iRotateGrasp: Automatic Screen Rotation based on Grasp of Mobile Devices

Lung-Pan Cheng, Meng-Han Lee, Che-Yang Wu, Fang-I Hsiao, Yen-Ting Liu, Hsiang-Sheng Liang, Yi-Ching Chiu, Ming-Sui Lee, Mike Y. Chen
Department of Computer Science
National Taiwan University
Taipei, Taiwan 10617
{r98922168, r00922006, b98902113, b96902005, b97901121, r01922036, b99901146, mslee, mikechen}@csie.ntu.edu.tw

ABSTRACT
Automatic screen rotation improves viewing experience and usability of mobile devices, but current gravity-based approaches do not support postures such as lying on one side, and manual rotation switches require explicit user input. iRotateGrasp automatically rotates screens of mobile devices to match users' viewing orientations based on how users are grasping the devices. Our insight is that users' grasps are consistent for each orientation, but significantly differ between different orientations. Our prototype used a total of 44 capacitive sensors along the four sides and the back of an iPod Touch, and uses support vector machine (SVM) to recognize grasps at 25Hz. We collected 6-users' usage under 108 different combinations of posture, orientation, touchscreen operation, and left/right/both hands. Our offline analysis showed that our grasp-based approach is promising, with 80.9% accuracy when training and testing on different users, and up to 96.7% if users are willing to train the system. Our user study (N=16) showed that iRotateGrasp had an accuracy of 78.8% and was 31.3% more accurate than gravity-based rotation.

ACM Classification: H.5.2 [User Interfaces]: Input devices and strategies, Interaction styles; H.1.2 [User/Machine Systems]: Human factors

Keywords: Auto rotation; grasp recognition; mobile device; adaptive user interface; device orientation

INTRODUCTION AND RELATED WORK
Modern mobile devices, such as the iPhone, iPad, Android phones, and tablets, support automatic screen rotation in order to improve the viewing experience and usability. Various types of sensors have been used for gravity-based screen rotation for mobile devices, including mercury switches [11] and 2-axis accelerometers [1,7].

Current gravity-based approaches assume that users are standing or sitting upright while using the devices, which causes the screen to rotate incorrectly when users are in near horizontal postures, such as when lying down on one side. A previous survey of this approach (N=513) [4] shows that 91% of the respondents have experienced incorrect automatic rotation, with 42% of the respondents encountering the problem several times a week.

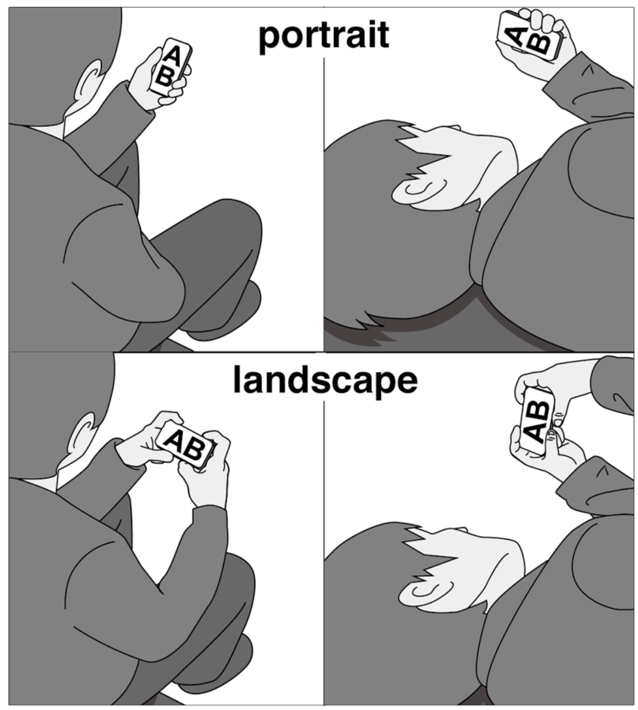

Figure 1: Example grasps of a smartphone in portrait and landscape orientations: a user's grasp remains consistent even while sitting and lying down, but is significantly different between screen orientations.

Computer vision-based automatic rotation approaches [2,4] use face detection to track a user's intended viewing orientation, rotating the screen accordingly. However, a user's face may not be clearly visible due to reasons including: motion blur, fingers blocking the front cameras, limited lighting, and device tilt. Even when a user's face is visible, often the images are limited to showing only partial facial features. iRotate [4] reported a successful face recognition

CHI 2013, April 27–May 2, 2013, Paris, France.

and rotation rate of 11% for iPhones and iPads using Apple's iOS 5 face detection API.

Another common solution to the screen rotation problem is to lock the device into the current screen orientation, which requires the user to manually disable the auto-rotation feature. This manual approach, however, requires cognitive effort and explicit user input. Even with a visual rotation lock indicator, a previous survey [4] shows that for users who have used rotation lock have experienced forgetting to turn off the lock 58% of the time. Although several gestures have been proposed to temporarily override the auto-rotation setting [5,8,10], these techniques still require explicit user input and require the user to learn new gestures as well.

To overcome the challenges of existing automatic screen rotation systems, we present iRotateGrasp, a system that automatically rotates a mobile device's screen based on a user's grasps. Our insight is that a user's grasp is consistent for a particular viewing orientation, but is significantly different between screen orientations. For example, when using a smartphone in portrait mode, a user's grasp is nearly identical—whether the users are sitting, walking, or lying down on one side. Figure 1 shows examples of grasps in different postures for both portrait and landscape modes.

Grasp-based interaction have been researched by several works [6,9,12,13]. iRotateGrasp senses how a user holds the devices, and uses machine learning to learn how various grasps map to intended screen orientations. To reduce power consumption, iRotateGrasp only performs grasp recognition when users unlock a device and when the accelerometer senses significant changes in device orientation.

To explore how well grasp can be used to infer the correct screen orientation, we implemented a phone-sized grasp sensing prototype by capacitive sensors and an iPod Touch. We recorded grasp data of 6 users performing 108 conditions twice which in combination of different touchscreen usages, upright and lying postures, holding hands and orientations. We sampled a total of 162000 grasp recordings, and we trained a multi-class support vector machine (SVM) to classify which orientation a grasp pattern belongs to. The results of 6-fold subject-independent cross validation shows 80.9% accuracy.

Our paper makes the following contributions: 1) we show that grasps can be used to automatically rotate screens to match users' view orientation with high accuracy, 2) we demonstrate that grasp-based auto rotation improves upon gravity-based approaches by supporting both upright and horizontal postures, 3) we implement a real-time grasp sensing and recognition prototype that is significantly more robust than previous computer vision-based approaches.

Feasibility Study

To better understand how users grasp devices, we conducted a 20-user feasibility study (10 females, age from 21 to 36) to observe their grasps of mobile phones. Participants were asked to use an iPhone 4 to read an article in portrait and landscape orientations while sitting and lying down. They followed on-screen messages to perform the following tasks: 1) sitting and reading in portrait orientation, 2) lie down on one side, 3) rotate the device 90 degrees into landscape orientation, 4) return to sitting position, 5) rotate the device 90 degrees to portrait. Each task lasted 15 seconds and each participant performed one trial.

We video recorded each session and analyzed users' grasps. Our observations show that: 1) none of the users change their grasps when changing their postures, 2) grasps for portrait orientation are distinct from grasp for landscape orientations, 3) single-handed grasps only touch the sides of the devices, 4) two-handed grasps touch both the sides and the back of the devices. Based on these observations, we implemented a grasp-sensing prototype.

Figure 2: Photos of iRotateGrasp prototype, showing an iPod Touch, an Arduino board, 2 multiplexers, and 44 capacitive sensors in an iPhone 4S case.

DESIGN AND IMPLEMENTATION

Our goal is to build a smartphone-sized device that can sense users' grasps. The form factor and weight should be similar to a typical smartphone to minimize changes in users' behavior. After exploring light sensors, which was sensitive to lighting conditions, we decided to use capacitive sensors because we can more easily position the sensors. The prototype consists of the following components: an iPod Touch 4, an Arduino Pro Mini 328 circuit board, 4 MPR 121 capacitive touch sensor controllers connected to 44 copper foils, and an iPhone 4S case. The prototype, as shown in Figure 2, is similar in size to iPhone 4S and its weight is 150g, 10g heavier than iPhone 4S. The Arduino board samples capacitive sensors at 60Hz. All subsequent processing is done by the iPod Touch.

Based on our observations from the feasibility study on where users hands contact the devices, we positioned sensors more densely on the four sides than on the back. As shown in Figure 2, we placed 10 sensors on each of the long sides and 5 sensors on each of the short sides, with 0.2cm distance between each sensor. We additionally added 4 sensors at each corner. On the back, we placed 10 larger sensors of (2.5cmx2.0cm).

Classified as →	a	b	c	Accuracy
a=portrait	46446	9218	6217	75.1%
b=landscapeLeft	14588	40112	5912	66.2%
c=landscapeRight	4034	1078	70839	93.3%

Figure 3: Confusion matrix of subject-independent cross validation (6-fold, leave-one-subject-out).

P1	P2	P3	P4	P5	P6	Average
71.4%	71.3%	84.8%	82.7%	88.0%	87.0%	80.9%

Figure 4: Recognition accuracy for each participant in leave-one-subject-out cross validation.

The iPod Touch runs iOS 5.1 and is jailbroken so we can use its serial port to receive sensor data. It also provides 3.3v to power the Arduino board and sensors.

Recognizing Grasp Orientation

Modern smartphones, such as iOS and Android, support 3 orientations: portrait, landscape left and landscape right. The fourth orientation, portrait upside-down, is supported on tablets but not on smartphones. We use LIBSVM [3], a support vector machines library, for the grasp orientation recognition. We use 1-vs-1 multi-class classification and the radial basis function (RBF) as the kernel. We treated each sensor reading as a 10-bit input and combined into a 44-dimension feature vector.

iRotateGrasp makes use of device power state and built-in accelerometers to reduce power consumption. Grasp recognition only runs when a device is unlocked and when the 3-axis accelerometer detects rotation exceeding the system threshold. These two events will trigger grasp recognition for a 0.2-second window, and uses voting to determine which orientation to rotate to. Since SVM can report the probability estimates for each class, we set a probability threshold 0.5 to reduce incorrect recognition. If voting does not produce a winner, due to a tie or no recognition, then iRotateGrasp falls back to gravity-based rotation.

EVALUATION

We evaluated our grasp-based approach through a 6-person data collection and offline analysis, and a 16-person user study using our real-time prototype.

Data Collection

We recruited 6 participants (3 female, age 20-24), and asked them to perform scrolling, pinch-to-zoom, and typing on our prototype to simulate typical usage of mobile phones. The 3 actions were performed under the following 3 × 4 × 3 conditions for a total of 108 tasks:

- grasping the device using left, right, and both hands (3) ×
- while standing, sitting, lying down (facing up), and lying down on one side (4) ×
- in portrait, landscape-left, and landscape-right orientations (3).

	P1	P2	P3	P4	P5	P6	Average
Accuracy	79.2 %	91.1%	89.8%	93.2%	92.6%	96.7%	90.4%

Figure 5: Recognition accuracy for within-subject cross validation.

For each task, participants followed on-screen prompt and we recorded 5 seconds of sensor data from each task. After performing the 108 tasks once, participants took a 5-minute break, and then repeated the 108 tasks again. Overall, we collected 6 users × 108 tasks × 2 trials × 25Hz × 5 seconds, 162000 samples in total. We ran a grid search to find a proper complexity parameter C=8 and width γ=0.125.

Recognition Accuracy

Figure 3 shows the confusion matrix for a 6-fold subject-independent (leave-one-out) cross-validation, in which the SVM is trained on 5 users and tested on the 6th user. This shows how well the approach works when there is no training data from the current user. The accuracy for each subject ranged from 71.4% to 88.0% in this leave-one-subject-out cross validation, and the overall average was 80.9%, as shown in Figure 4.

Figure 5 shows the within-subject cross-validation accuracy, in which the SVM. This is shows how well our approach could perform if the user is willing to train the system. The average accuracy is 90.4% across all users, and could be up to 96.7%.

Live System Evaluation

To evaluate our iRotateGrasp prototype, we conducted a 16-person user study (6 females, age 16-37). The participants performed the same tasks as in the feasibility study, so each user had 5 orientations for a total of 80. The participants were free to use any grasp they liked.

The grasp recognizer was triggered at the beginning of the session, and also when the device detected a change in device orientation. Participants were shown reading material on screen and were prompted to perform the next task every 15 seconds. The screen would auto-rotate according to

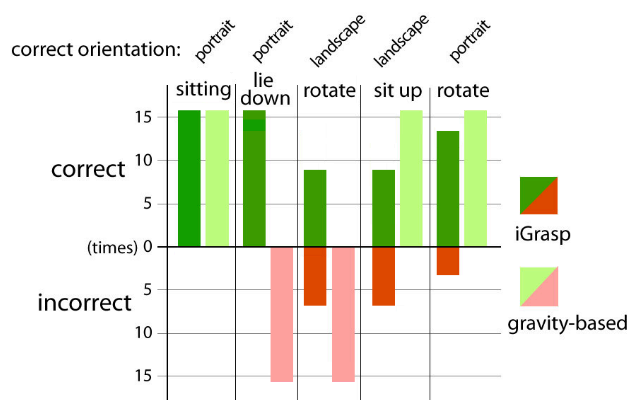

Figure 6: Correct and incorrect rotations by iRotateGrasp compared to gravity-based approach.

the grasp recognizer, which returns the most frequently detected orientation with confidence level over 50% in 0.2 seconds (5 grasp images at 25Hz recognition rate) after iOS system detecting auto-rotation threshold.

We recorded the number of correct and incorrect rotation for each session, and summarized the results in Figure 6. Overall our prototype had 78.8% accuracy, compared to 60% for gravity-based approach, an improvement of 31.3%. However, incorrect rotation occurred in upright posture for which gravity-based approach would always be correct.

DISCUSSION

Since grasp varies from person to person, an unusual grasp that had not been collected as part of the training dataset will cause incorrect rotation. One insight to enable training from normal usage data is that modern smartphones, such as iOS and Android, are designed to only be unlocked in portrait orientation. In these cases, we can tag the grasp image by the correct orientation, and train the classifier to learn the portrait grasp of the user. Computer vision-based approaches could also provide ground truth for training.

Although our accuracy of 78.8% (live) to 80.9% (offline) appears to be significantly higher than the 11% accuracy reported by previous computer vision-based approach [4], one key difference is that face detections typically fails and returns "unknown", where as our approach is more likely to fail to a wrong class. Increasing confidence threshold in our prototype will likely decrease its accuracy.

Our live testing results show that our current prototype is much less accurate for landscape orientation than for portrait orientation. If we are willing to assume that users will not be using their device up side down, one approach to improve the performance for landscape in upright posture is to remove that impossible orientation.

CONCLUSION & FUTURE WORK

We have presented iRotateGrasp, a grasp-based approach to automatically rotate screens to match users' viewing orientation. It augments gravity-based rotation techniques by classifying users' grasps into users' viewing orientations before falling back to gravity-based rotation. The approach can rotate screens correctly in different user postures and device orientation without explicit user input.

We implemented a grasp-sensing prototype and conducted several studies to demonstrate that grasps can be used to classify users' viewing orientation at 80.9% accuracy, and could reach 90.4% if the user were willing to train the system. In addition, it runs in real-time on current mobile devices. The live evaluation of our prototype demonstrated that the approach is promising at 78.8% accuracy and outperforms current gravity-based approaches by 31.3%.

Currently, we are combining grasp and computer vision-based approaches. We are also investigating incremental learning in order to personalize the classifiers. Looking at the delta between grasps that occur before and after orientation change to determine the screen orientation is another way we attempt to improve the accuracy. In addition, if we can accurately recognize grasp and orientation, we can infer the user's posture based on orientation and accelerometer readings. For example, if a user's viewing orientation is detected as portrait but device orientation is detected as landscape, the user may be lying on one side. We plan to explore uses of user posture as an interaction technique.

REFERENCES

1. Bartlett, J.F. Rock 'n' Scroll is Here to Stay. *IEEE Computer Graphics and Applications*, May (2000), 40–45.
2. Blaskó, G., Beaver, W., Kamvar, M., and Feiner, S. Workplane-orientation sensing techniques for tablet PCs. *Proc. UIST '04*, 1-2.
3. Chang, C.-C. and Lin, C.-J. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST) 2*, 3 (2011).
4. Cheng, L.-P., Hsiao, F.-I., Liu, Y., and Chen, M.Y. iRotate : Automatic Screen Rotation based on Face Orientation. *Proc. CHI '12*, 2203–2210.
5. Forstall, S. and Blumenberg, C. Portrait-Landscape Rotation Heuristics for a Portable Multifunction Device. *US Patent. 20,080/165,144*, 2011.
6. Goel, M., Wobbrock, J.O., and Patel, S.N. GripSense : Using Built-In Sensors to Detect Hand Posture and Pressure on Commodity Mobile Phones. *Proc. UIST '12*, 545-554.
7. Hinckley, K., Pierce, J., Sinclair, M., and Horvitz, E. Sensing techniques for mobile interaction. *Proc. UIST '00*, 91–100.
8. Hinckley, K. and Song, H. Sensor synaesthesia: touch in motion, and motion in touch. *Proc. CHI '11*, ACM, 801–810.
9. Kim, K., Chang, W., Cho, S., et al. Hand grip pattern recognition for mobile user interfaces. *Proceedings of the National Conference on Artificial Intelligence*, MIT (2006), 1789–1794.
10. Ording, B., Van Os, M., and Chaudhri, I. Screen Rotation Gestures on a Portable Multifunction Device. *US Patent App. 20,080/211,778*, 2011.
11. Schmidt, A., Beigl, M., and Gellersen, H.W. There is more to context than location. *Computers & Graphics 23*, 6 (1999), 893–901.
12. Taylor, B.T. and Bove Jr, V.M. Graspables: grasp-recognition as a user interface. *Proc. CHI '09*, 917–926.
13. Wimmer, R. and Boring, S. HandSense: discriminating different ways of grasping and holding a tangible user interface. *Proc. TEI '09*, 359–362.