, we calculated their descriptive statistics (mean and standard deviation). The values clearly show a distinction between the two states. As expected, the mean of the capture state is close to 9.8 m/s^2 and the standard deviation of the capture state is much lower compared with that of the move state (Fig. 8). After removing any outliers, we set the threshold for document capture detection. Finally, we decided to enable the document capture when the mean and standard deviation of the acceleration value are within the range of [9.45, 10.05] and [0, 0.45], respectively. Note that a clear differentiation exists between taking a photograph of a document and taking a normal photograph is because the Z-axis value of the accelerometer will be nearly zero.

4.3. Orientation inference: rotation tracking (gyroscope)

We chose a gyroscope to detect changes in orientation because it can accurately sense a rotation of the device (Fig. 9). The sampling frequency of the gyroscope was set to 128 Hz which was obtained by selecting the SENSOR_DELAY_FASTEST setting in the Android SensorManager class. We used a moving window approach. For sensor data processing, the sliding window of size w seconds moved over time, with a segment size of s s.

To determine the appropriate window size, w, we analyzed the rotation time. According to the analysis (Fig. 10), most rotating actions

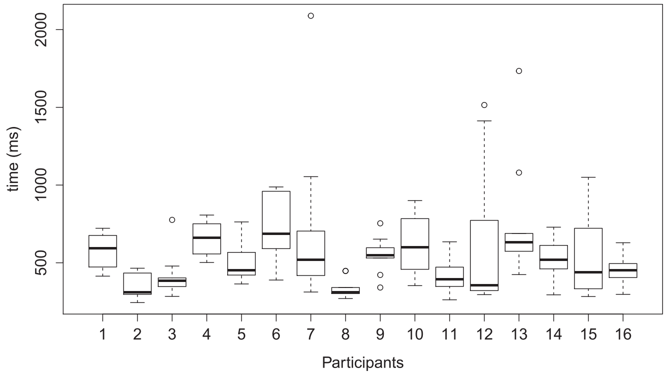


Fig. 10. Time distribution for device rotation.

took less than 2 s. The maximum length was 2089 ms and the minimum was 244 ms.

Because the sampling rate was 128 Hz (one sample per 66 ms), we set the segment size to s=0.66 s (ten samples per segment), and the window size to w=1.98 s (three segments per window). For a given window, we integrated the Z-axis values of the gyroscope samples. Thus, when this value was greater than the rotation threshold (ROT_THRESHOLD), we classified it as a left rotation event; if it was lower than the negative value of rotation threshold, it was classified as a right rotation event. Otherwise, we assumed that no rotation was made. By carefully analyzing the data set, we set the rotation threshold used to detect a rotation event to ± 0.5, which corresponds to ±28.6° according to parameter analysis (see Section 5.2). The overall process of this approach is described in the following pseudo code (Algorithm 1).

Algorithm 1. Pseudo code of rotation based ScanShot.

```
SET adjustedOrientation to DEFAULT_ORIENTATION;
repeat
    if a new window is generated from sensor listeners then
        adjustedOrientation = CalculateOrientationFromMicroTilt();
        if 9.45 < avg. acc-Z < 10.05 and std. acc-Z < 0.45 then
            Use adjustedOrientation;
        else
            Use conventional gravity-based orientation;
        end
    end
until shutter button is pressed;
```

4.4. Orientation inference: tilt monitoring (accelerometer)

Based on our observation that users tend to tilt their phones slightly, we devised a tilt based solution for the inferring correct orientation. Because an accelerometer contains information on the position of the device, including any tilting causing a skewness of the photographs, we first generated numerical features statistically representing the status of the sensor values during capture state. Different from the rotation based solution, this solution only makes use of sample values during the capture state. Once the capture mode is detected, ScanShot extracts the related features to update the orientation using the sensor values within the most recent time window.

The extracted features are the means of, and difference between acc-X and acc-Y. We also generated binary features by converting the numerical features into binary values (Table 7). We tested the binary features because the numeric values could be highly dependent on the characteristics of each participant, which would eventually affect the generalizability of our algorithm because binary features only contain high-level information of the user's posture (e.g., the direction the device is inclined toward). To investigate how well the features predict the orientation and how much difference exists between the numerical and binary features, we calculated information gain which is commonly used for measuring the predictive power of features (Fig. 8). The results confirmed that the binary features also show a comparable predictive power.

The following pseudo code describes how our tilt-based method is applied (Algorithm 2). Once the camera app is launched, the ScanShot loop begins. Every time a new window is generated from the sensor listener, it calculates the new orientation based on the tilting pattern data. If the device is capturing a document, the calculated orientation overrides the existing gravity-based orientation information. Otherwise, our method does nothing.

Algorithm 2. Pseudo code of tilt-based solution.

```
SET adjustedOrientation to DEFAULT_ORIENTATION;
repeat
    if A new window is generated from sensor listeners then
        Calculate cumulative sum of gyroscope for the window;
        if Cumulative sum > ROT_THRESHOLD then
            adjustedOrientation = LANDSCAPE_LEFT;
        else if Cumulative sum < -ROT_THRESHOLD then
            adjustedOrientation = PORTRAIT;
        end
        if 9.45 < avg. acc-Z < 10.05 and std. acc-Z < 0.45 then
            Use adjustedOrientation;
        else
            Use conventional gravity-based orientation;
        end
    end
until Shutter button is pressed;
```

5. Evaluation

We evaluated ScanShot using the data collected from our previous experiments because the entire sensor data while capturing document was captured in previous in-lab study. We first analyzed the accuracy of the document capture detection, and then evaluated the efficacy of the two orientation correction methods, namely our rotation- and tilt-based solutions. Our evaluation focused on the overall performance, user variance, parameter sensitivity, and an analysis of misclassified instances. For the tilt-based method, we additionally investigated the impact of the classifier differences. The dataset used for our evaluation is publicly accessible online (https://zenodo.org/record/56276). Note that this evaluation is mainly to benchmark the performance and effectiveness of the proposed methods such as 1) if the proposed methods are able to fix the orientation errors, 2) how well our methods works for correcting the orientation errors, 3) which parameters are effective for the proposed methods.

5.1. Document capture detection

To evaluate the performance of our capture moment detection algorithm, we first tested the its overall accuracy. We also investigated 1) the performance across different users and document types, 2) the reasons for any misclassified instances, and 3) the influences of the parameter values. To see how many instances were correctly identified as document capture, we first extracted samples marked as a capture state and then applied two conditions for distinguishing the moment of document capture (mean and standard deviation of acc-Z) for each sample. Before the analysis, we removed the first and last 10% of the samples in the capture state because the annotation of the states could contain some errors near the boundary between each state. Our evaluation showed that the overall accuracy of document capture detection was 93.44% (SD, 0.09), i.e., 299 out of 320 instances were correctly classified.

We further calculated the accuracy for each participant in order to check whether a user variance exists (Fig. 11). The majority of our participants (12 out of 16) showed an accuracy greater than 0.90. Although there were several participants who showed a lower performance (P2, P4, P8, and P15), their accuracy values were still greater than 0.70 Table 8.

Table 7
Two sets of features (numerical and binary).

Type	Features
Numerical	accX mean, accY mean, accX mean - accY mean
Binary	whether accX>0, whether accY>0, whether accX>accY

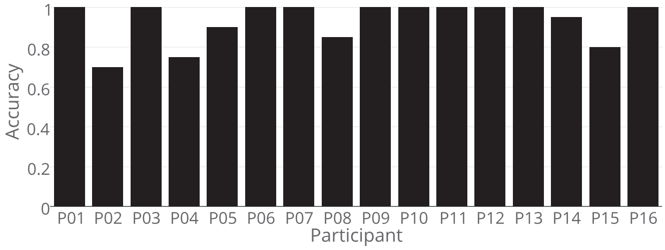


Fig. 11. Accuracy of document capture detection for each participant.

We also looked at 21 misclassified instances for better understanding of the errors that occurred, as shown in Table 9 where only users with errors are reported. The misclassified instances occurred mainly owing to the fact that the mean of acc-Z is smaller than 9.45 m/s^2, which was our threshold (18 instances in total). This means that some of the users tended to tilt the device more than we expected. On the other hand, five of the instances did not satisfy the standard deviation, which means that some of the users shook their hands during the capturing, which resulted a higher deviation than our threshold. In addition, two instances failed to meet both conditions.

Finally, we analyzed the sensitivity of parameters by changing the parameter values. We analyzed two parameters: the threshold for the mean and the standard deviation. The threshold for the mean was defined as the allowed difference from 9.8 m/s^2. For both parameters, accuracy converged to 100% as the threshold increased. As described in Fig. 12, thresholds for the mean and the standard deviation should be at least 0.25 and 0.35 respectively to achieve 90% accuracy. It should be noted that overly large thresholds may include non-document capture behavior as document capture.

5.2. Rotation-based solution

The rotation-based solution demonstrated its effectiveness by fixing most of the orientation errors for landscape documents (92.85 percentage points) as shown in Fig. 10. It significantly reduced the error rates for the portrait task. We manually investigated the error cases (four instances in total), but did not find any notable patterns. Compared to our previous work (Oh et al., 2015), we were able to improve the performance of the algorithm significantly (i.e., 59% point to 92% points in the case of landscape shots) because we considered sensor data processing even before launching the camera app.

Our initial parameters (rotation threshold, 0.5; window size, 1.98) were carefully chosen. As described in the following, we performed additional sensitivity analysis of these parameters. We systematically evaluated two of the parameters: the threshold for rotation detection and window size for capturing the rotation events.

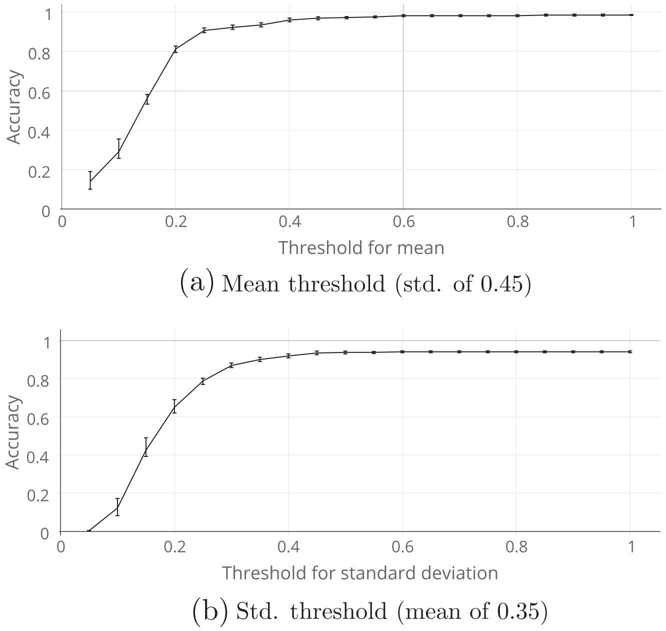
For the rotation threshold, we tested the values ranging from 0.1 to 1.9 with a 2-second window because the smallest rotation time was 244 ms and the maximum of rotation time observed was 2 s (Section 4.3). As the rotation threshold increased, the accuracy increased up to 0.98, where the thresholds were 0.5 or 0.6, and started to monotonically decrease after 0.70 (Fig. 13a). This is because such a small threshold value is insufficient for capturing a rotation event, whereas overly large values make the capturing too strict, causing many events to be missed. From this result, we can conclude that a rotation threshold of 0.5 is the best choice for detecting changes in orientation when using a gyroscope.

Table 8
Information gain of extracted features.

Feature	Type	Information Gain
accX mean - accY mean	Numerical	0.6704
accY mean	Numerical	0.4954
whether accX>accY	Binary	0.4743
whether accY > 0	Binary	0.4473
accY mean	Numerical	0.3733
whether accX>0	Binary	0.0863

Table 9
Number of misclassified instances of document capture detection.

Condition	P2	P4	P5	P8	P14	P15	Total
Mean of Acc-Z	6	5	0	3	1	3	18
Std. of Acc-Z	0	0	2	0	0	3	5
Total	6	5	2	3	1	4	21



(a) Mean threshold (std. of 0.45)

(b) Std. threshold (mean of 0.35)

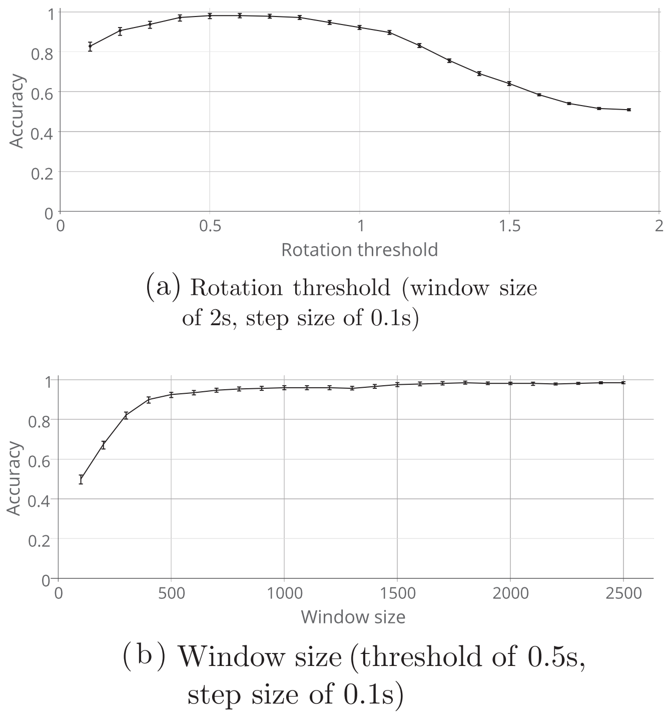
Fig. 12. Parameter sensitivity analysis for document capture detection.

For the window size, we observed that a window size of smaller than 500 ms is less effective for detecting a change in orientation. We varied the window size using 100 ms increments. As the window size gets larger, the accuracy increases as well. There were drastic changes near 400 ms mark, where the accuracy was increased to 0.90, and it slowly converged to an accuracy of 0.98 (Fig. 13b). This is because an overly small window size was insufficient to capture the entire rotational movement of an orientation change. If the window size becomes larger than a certain level (i.e., 500 ms), then multiple movements are clumped together, which makes the movement difficult to be detected.

5.3. Tilt-based solution

To build a classification model for inferring the orientation mode, we used Weka 3.7 which is a well-known tool for machine learning and tested widely used classifiers, namely a decision tree (DT) and support vector machine (SVM), for activity recognition researches (Bao and Intille, 2004). For a DT, our model uses the C4.5 algorithm. For an SVM, our model utilizes a radial basis function (RBF) kernel for model training. To calculate the overall accuracy of each classifier, we conducted a ten-fold cross validation. We tested two feature sets introduced in Section 4.4 using these two classifiers. We also tested other learning models such as nearest neighbors and Naive Bayes, but did not observe any improvements over these algorithms Table 10.

The tilt-based solution lowered the error rate of landscape tasks by 81.6 percentage points (Table 11). Although the error rate slightly increased by 2.51 percentage points for portrait tasks (owing to the existence of sensing errors), the total error rate was still significantly



(a) Rotation threshold (window size of 2s, step size of 0.1s)

(b) Window size (threshold of 0.5s, step size of 0.1s)

Fig. 13. Parameter sensitivity analysis for rotation detection.

reduced by 38.44 percentage points.

We also examined the performance of each participant to check the user variance. During the evaluation, we applied a leave-one-subject-out model building and evaluation, where we trained a given machine learning model using the data of all participants except for one participant, and tested the model using the data of the omitted participant. We found that most of the errors were dependent on user-specific behavioral characteristics (Fig. 14). Interestingly, nine out of the 16 participants showed an accuracy of 100% accuracy. The result also showed that the classifier did not work well for several participants, namely P1 and P3. According to a gesture recognition study (Bulling et al., 2014), building a user-specific model can significantly improve the classification performance (i.e., data from a given user are used to train the model). Our cross-validation results showed no notable improvements. Our manual investigation of the dataset showed that the participants with a low accuracy were likely to be *highly precise* in balancing the device (i.e., almost perfectly parallel to the ground). For instance, the means of acc-X and acc-Y were almost zero for P1 (60% accuracy) as shown in Fig. 15. Users with a low accuracy (i.e., P3, P7, P11, P12, P13, and P16) showed a similar pattern. On the other hand, the patterns from the remaining participants showed a clear distinction between portraits and landscapes.

The best performing combination was an SVM with binary features, which reached an accuracy of 89.7%, although the remaining combinations showed accuracy levels reaching nearly 90% too (Table 11). It is worth noting that the classifier with binary features performed better than the classifier with numerical features, which may be owing to the fact that binary features are less susceptible to an overfitting.

Table 10
Comparison of error rates without and with the use of the rotation-based method.

Document	w/o ScanShot	w/ ScanShot	Diff.
Portrait	5.62%	0.0%	−5.62
Landscape	94.10%	1.25%	−92.85
Average	48.75%	0.63%	−48.12

Table 11
Comparison of error rates without and with tilt-based solution.

Document	w/o ScanShot	w/ ScanShot	Diff.
Portrait	5.62%	8.13%	+2.51
Landscape	94.10%	12.50%	−81.60
Average	48.75%	10.31%	−38.44

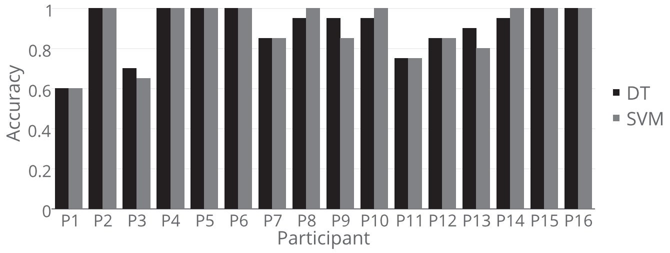


Fig. 14. Classification accuracy by participant (tilt-based solution).

Overall, our results showed that the proposed rotation-based solution can fix most errors and is superior to the tilt-based solution in terms of performance. Nonetheless, several cases exists in which the tilt-based solution has an advantages over the rotation-based solution. First, it does not need to collect the sensor data before launching the camera app. Recall that the rotation-based solution needs to monitor the sensor value prior to launching the camera app in order to achieve a high level of accuracy. Second, it can also be applied to mobile devices without the need of a gyroscope sensor. Smart watches with a camera (e.g., Samsung Gear 2) or a conventional digital camera (e.g., GoPro) may not have a gyroscope installed. In this case, our tilt-based solution can be an alternative solution (Table 12).

6. Discussion

In this paper, we proposed the use of ScanShot, which identifies the moment of document capture and corrects any orientation errors. For document capture, we propose the application of a simple accelerometer-based detector, and for rotation correction, we propose approaches, namely rotation event detection and micro-titling detection. Our experiment showed that these approaches can reduce rotation errors significantly.

Because our experiment only considered in-lab conditions, we will next discuss whether ScanShot is generalizable for various situations. We will present additional test results under a more realistic setting, and discuss the generalizability of ScanShot. Next, we will illustrate how our method can be integrated into gravity-based orientation management schemes of current smartphones. Finally, we will discuss the design implications derived from our study. Specifically, we will discuss 1) various application scenarios of document capture detection, 2) the diversity of hand grips applied during document capture and the potential issues for designing a mobile interaction system, and 3) the recognizability issues of UI indicators in smartphone camera's viewfinders.

6.1. Generalizability issues

There are additional factors that may affect document capture such as the initial location of the device, the possibility of taking a series of document photographs or alternating between regular use and photographing documents, and variations in the software and hardware. We will discuss the potential impacts of these factors.

First, the initial location of the device will not significantly influence the performance of our algorithm because ScanShot updates the

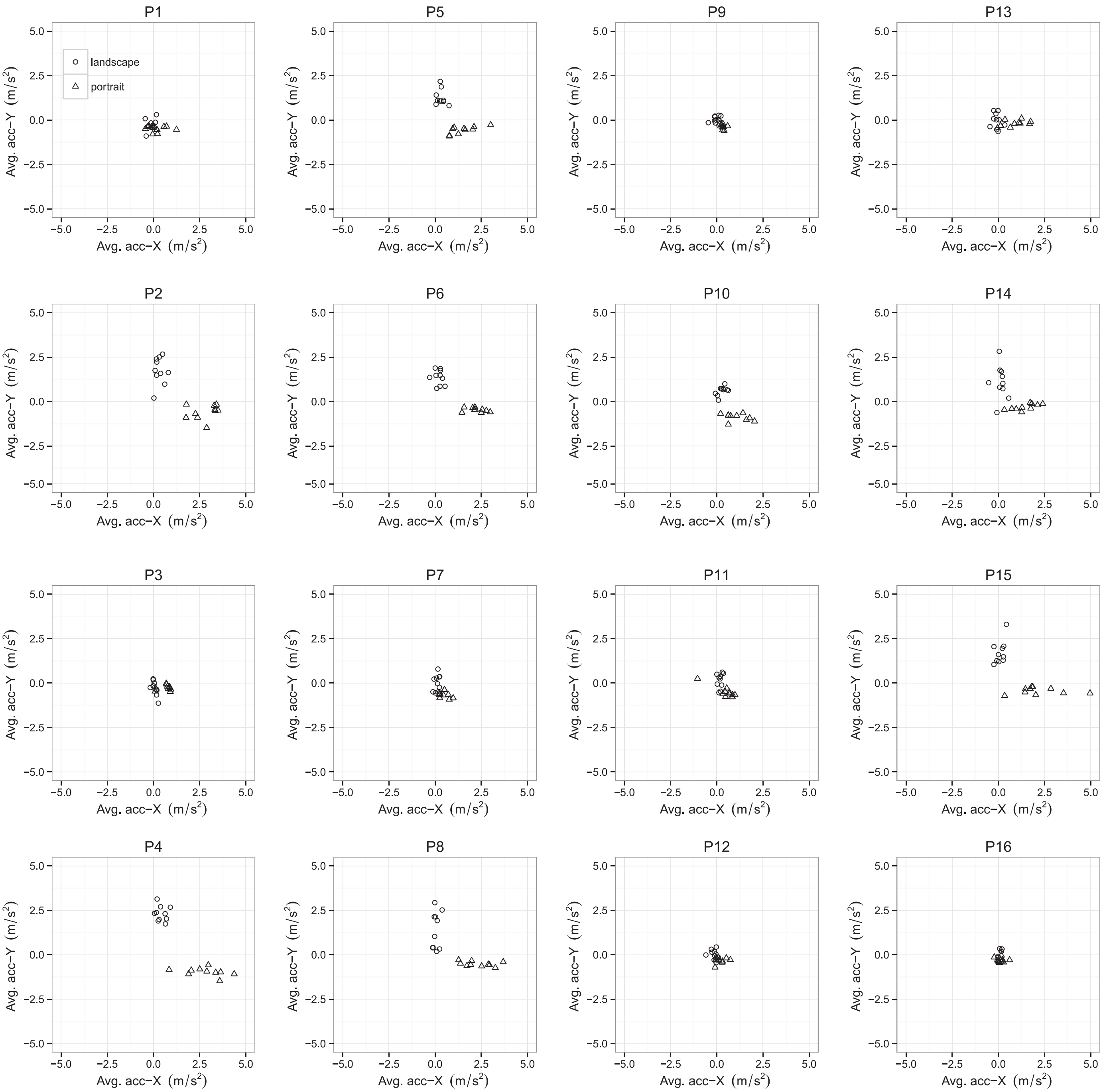


Fig. 15. Distribution of acc-X and acc-Y during capture state.

orientation only when document capture is observed just before the shutter button is pressed. ScanShot may fail to correct the orientation under a certain contrived scenario, e.g., although the default orientation of the smartphone at the time the app is launched (or when the smartphone is turned on) is portrait mode, its actual orientation may be in a different orientation mode, say, landscape mode. In our experiment, such as case was rarely observed. For the micro-tilt detection, this type of mismatch problem does not affect the performance because the detection relies solely on the accelerometer's values at the time of document capture.

Taking consecutive shots does not have a significant influence on how ScanShot operates. While tracking the rotations using a gyroscope, ScanShot continuously tracks every rotation event, and we can thus always have the up-to-date orientation. For micro-tilt behavior detection, such behavior is observed every time a user captures a document. This means that micro-tilt detection is not likely to be affected by the taking of consecutive photographs. Thus, the accuracy when taking consecutive shots may be comparable to that involving the taking of individual photographs.

The diversity of the mobile devices may also affect our algorithms. We tested the camera apps on several platforms, namely, iPhone 5 and 6 and Galaxy S4 and S5, to see whether they would show the same behavior as the Nexus 5 used for our study. We used the default camera app pre-installed on each device. For the iPhone 5 and 6, the default orientation was portrait mode, and the orientation patterns were similarly to those observed in the Nexus 5. However, for Galaxy S4 and S5, the camera apps work differently. Interestingly, the default orientation of the camera apps was the one previously used by the

Table 12
Classification accuracy.

Classifier	Numeric features	Binary features
SVM	89.4%	89.7%
DT	86.9%	88.8%

camera app. This means that the default orientation may not be portrait mode. For example, if the user closed the camera app in landscape mode, its default mode will also be landscape mode. Our rotation-based solution tracks the rotation movement from an incorrect initial position. We can easily handle this case by simply overriding the default orientation of the camera app and use the system's default orientation when document capture moment is detected.

We additionally conducted small experiments under different settings than those used during our in-lab study in order to make sure that our proposed method generally works in realistic environment without any issues concerning the generalizability of ScanShot. We recruited three participants whose age are 32, 27, 24 respectively (mean: 27.67, and std.: 4.04) and asked them to perform a capturing task under more naturalistic settings. We tested two different conditions: i.e., a different initial location where the user has to take the phone out of their pocket, and a different document placement where the document was rotated by the right 30°. Under each condition, the participants were asked to capture two landscape and two portrait documents. They were asked to capture eight documents in total. Our results show that ScanShot works consistently under both conditions, as shown in Table 13. For generalizability-related issues, we found two cases of errors in our rotation-based solution. The first case occurred during landscape document capture where the user had already rotated their phone to landscape mode before turning on a device. Our rotation based solution failed to detect this movement because the rotation was finished before the sensor monitoring had a chance to start. The second case occurred because there was insufficient rotation to the left because the documents initial position was rotated toward the right. For the tilt-based solution, we did not observe any notable patterns of errors differing from those during our in-lab study.

6.2. Integrating scanShot with existing systems

Hinckley et al. (2000) studied how accelerometer-based tilt sensing can be used for an automatic adjustment of the orientation. In practice, we can easily detect a left-right tilt as well as a forward-backward tilt of a mobile phone. For example, a portrait mode has +90° of forward-backward tilt, and zero degrees of left-right tilt. Likewise, a landscape mode has −90° of left-right tilt, and +90° of forward-backward tilt. In their work, the gray zones are ±5° dead bands that prevent jitter; the tilt angles must pass the gray region in order to change the display orientation. Interestingly, the device is intended to rest in *flat* mode when the tilt angles fall within ±3°, and the device does not change the display orientation.

ScanShot supplements a flat and gray range (±5.0°) in which the system is unaware of the current orientation. A document capture is detected only with a tilting threshold angle from the flat plane within ±12.43°, which corresponds to a mean acceleration-Z value within the range of [9.45, 10.5], as described in Section 4. For the range of 5–12.43°, the system designer can decide which method should have priority (e.g., ScanShot has a higher priority when a camera app is running).

We also measured Android's tilt threshold for changes in orientation to check the co-existence with our solutions. According to our measurement using a Nexus 5 phone, which was used in our experiment, the threshold values in portrait and landscape modes were measured to be 18.40° and 23.32°, respectively. These tilt thresholds for changes in orientation are much greater than the thresholds used by our algorithms. Thus, ScanShot can co-exist with Android's orientation management schemes without any conflicts. Consequently, ScanShot can be seamlessly integrated into an OS's orientation management scheme. It is worth noting that combining the two proposed methods could compensate each other. The rotation based method might work better in the situation that time for photo taking is too short to observe tilting after moving toward the subject. The tilt based method would not work well for those users who have a usage habit of maintaining strict balance. Depending on photo taking time and tilting habits, we could selectively enable one of the methods.

Table 13
Error rate for generalizability test.

Condition	Initial location (in the pocket)	Rotated document (30°)	Average
Default	54.17%	66.67%	60.42%
Rotation	8.33%	8.33%	8.33%
Micro-tilt	12.50%	12.50%	12.50%

6.3. Design implications

6.3.1. Context-aware services for document capture

Detecting document capture intention enables novel context-aware services for document capture such as document photograph management, and adaptive user interaction support. In document photograph archival scenarios, mobile devices can handle mobile document capture events for document management separately. Existing image classification techniques are mostly based on computer-vision methods applied to infer semantic meaning from photographs, by analyzing the content (e.g., color and texture distribution) and meta-data (e.g., exposure time and flash) (Szummer and Picard, 1998; Vailaya et al., 1999). ScanShot supplements these computer vision approaches in that we can fairly accurately infer a users document capture intention at minimal cost. Another promising application is adaptive user interaction support. By recognizing a user's intention we can adaptively tailor a camera's viewfinder screen specifically for document capture. For example, the default camera app can adaptively present different UI menus for document capture such as document cropping, document contrast control, and document boundary detection. Furthermore, the taken photographs can be automatically delivered to document-related applications (e.g., opening a document management app or a note app). Compared with more general vision based methods, our accelerometer-based intention inference methods require considerably fewer resources (energy and processing), and it can be applicable to any digital camera devices with an accelerometer. Thus, its potential applications are quite wide.

6.3.2. Diversity of hand grip positions for mobile interactions

The hand postures of mobile devices are one of the important contextual factors for mobile interaction. Prior studies have shown that hand posture and its related information (e.g., the number of hands and fingers) has a critical impact on mobile device usage (e.g., interaction performance) (Wobbrock et al., 2008). In addition, we argue that hand positions are strongly related with the design of the user interface components on a touch screen device such as the button placement. Researchers have previously explored a number of techniques for sensing the hand position and its related measurements (e.g., device orientation, sensing pressures imposed on a touch screen, and customized interaction techniques based on grip) (Taylor and Bove, 2009; Goel et al., 2012; Wimmer and Boring, 2009; Hwang and Bianchi, 2013). Yet, these studies have only explored typical hand grips and have not considered hand grips that are related to a camera app. For example, Kim et al. (2006) built a hand-grip classifier for various applications (e.g., calling, sending text messages, camera use, watching videos, and playing games), but only a few hand grips were considered. According to our study, we found that hand grips for camera manipulation are much more diverse than those reported in previous studies. For example, we found that at least there are eight hand grips for mobile camera interaction. Camera apps seem to have more variations than other apps because users should not block the camera lens on the back of the mobile device, and should maintain a stable posture for capturing clear images. For example, users are likely to grasp the side of the device instead of the back where the lens is located. Therefore, our results show that different hand grips should be further studied by considering various application contexts (e.g., camera, SMS), hardware characteristics (e.g., external buttons, lens), and the screen sizes. Additionally, we assumed that user variations (e.g., hand size and height), and environmental factors (e.g., the document size) may affect the behavior.

6.3.3. Increasing awareness of peripheral UI indicators

While capturing a document using a camera app, we found that users rarely recognized the current orientation mode of the device. As shown in Fig. 16, a camera icon is typically used as the orientation indicator. In Fig. 16, the device shown is currently turned to landscape mode, but the icon shows that it is actually in portrait mode, i.e., the system failed to capture the change in orientation. To our surprise, only a few of the users recognized the orientation change. The majority of our participants did not recognize the orientation errors despite the fact that the information was presented in the orientation indicator UI. This clearly shows that the shape and arrangement of the indicator UI components were ineffective at delivering this information while the user interacted with the viewfinder.

Theories on visual attention provide possible explanations for this observation. Our sight consists of central and peripheral vision. Central vision provides much better resolution than peripheral vision (Johnson, 2010). Thus, the spatial resolution of our visual recognition decreases significantly from the center to the edges (Lindsay and Norman, 2013). Our low-resolution peripheral vision exists mainly to guide our central vision to visit the interesting parts of our visual field.

While the users are capturing a photograph, their central vision is primarily on the object in the preview screen (e.g., moving around to capture the area of interest and touching the screen to re-focus the camera). The UI indicator for the current orientation is located in the user's peripheral vision and likely to fail to draw the user's attention for several reasons. Because the user's focused on their central vision for the capturing tasks, and because our peripheral vision has a low resolution, the visual indicators should be salient enough to draw the user's attention. The current indicator simply rotates the black or gray camera icon slowly, which is rarely recognized by the user. Designers are recommended to employ clearer visual cues. For example, we can have a blinking indicator to show the current orientation. Alternatively, we can display an overlay guide on top of the central vision area, and we can show a paper-shaped guide such that users can easily recognize the current orientation while a capturing document. Considering these cognitive issues, there should be further study on how to design UI components indicating the system's status (e.g., orientation changes) while users are interacting with other parts of the screen (e.g., a preview screen).

6.4. Limitations and future work

For online survey result, there could be undiscovered categories of information capture because our respondents were recruited from online university community. However, we believe the most of use cases were covered in our survey because university students are usually an early adaptor group for technologies.

Although studies regarding information capture using the camera phone, such as the general motivation for producing digital photographs, have been conducted (Lux et al., 2010; Kindberg et al., 2004, 2005), little is known regarding user behavior, especially in the context of document capture using a camera. In addition to online and alb study result, there should be further studies on understanding the entire process of information capture, ranging from the motivation of document capture, and the management of captured documents.

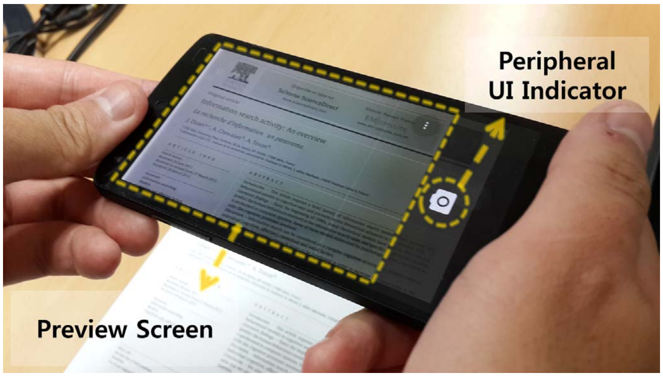


Fig. 16. Camera app UI showing an incorrect orientation when a user is capturing a landscape document.

For lab study, further experiments with different factors (e.g., age, task, and platform) are required for a better understanding of the impact of these factors on the document capture behaviors even though we discussed the generalizability issue. Because ScanShot can work with existing camera apps, we will add ScanShot to the various app stores. This kind of in-the-field experiment will bring about more insight into the generalizability issue.

7. Conclusion

We analyzed the problem of orientation errors in document capture using a smartphone camera. We investigated the error rates of the orientation error, the hand grips used for capturing a document, and the skew angle of captured documents. Based on the user study, we proposed ScanShot, which automatically detects the document capture to update orientation change. ScanShot supports these features solely with the use of built-in motion sensors, namely an accelerometer and gyroscope. Our evaluation showed that the rotation-based method and the tilt-based solution reduce the error rate by 92.85 and 82.60 percentage points respectively. We discussed the generalizability and integration issues of our proposed methods and design implications including context-aware services for document capture, the diversity of hand grips used, and increasing awareness of camera UI indicators.

Acknowledgements

This work was partly supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. B0101-15-1272, Development of Device Collaborative Giga-Level Smart Cloudlet Technology), and by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Korea government (MSIP) (No. NRF-2015R1D1A1A01059497).

References

Ahmed, S., Kise, K., Iwamura, M., Liwicki, M., Dengel, A., 2013. Automatic ground truth generation of camera captured documents using document image retrieval. In: Proceedings of the 2013 12th International Conference on Document Analysis and Recognition. pp. 528–532.

Bao, L., Intille, S. S., 2004. Activity recognition from user-annotated acceleration data. In: Proceedings of the International Conference on Pervasive Computing. Springer, pp. 1–17.

Beyer, H., Holtzblatt, K., 1997. Contextual Design: Defining Customer-centered Systems. Elsevier, USA.

Brown, B.A.T., Sellen, A.J., O'Hara, K.P., 2000. A diary study of information capture in working life. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI'00. ACM, New York, NY, USA, pp. 438–445, ⟨http://doi.

acm.org/10.1145/332040.332472⟩.

Bulling, A., Blanke, U., Schiele, B., 2014. A tutorial on human activity recognition using body-worn inertial sensors. ACM Comput. Surv., 46(3), 33:1–33:33, ⟨http://doi.acm.org/10.1145/2499621⟩.

Cheng, L.-P., Hsiao, F.-I., Liu, Y.-T., Chen, M. Y., 2012. iRotate: automatic screen rotation based on face orientation. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI'12. ACM, New York, NY, USA, pp. 2203–2210,⟨http://doi.acm.org/10.1145/2207676.2208374⟩.

Cheng, L.P., Lee, M.H., Wu, C.Y., Hsiao, F.I., Liu, Y.T., Liang, H.S., Chiu, Y.C., Lee, M.S., Chen, M.Y., 2013. iRotateGrasp: automatic screen rotation based on grasp of mobile devices. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI'13. ACM, New York, NY, USA, pp. 3051–3054, ⟨http://doi.acm.org/10.1145/2470654.2481424⟩.

Doermann, D., Liang, J., Li, H., 2003. Progress in camera-based document image analysis. In: Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on, vol. 1, pp. 606–616, ⟨http://dx.doi.org/10.1109/ICDAR.2003.1227735⟩.

Goel, M., Wobbrock, J., Patel, S., 2012. Gripsense: using built-in sensors to detect hand posture and pressure on commodity mobile phones. In: Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology. UIST'12. ACM, New York, NY, USA, pp. 545–554, ⟨http://doi.acm.org/10.1145/2380116.2380184⟩.

Gye, L., 2007. Picture this: the impact of mobile camera phones on personal photographic practices. Cont.: J. Media Cult. Stud., 21(2), 279–288, ⟨http://www.tandfonline.com/doi/abs/10.1080/10304310701269107⟩.

Hinckley, K., Pierce, J., Sinclair, M., Horvitz, E., 2000. Sensing techniques for mobile interaction. In: Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology. UIST'00. ACM, New York, NY, USA, pp. 91–100, ⟨http://doi.acm.org/10.1145/354401.354417⟩.

Hinds, S.C., Fisher, J.L., D'Amato, D.P., 1990. A document skew detection method using run-length encoding and the hough transform. In: Pattern Recognition, 1990. Proceedings. 10th International Conference on., vol. 1. IEEE, pp. 464–468, ⟨http://dx.doi.org/10.1109/ICPR.1990.118147⟩.

Hwang, S., Bianchi, A., Wohn, K.y., 2013. Vibpress: estimating pressure input using vibration absorption on mobile devices. In: Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services. MobileHCI'13. ACM, New York, NY, USA, pp. 31–34, ⟨http://doi.acm.org/10.1145/2493190.2493193⟩.

Johnson, J., 2010. Designing With the Mind in Mind: Simple Guide to Understanding User Interface Design Rules. Morgan Kaufmann, USA, 65–77.

Kim, K.E., Chang, W., Cho, S.J., Shim, J., Lee, H., Park, J., Lee, Y., Kim, S., 2006. Hand grip pattern recognition for mobile user interfaces. In: Proceedings of the 18th Conference on Innovative Applications of Artificial Intelligence – Volume 2. IAAI'06. AAAI Press, pp. 1789–1794, ⟨http://dl.acm.org/citation.cfm?Id=1597122.1597138⟩.

Kindberg, T., Spasojevic, M., Fleck, R., Sellen, A., 2004. How and why people use camera phones. HP Lab. Tech. Rep. HPL, 2004–2216.

Kindberg, T., Spasojevic, M., Fleck, R., Sellen, A., 2005. The ubiquitous camera: an in-depth study of camera phone use. Pervasive Comput. IEEE, 4(2), 42–50, ⟨http://dx.doi.org/10.1109/MPRV.2005.42⟩.

Kunze, K., Lukowicz, P., Partridge, K., Begole, B., 2009. Which way am i facing: inferring horizontal device orientation from an accelerometer signal. In: Wearable Computers, 2009. ISWC'09. International Symposium on. pp. 149–150, ⟨http://dx.doi.org/10.1109/ISWC.2009.33⟩.

Kwag, H., Kim, S., Jeong, S., Lee, G., 2002. Efficient skew estimation and correction algorithm for document images. Image Vision Comput., 20(1), 25–35, ⟨http://dx.doi.org/10.1016/S0262-8856(01)00071-3⟩.

Le, D.S., Thoma, G.R., Wechsler, H., 1994. Automated page orientation and skew angle detection for binary document images. Pattern Recog., 27(10), 1325–1344, ⟨http://dx.doi.org/10.1016/0031-3203(94)90068-X⟩.

Lee, J., Ju, D.Y., 2013. Defying gravity: a novel method of converting screen orientation. Int. J. Smart Home. 7(5), 83–90, ⟨http://dx.doi.org/10.14257/ijsh.2013.7.5.09⟩.

Lindsay, P.H., Norman, D.A., 2013. Human Information Processing: An Introduction to Psychology. Academic Press, USA.

Lu, S., Wang, J., Tan, C.L., 2007. Fast and accurate detection of document skew and orientation. In: Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on., vol. 2. pp. 684–688, ⟨http://dx.doi.org/10.1109/ICDAR.2007.4377002⟩.

Lux, M., Kogler, M., del Fabro, M., 2010. Why did you take this photo: a study on user intentions in digital photo productions. In: Proceedings of the 2010 ACM Workshop on Social, Adaptive and Personalized Multimedia Interaction and Access. SAPMIA'10. ACM, New York, NY, USA, pp. 41–44, ⟨http://doi.acm.org/10.1145/1878061.1878075⟩.

Mizell, D., 2003. Using gravity to estimate accelerometer orientation. In: Wearable Computers, 2003. Proceedings. Seventh IEEE International Symposium on. pp. 252–253, ⟨http://dx.doi.org/10.1109/ISWC.2003.1241424⟩.

Oh, J., Choi, W., Kim, J., Lee, U., 2015. Scanshot: detecting document capture moments and correcting device orientation. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. CHI'15. ACM, New York, NY, USA, pp. 953–956, ⟨http://doi.acm.org/10.1145/2702123.2702440⟩.

Okabe, D., 2006. Everyday contexts of camera phone use: steps toward techno-social ethnographic frameworks. In: Mobile Communications in Everyday Life: Ethnographic Views, Observations, and Reflections. MIT Press, pp. 79–102.

Szummer, M., Picard, R., 1998. Indoor-outdoor image classification. In: Content-Based Access of Image and Video Database, 1998. Proceedings, 1998 IEEE International Workshop on. pp. 42–51, ⟨http://dx.doi.org/10.1109/CAIVD.1998.646032⟩.

Taylor, B.T., Bove, Jr., V.M., 2009. Graspables: grasp-recognition as a user interface. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI'09. ACM, New York, NY, USA, pp. 917–926, ⟨http://doi.acm.org/10.1145/1518701.1518842⟩.

Vailaya, A., Figueiredo, M., Jain, A., Zhang, H., 1999. Content-based hierarchical classification of vacation images. In: Multimedia Computing and Systems, 1999. IEEE International Conference on, vol.1. pp. 518–523, ⟨http://dx.doi.org/10.1109/MMCS.1999.779255⟩.

Wimmer, R., Boring, S., 2009. Handsense: discriminating different ways of grasping and holding a tangible user interface. In: Proceedings of the 3rd International Conference on Tangible and Embedded Interaction. TEI'09. ACM, New York, NY, USA, pp. 359–362, ⟨http://doi.acm.org/10.1145/1517664.1517736⟩.

Wobbrock, J.O., Myers, B.A., Aung, H.H., 2008. The performance of hand postures in front- and back-of-device interaction for mobile computing. Int. J. Hum.-Comput. Stud., 66(12), 857–875, (mobile human-computer interaction), ⟨http://dx.doi.org/10.1016/j.ijhcs.2008.03.004⟩.