Headshot: A Phone Application That Helps People Take Pictures of Themselves

Julia Schwarz
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA 15217
julenk@cmu.edu

ABSTRACT
This note presents a new technique for helping users take photos of themselves. Specifically, it presents Headshot: a phone application that combines face detection and voice-based guidance to help people take better pictures of themselves on mobile phones. The note gives an overview of implementation and presents results from an informal user study evaluating the application’s effectiveness. I found that this technique is especially effective in helping users take photos of themselves when users want their faces to be on the side of the photo (when, for example, they want their head to be next to a landmark or other point of interest), offering a twofold increase in accuracy over guidance-free self-photos.

Author Keywords
Photography, self-photography, face detection

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI)

INTRODUCTION
Mobile phones are becoming the primary camera for many people. Many mobile phone cameras these days are capable of taking excellent photos. However, one scenario that remains lacking is taking photos when the subject is the photographer him/herself. Self-portraits are still difficult to take, especially in impromptu situations and on mobile phones when people do not have access to shutter triggers, tripods, and front-facing displays. Oftentimes the user’s face is not where they want it to be, or is not in the picture at all. Front-facing cameras on mobile phones begin to solve this problem, but these have much lower (VGA-quality) resolution than rear-facing cameras, and are unlikely to improve any time soon.

This paper presents a new technique to help users take better pictures of themselves. This technique combines face detection with voice guidance to help users position their cameras when taking self-photos so that their faces are where they want (i.e. in the center or on the side of pictures). After giving an overview of existing work in the area, I describe our implementation, and present results from a user study evaluating the effectiveness of the technique. Overall, I found this technique to facilitate significantly more precise head positioning when the head position was on the left or right side of the photo, providing a twofold increase in accuracy over the headshot-free condition.

BACKGROUND
Self-portraits are a very common use-case for photography, especially in informal settings such as when users are taking photos with their mobile phones. As a result, manufacturers and even users themselves have devised several techniques aimed at reducing these problems.

One simple solution is to put a mirror on a mobile phone [5], however this requires purchasing extra items and modifying the phone. Some cameras such as the Samsung DualView TL220 12 MP Digital Camera [9] offer front-facing displays, however in a mobile phone setting such displays are prohibitively expensive and are unlikely to show up on mainstream phones any time soon. Finally, many phone manufacturers have begun building front-facing cameras into their mobile phones. Some examples are the iPhone 4S [2] and Samsung Focus S [10]. However these front-facing cameras have much lower resolution than their rear-facing counterparts. For example, the iPhone 4s has an 8 MP rear-facing camera, while the front facing camera is 640x480 (0.3 MP) [3].

In addition to modifying hardware, some software modifications have recently been introduced to improve self-portrait quality. For example, the SelfPhoto app for the iPhone takes a photo of a user once their head is in the camera frame [11]. Additionally, some Sony cameras have a “smile shutter” feature which takes photos only when all subjects in a photo are smiling [1]. Finally, the Morpho iPhone app uses voice guidance to help users take better photos [6], however the app is not publicly well-documented, thus it is unclear how effectively the system works.

Surprisingly, there has been little to no work in academic circles regarding techniques for improving self-photography. The contribution of this paper is in presenting an entirely software-based technique for taking better self-
photos. Additionally, this paper describes the technique in detail and provides an evaluation of effectiveness.

SYSTEM OVERVIEW
The principles behind our technique are simple. When the user wants to take a picture of him/herself, he/she specifies where their head should be by moving a target, then as he/she takes a photo of him/herself the device tells him/her how to rotate the camera to get his/her head in the correct position. The voice feedback is presented as one of a set of four instructions: “up”, “down”, “left”, or “right”, indicating how the phone should be rotated. Once the user’s head is in the correct position, the device says “perfect” indicating that the user’s head is in the correct position.

Headshot is implemented in C#, and is built for the Windows phone. Because it is implemented for the Windows Phone, a custom face detection algorithm was built as no library could be used. The details of the face detection are provided below, and the code is publicly available at http://facedetectwp7.codeplex.com.

Face Detection
The face detection algorithm implemented for this application was based off of the viola-jones object detection framework [8]. This algorithm builds a machine-learned cascade of classifiers using the Adaboost algorithm. The features used are Haar-like features representing differences of sums of different rectangular regions. Because training a robust model is very time-consuming, our algorithm instead used an existing model provided by OpenCV [7]. Reading the OpenCV XML file description of the model was non-trivial, and our implementation was based off of the JViolaJones library [4] which accomplished a similar task in Java.

Our face detection code was built as a library for Windows Phone and is publicly available at http://facedetectwp7.codeplex.com. The face detection for our application detects faces within 100 ms, which is acceptable for our application.

EVALUATION
To evaluate the effectiveness of our technique, I ran an informal 10-person within-subjects pilot study to see whether our technique allowed people to more precisely position their heads when taking pictures of themselves. I tested 4 conditions: whether the target was in the center of the image or on the left side of the image, and whether or not the user was receiving voice feedback. 4 participants were male, and ages ranged from 22 to 60.

After users took a photo I measured how far away (in pixels) their face was from their desired location. This distance was measured based on the distance of the center of the detected face region to the target location.

Results
Figure 2 illustrates results. As the figure indicates, users were able to position their heads fairly accurately when the target was in the center even without guidance from Headshot. However, when the target was on the side of the portrait (this simulates a scenario when the user wants to take a picture of themselves next to a landmark), users were able to position their heads much more closely to the target when using Headshot. These results suggest that Headshot is particularly helpful when users want to take pictures of themselves with their heads on the side of an image, such as when near landmarks or when trying to follow the rule of thirds.

LIMITATIONS AND FUTURE WORK
There are several limitations to our technique and our pilot study.

In the study I measured distance to goal by using distance from the detected facial region. This gives Headshot an unfair advantage because the feedback relies on the detected face location as well, and in principle the detected region could be completely incorrect. However, in practice I found the detected facial region to be highly accurate, and
would be more accurate than the alternative of manually computing the face location every time.

With regards to Headshot, audio-based feedback would not work in noisy environments such as on busy city streets. I experimented with providing haptic feedback, however lack of directional feedback (i.e. only communicating how close the user’s head was to the target) severely limited the effectiveness of haptic feedback in pilot studies, so I did not include this in the final application. Second, the application can only tell users where their head is, and cannot ensure that a landmark or point of interest is visible. This makes it a little challenging to take pictures next to landmarks. Finally, the audio feedback occurs about once every second, and thus users may over-compensate when rotating and accidentally take a photo which they think is perfect simply because the audio feedback did not have time to tell them to rotate backward. All of these drawbacks can be solved using front-facing, visual feedback, however as mentioned before such solutions are expensive and not everybody can afford this. Headshot provides a nearly free, in-software solution for taking self-portraits.

Future work on this application lies in two areas. First, I hope to improve the face detection algorithm to more quickly find faces and reduce latency. Second, I would like to continue developing applications that help people take photos of themselves. The next idea I’d like to implement is called Remoteshot, an application that allows users to take a group photo without using a timer. One phone acts as the camera, and another phone acts as the trigger.

CONCLUSION

This paper presents a new technique to help people take self-photos. The technique uses face detection and voice feedback to tell users how to position their heads when taking pictures of themselves. This paper describes the implementation of our technique and presents results from a study that indicate our technique provides a twofold increase in accuracy when taking photos where the head is on the side of the photo. In conclusion, Headshot is a free, in-software technique for helping people take better pictures of themselves.

ACKNOWLEDGMENTS

I would like to thank Professor Alexei Efros for his excellent class, our TA Natasha for the help and prizes, and the folks who wrote the jviolajones library for their code (http://code.google.com/p/jviolajones/).

REFERENCES

2. iPhone 4s. http://www.apple.com/iphone/