ScanShot: Detecting Document Capture Moments and Correcting Device Orientation

Jeungmin Oh, Woohyeok Choi, Joohyun Kim, Uichin Lee
Department of Knowledge Service Engineering, KAIST
{jminoh, woohyeok.choi, joohyun.kim, uclee}@kaist.ac.kr

ABSTRACT
Document capturing with smartphone cameras is performed increasingly often in our daily lives. However, our user study results (n=10) showed that more than 80% of landscape tasks had incorrect orientations. To solve this problem, we systematically analyzed user behavior of document capturing and proposed a novel solution called ScanShot that detects document capturing moments to help users correct the orientation errors. ScanShot tracks the gravity direction to capture document capturing moments, analyzes logged gyroscope data to automatically update orientation changes, and provides visual feedback of the inferred orientation for manual correction. Our user study results (n=20) confirmed that capturing moments can be recognized with accuracy of 97.5%, our update mechanism can reduce the orientation errors by 59 percentage points.

Author Keywords
Device orientation; automatic rotation; document capturing; smartphone camera

ACM Classification Keywords
H.5.2. [User Interface]: Input devices and strategies.

INTRODUCTION AND RELATED WORK
People use the built-in camera on a smartphone for various reasons, ranging from personal reflection to social experience and functional tasks [5, 8]. Our work considers the action of document capturing that occurs increasingly often in our daily lives (e.g., capturing a magazine or news article) [1].

Document capturing is typically done by configuring the angle of a smartphone camera to a top-down view (or bird's-eye view). However, orientation errors are often observed in the captured images. We discovered that this problem orginates from the phone's inferred orientation being different from the user's capturing orientation (hand posture). Smartphones typically have four orientation modes in 2D space (just as in a picture frame on a wall), namely portrait, upside down, landscape left, and landscape right (see Figure 1). For a given capturing orientation, there are three incorrect modes that cause orientation errors from the top-down angle.

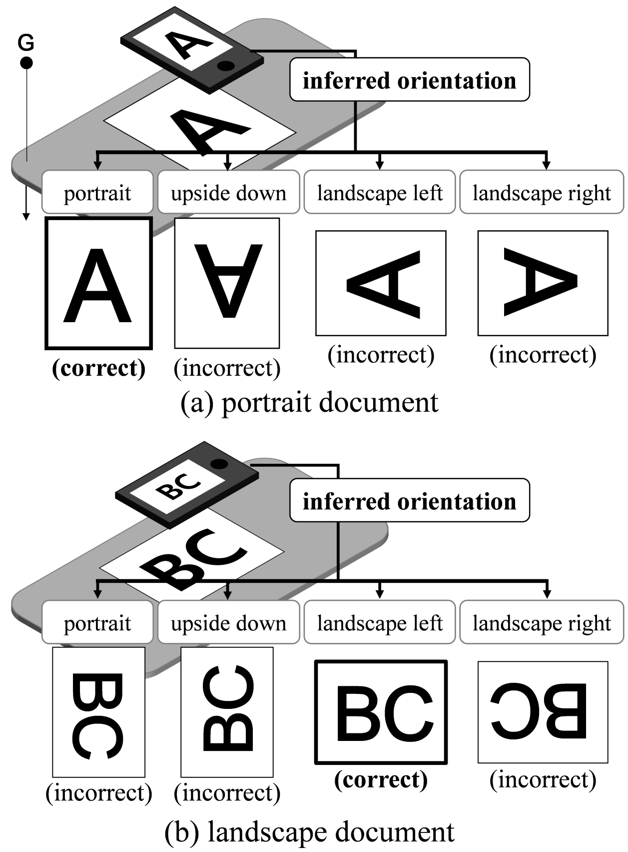


Figure 1: Orientation problem in document capturing

Why do such orientation errors happen? Smartphones typically use gravity-based screen rotation algorithms (e.g., using an accelerometer), assuming that users are standing or sitting upright while interacting with the device in their hands [7, 4]. That is, for orientation inference, the algorithms mainly consider two axes (X and Y axis) of the gravity vector. However, orientation inference fails to work properly if smartphones are close to the plane that is parallel to the ground, when the X and Y axes of the gravity are close to zero (e.g., taking a top-

CHI 2015, April 18 - 23 2015, Seoul, Republic of Korea.

ACM 978-1-4503-3145-6/15/04 $15.00
http://dx.doi.org/10.1145/2702123.2702440

down shot, or placing a phone on a table). If we rotate the phone parallel to the ground while taking a top-down shot, the phone's inferred orientation remains the unadjusted, and thus, an erroneous inferred orientation is obtained.

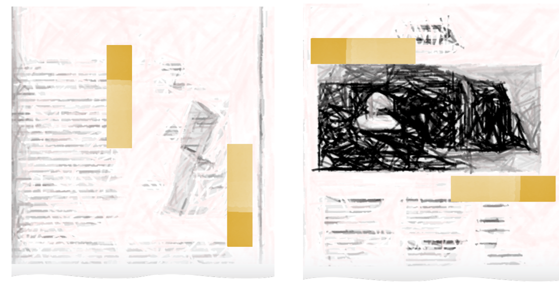
In the HCI literature, none of the earlier works have studied this problem systematically. Instead researchers have mainly focused on studying how to prevent unwanted screen rotation when users change their body posture (e.g., lying down). For example, a user's front facing camera was used to detect face orientation that is often consistent with the correct orientation [2]. Researchers showed that another good indicator is how the phone is grasped [6]. Cheng et al. demonstrated its usefulness by prototyping a touch-sensor based phone case (achieving about 80% inference accuracy) [3]. However, it is difficult to apply such techniques to top-down shots: a user's face cannot be captured consistently with this camera angle and the manner of grasping a phone is significantly different to regular smartphone use. For stable document capturing, users typically use both hands. Alternatively we can use optical character recognition (OCR) to detect the document's orientation for automatic correction, but its performance is heavily influenced by context (e.g., font, handwriting, language, and light conditions), and it poses significant processing overhead when compared to motion data processing.

The goal of this work is to systematically understand document capturing behavior in top-down shots and to design solutions that can mitigate orientation errors. We first perform a user study (n=10) to quantify the severity of orientation errors and analyze user behavior in typical document capturing tasks. We found that the error rates were surprisingly high particularly in the landscape document capturing (capture errors: portrait = 5% vs. landscape = 82%). Furthermore, nine out of ten participants were unaware of the concept of inferred orientation, thus failing to recognize orientation signifiers in a camera app (i.e., icon rotation).

Our task analysis of document capturing inspired us to propose ScanShot, a novel method to detect document capturing moments and fix the erroneous orientations. Our approach is composed of three steps. ScanShot first detects a document capturing moment (when a phone's body becomes parallel to the ground) by monitoring the gravity direction using an accelerometer. Secondly, when the document capture moment is detected, ScanShot attempts to automatically update the orientation by analyzing recorded gyroscope data. Finally, ScanShot provides visual feedback where a translucent paper icon is overlaid on the screen to visualize inferred orientation. Our user study results (n=20) showed that document capturing moments can be detected with accuracy of 97.5% and automatic rotation achieves accuracy of 82.16%. In addition, the participants noted the usefulness of visual feedback.

DOCUMENT CAPTURE BEHAVIOR ANALYSIS

Study Setup: We recruited 10 smartphone users who had experience in document capturing. Their ages ranged from 23 to 31 (mean: 26.80, SD: 2.39). Half of them were males, and there was one left-handed participant (P5). To understand document capture behavior, we carefully selected 20 parts in a National Geographic magazine. As shown in Figure 2, we



(a) Portrait document (b) Landscape document

Figure 2: Two types of documents to capture for experiment

highlighted the capture area using sticky notes. The sizes of chosen area ranged from a small portions of the magazine to entire pages. Two types of capture areas were investigated. Areas with large width corresponded to landscape tasks, and those with significant height corresponded to portrait tasks. We asked participants to always place the phone on the table after each capturing task so that every capture is treated as an individual task. We used Nexus 5, the current Android's reference phone and its pre-installed camera app in Android 4.4.4. Before this task, the participants engaged in a training session in which they took shots in order to become familiarized with the device and its camera app. After the experiment, the participants were asked if they were aware of the concepts of inferred orientation and rotation signifiers in a camera app (i.e., icon rotation). All sessions were videotaped for a subsequent behavioral analysis. Each participant was compensated with a gift card equivalent to $5.

Frequency of Orientation Errors: The error rates in the landscape and portrait tasks were 0.82 (SD: 0.27) and 0.05 (SD: 0.07), respectively. The landscape capture tasks had a significantly larger number of errors. The portrait capture tasks had only a 5% error rate because the portrait orientation is the Android device's default orientation. By examining video recordings and recorded sensor data, we found that the participants tended to hold the device orthogonal to gravity, which causes the gravity-based rotation system to be disorientated [4]. In Table 1, we present each participant's error rates in both portrait and landscape tasks. P7 and P10 showed relatively low error rates in the landscape tasks since both users tended to tilt the device toward their bodies to look at the preview images on the screen; P10 tilted more frequently than P7. For those two participants, gravity-based rotation sometimes worked, but we found that the majority of the participants had higher error rates due to erroneous orientation.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Portrait	.0	.0	.1	.0	.1	.1	.0	.2	.0	.0
Landscape	1	1	.9	1	1	.9	**.6**	1	1	**.4**

Table 1: Ratio of orientation errors in capturing tasks

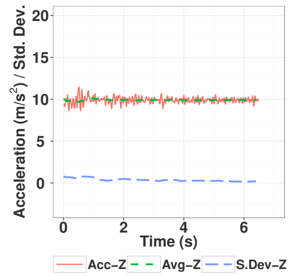
Most participants (9 out of 10) were not even aware of when and why the problem occured even though they were familiar with automatic screen rotation. The camera UI had visual indicators of the current orientation, but they did not notice that the camera's orientation also changed as they rotated. This may be due to the fact that users generally do not need to pay attention to the orientation problem in general photo taking as smartphone cameras typically perform well in this regard. Another factor of error comes from the user's failure to recognize changes in UI when the user's focal vision is on the target object on the preview screen during a top-down shot. The visual indicator is typically located in the peripheral screen area such as a rotating camera icon near the bezel frame.

Analysis of User Behavior: We investigated recorded videos in order to better understand document capturing behavior, which helped us design algorithms for automatic correction. The overall process consists of 1) launching a camera app, 2) adjusting orientation (optional), 3) composing a top-down shot (including zooming and panning), and 4) touching to focus (optional) and pressing a shutter button. In the case of landscape tasks, the participants adjusted the phone's orientation. Portrait capturing tasks do not require orientation adjustment because a phone's default orientation is the portrait mode (as it is the case of a smartphone's home screen orientation). Among four orientations, we observed only two orientation modes (portrait and landscape left) in the experiment. This pattern was consistent with that of the left-handed participant (P5). This important observation allowed us to reduce the number of possible states when designing an automatic algorithm. Note that this step generally came after launching the camera app, but we also observed that in a few landscape sessions, app launching was followed by device rotation. The participants then held their phones parallel to the plane where the document lies and tried zooming/panning to capture the target area. There were participants (P7, P10) who tended to tilt the device approximately up to 45° from the ground plane. Once the target area was determined, the participants touched to focus (occasionally) and pressed the shutter button to take a photo. In the landscape task, most participants grasped the phone with two hands. Likewise, in the portrait shot, two hands were typically used (sometimes one hand grasping the top, and the other hand the bottom).

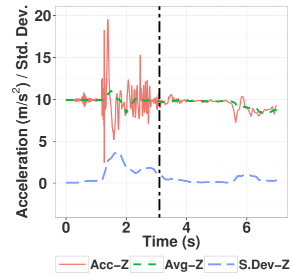
SCANSHOT DESIGN

Our preliminary user study helped us design ScanShot. 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 gyroscope data to fix any orientation changes (i.e., portrait to landscape posture). Finally, ScanShot displays a translucent paper icon overlaid on the screen to visualize the inferred orientation, which allows users to fix orientation errors if needed.

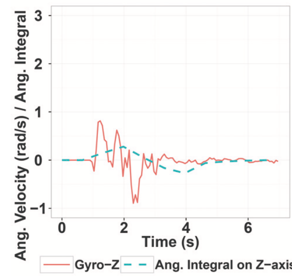
Detecting Document Capture Moments: To detect document capturing moments, we propose an accelerometer-based detection method. When a camera app launches, the accelerometer values fluctuate because of movements (switching to a top-down shot with zooming/panning). After a few seconds, the accelerometer values are soon stabilized because the user typically touches to focus and then presses the shutter button. When photo taking is about to happen, the accelerometer's Z-axis is close to the gravitational acceleration, or 9.8m/s^2.



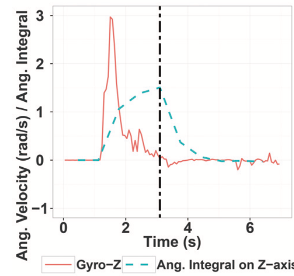
(a) Acceleration (portrait, no rotation)



(b) Acceleration (landscape, rotation)



(c) Angular value (portrait, no rotation)



(d) Angular value (landscape, rotation)

Figure 3: Accelerometer and gyroscope traces in the capturing tasks: portrait for (a),(c), and landscape for (b),(d) (vertical dotted lines in (c) and (d) represent the time mark that rotation has happened)

As shown in Figure 3, when taking a top-down shot, the participant's hands move with a maximum fluctuation value of ± 0.3. When a camera app is running, the document capture moment is detected when the Z-value is within the range from 9.5m/s^2 to 10.1m/s^2.

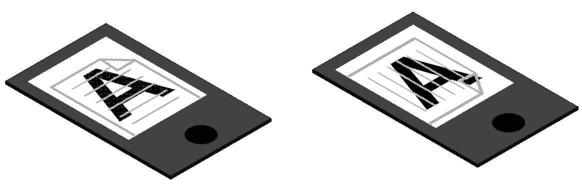
This mechanism prevents our algorithm from conflicting with the existing gravity-based orientation system because the Z-value of the accelerometer is nearly zero when taking a non-document photo. Thus, it is not influenced by the initial location of the device (e.g., on the desk or in the pocket).

Updating Orientation: We chose a gyroscope to detect an orientation change event because it can accurately sense device rotation. We used a moving window approach. For sensor data processing, the sliding window of size w seconds moved in time, with a segment size of s seconds. According to our user behavior analysis, rotating action typically took less than 2 seconds. Since the sampling rate was 15 Hz (one sample per 66ms), we set the segment size as $s = 0.66$s (10 samples per segment), and the window size as $w = 1.98$s (3 segments per window). For a given window, we integrated the Z-axis values of gyroscope samples. By carefully analyzing the data set, we set the threshold to detect a rotation event as $\pm$ 0.5, which corresponds to $\pm$ 28.6°. Thus, if this value was greater than 0.5, we classified it as a left rotation event; if it was lower than -0.5, it is classified as a right rotation event. Otherwise, we assumed that no rotation was made (see Figure 3). We used only two orientation modes (portrait and landscape left) for adjustment since these modes dominated in the user study.

```
SET adjustedOrientation to DEFAULT_ORIENTATION;
repeat
    if a new window is generated from sensor listeners then
        Calculate cummulative sum of gyroscope for the window;
        if the cummulative sum > 0.5 then
            adjustedOrientation = LANDSCAPE_LEFT;
        else if the cummulative sum < -0.5 then
            adjustedOrientation = PORTRAIT;
        end
        if 9.5 < avg. acc-Z < 10.1 and std. acc-Z < 0.3 then
            Use adjustedOrientation;
        else
            Use traditional gravity-based orientation;
        end
    end
until shutter button is pressed;
```

Algorithm 1: Pseudo code of ScanShot

Displaying Paper Overlay: According to the preliminary study, our participants generally did not notice orientation errors. When a top-down shot is detected, as a backup, ScanShot immediately provides a visual feedback by displaying a translucent paper icon overlaid on the screen (see Figure 4). For example, assuming that a user is taking a portrait shot, Figure 4(a) shows a correct overlay of a portrait shot. Figure 4(b) shows an incorrect orientation (landscape left). This design decision was made because when a user is taking a photo, the user's focal vision remains on the target object on the screen.



(a) Overlay with correct orientation (b) Overlay with incorrect orientation

Figure 4: Paper overlays for a portrait shot

EVALUATION

Setup: We recruited 20 participants to evaluate the performance of ScanShot. Their ages ranged from 22 to 34 (mean: 27.5, SD 3.25). The same tasks as the first experiment were given. Our ScanShot implementation took a user's input and automatically updated the orientation by modifying the output image file. Each participant was compensated with a gift card that is equivalent to $5.

Document	w/o ScanShot	w/ ScanShot	Diff.
Portrait	.08	.07	-.01
Landscape	.91	.32	-.59

Table 2: Error rate comparison between w/o and w/ ScanShot

Results: Our results showed that the accuracy of detecting document capture moments is 97.5%, thereby confirming the efficiency of our algorithm. We then tested the accuracy of detecting rotation events. Table 2 presents the detection accuracy. The error rate without ScanShot was similar to that of the previous study, and thus, we can assume that the nature of the two participant groups was similar. Overall, ScanShot fixed many instances of the orientation errors. In particular, there was significant decrement of error rates in the landscape tasks, from 91% to 32%. We manually studied failure instances and found that most errors were due to participants launching the camera app after rotation, which caused the algorithm to miss the sensor data corresponding to the rotation. This could be enhanced by continuously monitoring sensor prior to launching the app. The collected data are available online.[1]

CONCLUSION AND FUTURE WORK

We analyzed the problem of orientation errors in document capturing using smartphone camera. We proposed ScanShot, which automatically detects document capturing moment to automatically update orientation changes and then enables manual orientation correction with visual feedback. ScanShot supports these features solely with the use of built-in motion sensors, namely an accelerometer and gyroscope. Our user study results (n=20) showed that ScanShot detected capturing moments with 97.5% accuracy, and lowered rotation error rates to 32%. Our algorithm can be further improved by employing more sophisticated machine learning algorithms or adaptively changing window sizes.

ACKNOWLEDGEMENTS

This work was supported by ICT R&D program of MSIP/IITP [10041313, UX-oriented Mobile SW Platform].

REFERENCES

1. Brown, B. A., Sellen, A. J., and O'hara, K. P. A diary study of information capture in working life. In *Proc. CHI* (2000).
2. Cheng, L.-P., Hsiao, F.-I., Liu, Y.-T., and Chen, M. Y. iRotate: automatic screen rotation based on face orientation. In *Proc. CHI* (2012).
3. Cheng, L.-P., Lee, M. H., Wu, C.-Y., Hsiao, F.-I., Liu, Y.-T., Liang, H.-S., Chiu, Y.-C., Lee, M.-S., and Chen, M. Y. iRotateGrasp: automatic screen rotation based on grasp of mobile devices. In *Proc. CHI* (2013).
4. Hinckley, K., Pierce, J., Sinclair, M., and Horvitz, E. Sensing techniques for mobile interaction. In *Proc. UIST* (2000).
5. Kindberg, T., Spasojevic, M., Fleck, R., and Sellen, A. The ubiquitous camera: an in-depth study of camera phone use. *IEEE Pervasive Computing* (2005).
6. Lee, J., and Ju, D. Y. Defying gravity: a novel method of converting screen orientation. *IJSH* (2013).
7. Sahami Shirazi, A., Henze, N., Dingler, T., Kunze, K., and Schmidt, A. Upright or sideways?: analysis of smartphone postures in the wild. In *Proc. MobileHCI* (2013).
8. Sellen, A., Fleck, R., Kindberg, T., and Spasojevic, M. How and why people use camera phones. *Microsoft Technical Report* (2004).

[1] `https://github.com/ohtangza/chi2015-scanshot`