

LookToFocus: Image Focus via Eye Tracking

SAIKIRAN TEDLA, Dept. of Electrical Engineering and Computer Science York University, Canada

I. SCOTT MACKENZIE, Dept. of Electrical Engineering and Computer Science York University, Canada

MICHAEL S. BROWN, Dept. of Electrical Engineering and Computer Science York University, Canada

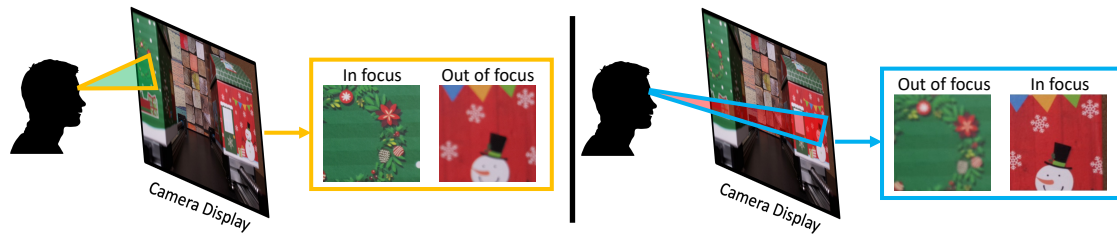


Fig. 1. LookToFocus allows users to control focus by looking at a region of interest (ROI). LookToFocus performs manual image focus and is faster and more intuitive than TapToFocus, a touch-based method.

We present LookToFocus, a method to perform real-time manual camera focus based on eye tracking. LookToFocus and two alternative methods for manual focus¹ photography tasks were compared in a user study. A novel manual focus camera simulation was used to test the methods. The first two methods were LookToFocus and LookToFocusNB (no bounding box). The third method, TapToFocus, used touch for manual focus and image capture, analogous to typical smartphone interaction. LookToFocusNB had the fastest mean capture time at 1429 ms; the mean capture times were 1431 ms for LookToFocusNB and 2416 ms for TapToFocus. When compared to TapToFocus, LookToFocus and LookToFocusNB had faster capture times because these algorithms start converging to the optimal focus as soon as the user spots the target. LookToFocus and LookToFocusNB also had no significant decrease in sharpness error compared to TapToFocus. Users preferred LookToFocus over both LookToFocusNB and TapToFocus.

CCS Concepts: • **Human-centered computing** → **Empirical studies in ubiquitous and mobile computing**; **Interaction devices**.

Additional Key Words and Phrases: Focus, Eye Tracking, Manual focus, Autofocus, Smartphone

ACM Reference Format:

SaiKiran Tedla, I. Scott MacKenzie, and Michael S. Brown. 2024. LookToFocus: Image Focus via Eye Tracking. In *2024 Symposium on Eye Tracking Research and Applications (ETRA '24)*, June 4–7, 2024, Glasgow, United Kingdom. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3649902.3656358>

1 INTRODUCTION

Smartphones capture over a trillion images every year [Haueter 2022]. The quality of these images is largely decided by the focus. Typically, users capture images in an autofocus (AF) mode that sets the focus automatically through

¹When using eye-tracking, we use the term “manual focus” to imply user-controlled focus (but not with the hands).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

Manuscript submitted to ACM

algorithms that analyze basic scene content. However, sometimes a user wants focus on a particular scene region or object that is missed by the algorithm. In such cases, a manual focus adjustment is needed [Wildfellner 2012]. Manual focus is commonly done using a region of interest (ROI) selected by positioning a bounding box with a finger touch. The focus is then adjusted by an AF algorithm based on the specified ROI. There has been little research on methods to improve the user experience when focusing images on a smartphone camera. Additionally, much of this work does not take advantage of pipelines for standard manual focus operations.

Recent smartphone designs include front-facing cameras with the potential for accurate eye tracking [Liu et al. 2015; Paletta et al. 2014]. We propose, LookToFocus, a new method that uses eye tracking for manual focus image capture. Our method is fast and accurate as it employs techniques used in standard manual focus pipelines; this is in contrast with previous work which does not use manual focus techniques [Clancy et al. 2011; Fuhl et al. 2017]. Figure 1 illustrates how LookToFocus works. In this paper, we compare TapToFocus, a touch-based image capture method, and LookToFocus for the task of image capture when adjusting focus.

Additionally, a novel manual focus simulation platform is presented that allows for repeatability and accuracy in experimental testing. An autofocus dataset [Abuolaim et al. 2018] was used to emulate a camera experience on a desktop by presenting different photos from a focal stack depending on the target region chosen by the user in an experiment task (e.g., “focus the camera on the face”). The goal was to evaluate the viability of eye tracking as an image capture method through quantitative and qualitative measures.

This work has the following contributions:

- (1) Presenting LookToFocus/LookToFocusNB as methods for real-time manual focus with eye tracking. This method is the first to implement eye-tracking focus using standard manual focus operations.
- (2) A novel manual focus simulation platform to test image capture methods.
- (3) The first user study to compare tap-based and eye-tracking based methods for manual focus. In our user study, we evaluate TapToFocus, LookToFocus, and LookToFocusNB.

1.1 Background

Focus is an important part of imaging because it determines what parts of an image are sharp [Shirvaikar 2004]. Almost all consumer cameras are equipped with autofocus (AF) algorithms that choose an ROI and adjust the focal length so the ROI is in focus [Jeon et al. 2011; Vuong and Lee 2013]. An ROI is said to be in focus when the ROI is within the depth of field (DOF) of the lens system. The DOF is the range of distance from the camera where objects are considered to be acceptably sharp [Liao et al. 2019]. AF has been widely studied [Abuolaim et al. 2018; Jeon et al. 2011; Ooi et al. 1990; Vuong and Lee 2013; Wang et al. 2021], with many different approaches. But, there is not always a single correct solution for what parts of the image to prioritize when implementing AF [Abuolaim et al. 2018]. Figure 1 illustrates two different focus settings that are acceptable depending on the user’s intent. Sometimes an “incorrect” focus is chosen by the AF algorithm, and when this occurs the user must perform manual focus to specify the desired ROI.

A typical manual focus operation on a smartphone consists of a user tapping on the screen to set an ROI with a fixed-size bounding box (BB); the algorithm then maximizes the ROI sharpness. Measuring the sharpness of an ROI is done through various methods. An example is a Sobel filter [Shih 2007] which measures the gradient in the horizontal and vertical direction by performing convolutions with edge-detecting filters within the ROI. The gradient magnitude can then be computed from the horizontal and vertical gradients. Maximizing the gradient magnitude within the ROI maximizes the ROI sharpness [Kittler 1983].

The AF used to maximize ROI sharpness typically consists of two parts: phase difference autofocus (PDAF) and contrast detection autofocus (CDAF). First, given an ROI, an estimated optimal focal position is computed using PDAF. Then, this estimated focal position is fine-tuned with CDAF. CDAF performs a local search for the maximum ROI sharpness by moving the focal position back and forth. One effect of CDAF is jitter from the local search.

Some research explores speeding-up manual focus with eye tracking. A full depth map of the scene [Fuhl et al. 2017] can be used to control the lens position as the user moves their eyes across the image. However, acquiring a full depth map is not trivial because it requires both time and scene constancy; this is not practical for real-time imaging situations. Others [Clancy et al. 2011] used eye-controlled focus for an endoscope, but this system can't be extended to consumer cameras because it makes strong assumptions about endoscope images.

Additionally, a few high-end DSLR cameras implement eye-controlled focus within a viewfinder [Weinblatt 1986]. However, it is not clear if viewfinder eye-controlled focus is preferred by the professionals with access to these high-end cameras as no user study has been done. Next, since real-time eye-control focus is only available on proprietary devices with viewfinders, there is no public framework for testing these types of interfaces. Thus, in our work we also provide a method for how to use a simulation platform to accurately model real-time eye-control focus. Finally, it is not clear if eye-control focus within a viewfinder could extend to eye-control focus without a viewfinder. Eye-control focus without a viewfinder could be used on mobile devices (e.g., smartphones) and reach a much larger audience than just professional photographers. This is our motivation for testing eye-control focus in a setting without a viewfinder.

There is also extensive work on using eye tracking for general-purpose target selection [Putze et al. 2016; Skovsgaard et al. 2010; Vertegaal 2008; Zhu et al. 2018]. A lot of this work suggests eye-tracking can perform faster target selection than other methods of input. However, it is not immediately clear if eye-tracking is faster or preferred for the task of manual focus. This is because manual focus naturally has jitter; that is, visually, the image changes as the eyes move.

Finally, we distinguish eye-controlled image focus from work that studied focus in the context of imaging the eye [Liu et al. 2007].

The lack of real-time eye-controlled focus methods is the impetus for our work. Thus, we propose real-time eye-controlled focus modeled from a typical manual focus operation. This is explored and reported through a corresponding user study.

2 MANUAL FOCUS SIMULATION PLATFORM

2.1 Camera Simulation

Ideally, we would use a smartphone for our initial implementation of LookToFocus, but it is not currently possible to obtain PDAF/CDAF camera information without proprietary access. Prior work [Abuolaim et al. 2018] acknowledges this limitation and models AF using the concept of “focal” stacks. Each “focal” stack in their dataset contains 50 different focal positions.

We propose a novel manual focus camera simulation by using their AF dataset and modifying their data browser.² This dataset consists of ten scenes captured by a Samsung Galaxy smartphone, where each scene is captured using stop-motion animation. For this research, we do not use the temporal aspect of the dataset and instead sample focal stacks at different time steps within the scenes. See Figure 2.

An important challenge in the simulation is to accurately model manual focus on a typical smartphone. We do this by first allowing the user to select an ROI. On a standard camera, an AF algorithm is called to maximize the sharpness

²Code for this simulation and data are at <https://github.com/tedlasai/LookToFocus>

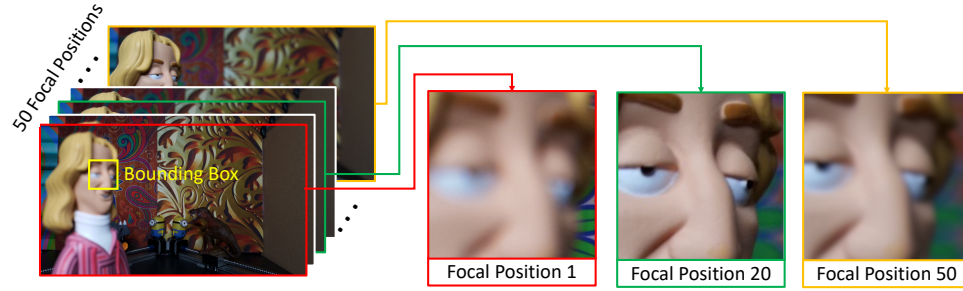


Fig. 2. The focal stack. Focal position is simulated by transitioning through images in the stack. The yellow box appears when using the LookToFocus and TapToFocus methods.

within a fixed-location ROI. However, in the simulated camera, the ROI location is continuously updating; thus, our simulated camera requires a modified AF model. Our modified AF aims to mimic the combination of PDAF and CDAF on standard cameras.

PDAF can be simulated by running CDAF on the ROI selected for all images in the focal stack and computing the optimal focal position o [Abuolaim et al. 2018]. Then, an error tolerance ρ is used to emulate the inaccuracy of a standard camera PDAF [Abuolaim et al. 2018]. This simulated PDAF uses a uniform random sampling between $o - \rho$ and $o + \rho$ to return a realistic PDAF focal position estimate f .

We discovered that using an error tolerance is not just helpful for realistic PDAF simulation, but also for replicating the CDAF used on a standard camera. With this in mind and as stated above, we implemented a new AF algorithm. First, we iteratively call PDAF (every 8 ms) with each new ROI and then adjust the current focal position c until it is within a threshold t from the aforementioned position estimate f (i.e., $|c - f| \leq t$). Each iteration of PDAF can only move c forward or backward in steps of one. Since f can be randomly updated at each time-step, it can cause $|c - f|$ to cross the threshold t repeatedly; this causes the focal position to jitter in a manner similar to CDAF. Finally, we set $\rho = t$ so the jitter stops once the current focal position equals optimal focal position ($c = o$).

For LookToFocus/LookToFocusNB, image capture uses a dwell-time criterion. For TapToFocus, image capture is done by pressing a capture button. Additionally, we call PDAF every 8 ms because this is the standard time for lens movement on a smartphone camera in preview mode. With this 8 ms delay, we found our simulation to be comparable to modern smartphones.

2.2 Image Capture Methods

We used the camera simulation platform shown in Figure 3 to support three image capture methods:

- TapToFocus - A tap on the touch screen moves the BB. The center of the ROI bounding box is specified by the location of the tap. Then, the user can “take” the image by tapping on the capture button displayed on the screen. The BB is visible to the user in this method.
- LookToFocus - Eye tracking is used to move the BB. The center of the BB is defined as the average of the last N gaze estimates from the eye tracker; this smooths the bounding box location. Additionally, we used dwell to capture the image. We detect a dwell by averaging the horizontal and vertical gaze variance for the last L

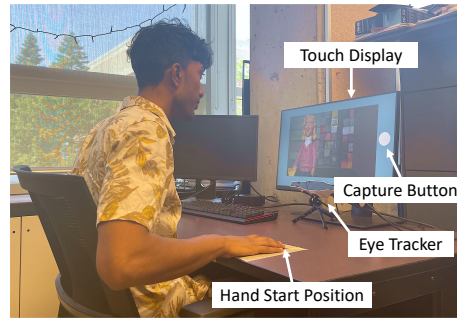


Fig. 3. A user capturing an image in our camera simulation. The hand start position and capture button are used for TouchToFocus.



Fig. 4. Example of a scene with “Face” as the object of interest. The starting image (left) has the center in focus but the ROI is out of focus. The ground truth image (right) has the optimal focus for the ROI. This is the image the user aims to “capture” in a trial.

ms is less than some threshold t . Based on initial testing, we found $N = 60$ gaze estimates, $L = 800$ ms, and $t = 1500$ pixels,² to be reasonable values for simulation. Finally, the BB is visible to the user in this method.

- LookToFocusNB (no box) - This method is the same as LookToFocus except the BB is not visible to the user.

Each image capture method continuously updates the location of a fixed-size ROI region. The BB controlled by the user is a square of size 100×100 pixels. This emulates the fixed-size BBs found on many consumer smartphones.

In a real-world situation with a smartphone camera, TapToFocus is analogous to a user moving their finger to touch the ROI on the camera display, which in turn directs the AF algorithm to focus on the target. However, with LookToFocus or LookToFocusNB, users just look at the target.

A user study compared the three different image capture methods: TapToFocus, LookToFocus, and LookToFocusNB. We evaluated the performance of these methods through quantitative measures and by asking users to rank the methods in order of preference and simplicity.

3 METHOD

We recruited 24 local university students who regularly use their smartphones for capturing images. Participants consisted of 16 males and 8 females with ages from 23 to 39 years (mean = 25.8, $SD = 3.97$).

The apparatus was built around the manual focus simulation platform discussed in Section 2. Figure 3 shows the setup consisting of an eye tracker and display. The simulation ran on a desktop running Windows 10. The eye tracker was a GazePoint GP3 running at 60 Hz with $0.5^\circ - 1^\circ$ visual angle accuracy. The touch monitor was a Dell P2418HT with a resolution of 2560×1440 pixels and a 60-Hz refresh rate. The camera simulation platform displayed the scene at 1500×1243 resolution centered within the screen (because initial testing of the eye tracker showed more accuracy error at the edges of the screen).

Participants sat in front of a computer monitor with the camera simulation centered within the screen. Eye tracker calibration was performed first, followed by instructions introducing the task, and then practice trials. Following this, participants completed nine trials, each containing a different scene for each method. For each trial, the simulation platform first presented a word describing an object in the scene. Next, the scene appeared with an incorrect focal position (i.e., the object ROI was not in focus). Then, the user adjusted focus and captured the image with the current image capture method. For TapToFocus trials, users were asked to start each trial with their right hand on the desk within a marked box 1 foot from the monitor. This emulates moving the hand from a neutral position to adjust the focus; a scenario like this arises when users are taking a selfie photo. See Figure 3 for an example of a user capturing an image in our camera simulation.

For any given scene, the starting incorrect focal position of the scene produces maximum sharpness at the center of the image. The object of interest is positioned off from the center of the image. Figure 4 shows an example of a scene. Participants were organized in six groups for counterbalancing; each group had a different permutation of testing the three image capture methods. The experiment took 10 to 15 minutes per participant.

This study employed a 3×9 within-subjects design with the following independent variables and levels:

- Image capture method: TapToFocus, LookToFocus, LookToFocusNB
- Scene: 1, 2, ... 9

Each scene contains a word describing an object in the image, a starting focal position where the ROI is incorrectly focused, and a ground truth focal position where the ROI has the highest sharpness. We used scenes where focusing on the center initially would produce an incorrect focal position; nine of ten scenes in the AF dataset [Abuolaim et al. 2018] fit this criterion.

The dependent variables were as follows:

- Capture time (ms) - the time to perform manual focus and then “capture” the image
- Sharpness error (%) - difference in sharpness, measured with the Sobel filter, of the scene-specific ROI between “captured” image and ground truth
- User experience - participant rankings on the three image capture methods on preference and simplicity

The total number of trials was $24 \text{ participants} \times 3 \text{ image capture methods} \times 9 \text{ scenes} = 648$.

4 RESULTS AND DISCUSSION

Results are now presented, organized by dependent variable. All ANOVA and post hoc results were computed with GoStats.³ The group effect was not statistically significant for capture time ($F_{5,18} = 0.99, ns$) and sharpness error ($F_{5,18} = 1.51, p > .05$), thus implying that counterbalancing effectively offset any order effects.

³<http://www.yorku.ca/mack/GoStats/>

4.1 Capture Time

LookToFocusNB and LookToFocus had mean capture times of 1429 ms per scene and 1431 ms per scene, respectively. The slowest method was TapToFocus at 2416 ms per scene. See Figure 5a. The differences were statistically significant ($F_{2,36} = 49.87, p < .0001$). A post hoc Fisher LSD test revealed a significant difference between TapToFocus and both LookToFocus methods. As evident in the figure, LookToFocus and LookToFocusNB are faster than TapToFocus by 40.9% and 40.8%, respectively.

LookToFocus and LookToFocusNB are faster than TapToFocus because they remove the manual target pointing step and capture button press. When using TapToFocus, the participant must find the target and tap on it. The time to first tap on the screen was on average 1527 ms. This means on average the LookToFocus methods had captured the image before TapToFocus even started to change the focus of the image. Additionally, after the first tap, time is required for the manual focus algorithm to converge and the capture button to be pressed (which took 889 ms on average). LookToFocus is advantageous in many scenarios because the focus starts converging to the optimal focus as soon as the user locates the target.

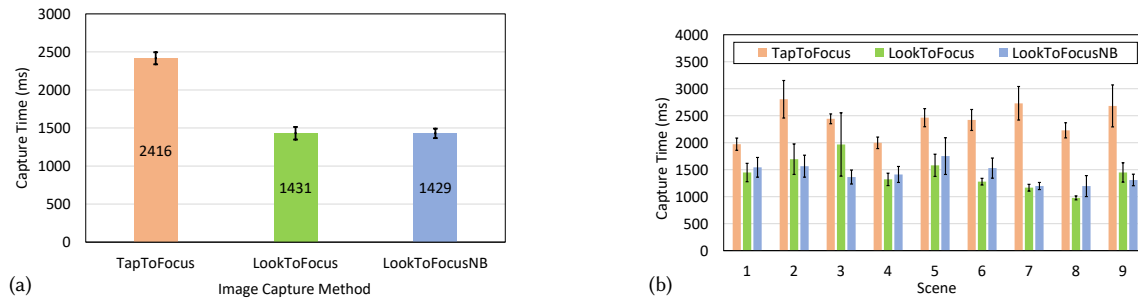


Fig. 5. Results for (a) capture time (ms) by image capture method and (b) capture time by scene and image capture method. Error bars show ± 1 SE.

Figure 5b compares the image capture methods across the nine scenes for capture time. LookToFocus and LookToFocusNB were faster than TapToFocus for all scenes. The figure shows that scene affects the capture time with an ANOVA revealing that the main effect of scene on capture time was statistically significant ($F_{8,144} = 3.38, p < .005$). The image capture method by scene interaction effect on capture time was not statistically significant ($F_{16,288} = 1.511, p > .05$).

4.2 Sharpness Error

Before performing the analysis, we removed trials with a sharpness error outside ± 3 SD from the grand mean. These were considered outliers due to two errant behaviours we observed: focusing on an incorrect object or prematurely capturing the image. In applying the criterion, 18 trials (2.7%) were deemed outliers and removed, leaving 630 trials.

The mean sharpness error by image capture method was 1.62% for TapToFocus, 1.73% for LookToFocus, and 1.79% for LookToFocusNB. All methods demonstrated less than 2% sharpness error, which is acceptable in most cases. The effect of image capture method on sharpness error was not statistically significant ($F_{2,36} = 0.489, ns$). Figure 6 compares the image capture methods by sharpness error.

We also observed a significant main effect of the scene on sharpness error ($F_{8,144} = 17.34, p < .0001$). This makes sense as scenes vary in terms of target salience, size, and object. The image capture method by scene interaction effect

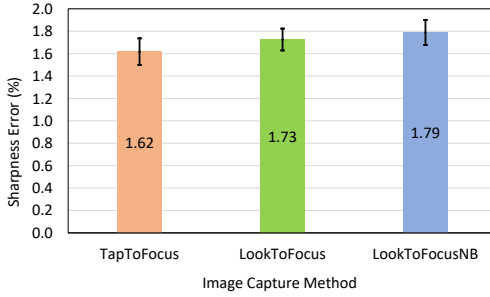


Fig. 6. Sharpness error by image capture method. Error bars represent ± 1 SE.

on sharpness was statistically significant ($F_{16,288} = 1.765, p < .05$), likely due to the very low p -value for the main effect of scene.

4.3 User Experience

Users were asked to rank all three methods in terms of preference and simplicity after completing the experiment. We converted the rankings to a 3-point rating scale where 2 is the most preferred and 0 is the least preferred.

4.3.1 Preference. For preference, the average rating (out of 2) was 0.65 (22% of overall rating points) for TapToFocus, 1.29 (43%) for LookToFocus, and 1.05 (35%) for LookToFocusNB. A Friedman test revealed that the differences were statistically significant ($\chi^2 = 9.25, p < .01, df = 2$). Conover’s post hoc pairwise test showed a significant difference between LookToFocus and TapToFocus. This means there was a clear preference for LookToFocus compared to TapToFocus, but not between LookToFocus and LookToFocusNB.

This is in line with what we observed in participant feedback. We noticed 22 out of 24 participants chose LookToFocus or LookToFocusNB as their first preferred method. Many participants had a strong opinion on the visible bounding box, with 10 participants rating the other LookToFocus method (the one they didn’t choose) last. Some users mentioned that the visual feedback (BB) present in LookToFocus was beneficial while other users thought the BB was distracting. Thus, we recommend any future implementation of LookToFocus make bounding box visibility a user option.

4.3.2 Simplicity. For simplicity, average rating (out of 2) was 1.08 (36% of overall rating points) for TapToFocus, 1.17 (38%) for LookToFocus, and 0.79 (26%) for LookToFocusNB. A Friedman test revealed no statistically significant difference ($\chi^2 = 2.15, p > .34, df = 2$). Thus, participants didn’t find one method simpler than the others.

5 LIMITATIONS

This evaluation of LookToFocus is limited in its external validity, since the apparatus used a touch-sensing desktop display. Using a smartphone was not possible since this requires proprietary access to the image capture pipeline. Although our setup was a simulation of a smartphone, the time saved by using LookToFocus should extend to smartphones since eye-tracking methods allow the focus to converge as soon as the object of interest is located.

Additionally, eye tracking accuracy is still not satisfactory on smartphones; however, recent work shows potential for improvement [Liu et al. 2015; Paletta et al. 2014]. Thus, this paper lays the groundwork for future smartphone versions of LookToFocus.

6 CONCLUSION

We introduced LookToFocus and LookToFocusNB, two novel methods for image capture with real-time manual camera focus. These methods leverage eye tracking for hands-free operation. We built a novel manual focus simulation that allows for manual focus with a continuously updating ROI so we could test LookToFocus and LookToFocusNB.

We also used TapToFocus as a third baseline image capture method that was analogous to a normal manual focus techniques on a smartphone. LookToFocus and LookToFocusNB were faster than TapToFocus, reducing capture time by 40.8% and 40.9%, respectively. Both eye-tracking methods were faster because they allow for manual focus to converge as soon as the target object is located. This allows LookToFocus and LookToFocusNB to be beneficial when applied on smartphones. Hands-free focusing is also potentially useful in accessible computing where manually touching the display is challenging or not possible. Furthermore, in real-world situations, users might be outdoors and wearing gloves or taking pictures in a weather-challenged situation.

Additionally, there was no significant difference in sharpness error between the three methods. Our user survey also indicated a preference for LookToFocus over LookToFocusNB and TapToFocus.

Finally, we expect our camera simulation platform is of use for research relating to controlling camera focus at image capture time. Our camera simulation platform can be extended for live manual focus on video and modeling more complex focus algorithms.

ACKNOWLEDGMENTS

A special thanks to Abullah Aboulaim for his helpful advice. Thank you to the participants of the user study. Thank you to Ankitha and the Desu family for inspiring this idea.

REFERENCES

- Abdullah Abuolaim, Abhijith Punnappurath, and Michael S Brown. 2018. Revisiting Autofocus for Smartphone Cameras. In *European Conference on Computer Vision (ECCV)*. Springer, New York, 523–537. https://doi.org/0.1007/978-3-030-01267-0_32
- Neil T Clancy, George P Mylonas, Guang-Zhong Yang, and Daniel S Elson. 2011. Gaze-contingent autofocus system for robotic-assisted minimally invasive surgery. In *Engineering in Medicine and Biology Society (EMBC)*. IEEE, New York, 5396–5399. <https://doi.org/10.1109/IEMBS.2011.6091334>
- Wolfgang Fuhl, Thiago Santini, and Enkelejda Kasneci. 2017. *Fast camera focus estimation for gaze-based focus control*. Technical Report. Cornell University. <https://doi.org/10.48550/arXiv.1711.03306>
- David Haueter. 2022. US photo merchandise study. <https://riseaboveresearch.com/rar-reports/2022-us-photo-merchandise-study/> Accessed Dec. 30, 2022.
- Jaehwan Jeon, Inhye Yoon, Jinhee Lee, and Joonki Paik. 2011. Robust focus measure for unsupervised auto-focusing based on optimum discrete cosine transform coefficients. In *International Conference on Consumer Electronics (ICCE)*. IEEE, New York, 193–194. <https://doi.org/10.1109/TCE.2011.5735472>
- J Kittler. 1983. On the accuracy of the Sobel edge detector. *Image and Vision Computing* 1, 1 (1983), 37–42. [https://doi.org/10.1016/0262-8856\(83\)90006-9](https://doi.org/10.1016/0262-8856(83)90006-9)
- Meihua Liao, Dajiang Lu, Giancarlo Pedrini, Wolfgang Osten, Guohai Situ, Wenqi He, and Xiang Peng. 2019. Extending the depth-of-field of imaging systems with a scattering diffuser. *Scientific Reports* 9, 7165 (2019), 1 – 9. <https://doi.org/10.1038/s41598-019-43593-w>
- Dachuan Liu, Bo Dong, Xing Gao, and Haining Wang. 2015. Exploiting eye tracking for smartphone authentication. In *International Conference on Applied Cryptography and Network Security*. Springer, New York, 457–477. https://doi.org/10.1007/978-3-319-28166-7_22
- Ruian Liu, Shijiu Jin, and Xiaorong Wu. 2007. Real time auto-focus algorithm for eye gaze tracking system. In *2007 International Symposium on Intelligent Signal Processing and Communication Systems*. IEEE, New York, 742–745. <https://doi.org/10.1109/ISPACS.2007.4445994>
- Kazushige Ooi, Keiji Izumi, Mitsuyuki Nozaki, and Ikuya Takeda. 1990. An advanced autofocus system for video camera using quasi condition reasoning. *IEEE Transactions on Consumer Electronics* 36, 3 (1990), 526–530. <https://doi.org/10.1109/30.103169>
- Lucas Paletta, Helmut Neuschmied, Michael Schwarz, Gerald Lodron, Martin Pszeida, Stefan Ladstätter, and Patrick Luley. 2014. Smartphone eye tracking toolbox: Accurate gaze recovery on mobile displays. In *Eye Tracking Research and Applications (ETRA)*. ACM, New York, 367–368. <https://doi.org/10.1145/2578153.2628813>
- Felix Putze, Johannes Popp, Jutta Hild, Jürgen Beyerer, and Tanja Schultz. 2016. Intervention-free selection using EEG and eye tracking. In *International Conference on Multimodal Interaction*. ACM, New York, 153–160. <https://doi.org/10.1145/2993148.2993199>

- Loren Shih. 2007. Autofocus survey: a comparison of algorithms. *International Society for Optics and Photonics* 6502, 0B (2007), 1–9. <https://doi.org/10.1117/12.705386>
- M.V. Shirvaikar. 2004. An optimal measure for camera focus and exposure. In *Southeastern Symposium on System Theory*. IEEE, New York, 472–475. <https://doi.org/10.1109/SSST.2004.1295702>
- Henrik Skovsgaard, Julio C. Mateo, John M. Flach, and John Paulin Hansen. 2010. Small-target selection with gaze alone. In *Eye-Tracking Research & Applications (ETRA)*. ACM, New York, 145–148. <https://doi.org/10.1145/1743666.1743702>
- Roel Vertegaal. 2008. A Fitts law comparison of eye tracking and manual input in the selection of visual targets. In *International Conference on Multimodal Interfaces*. ACM, New York, 241–248. <https://doi.org/10.1145/1452392.1452443>
- Quoc Kien Vuong and Jeong-won Lee. 2013. Initial direction and speed decision system for auto focus based on blur detection. In *International Conference on Consumer Electronics*. IEEE, New York, 222–223. <https://doi.org/10.1109/ICCE.2013.6486867>
- Chengyu Wang, Qian Huang, Ming Cheng, Zhan Ma, and David J Brady. 2021. Deep learning for camera autofocus. In *Transactions on Computational Imaging*, Vol. 7. IEEE, New York, 258–271. <https://doi.org/10.1109/TCI.2021.3059497>
- Lee S Weinblatt. 1986. Camera autofocus technique. US Patent 4,574,314.
- Aurel Wildfellner. 2012. Focus tracking for cinematography. In *Special Interest Group on Computer Graphics and Interactive Techniques (SIGGRAPH) Posters*. ACM, New York, Article 56, 1 pages. <https://doi.org/10.1145/2342896.2342966>
- Anjie Zhu, Shiwei Cheng, and Jing Fan. 2018. Eye tracking and gesture based interaction for target selection on large displays. In *Pervasive and Ubiquitous Computing and Wearable Computers*. ACM, New York, 319–322. <https://doi.org/10.1145/3267305.3267607>