International Journal of Advanced Smart Convergence Vol.13 No.3 191-198 (2024)
http://dx.doi.org/10.7236/IJASC.2024.13.3.191

IJASC 24-3-22

Applying MetaHuman Facial Animation with MediaPipe: An Alternative Solution to Live Link iPhone.

Balgum Song and Arminas Baronas

Professor, Department of International College, Dongseo University, Korea
B.A. Department of International College, Dongseo University, Korea
sbu1977@dongseo.ac.kr, baronasarminas@gmail.com

Abstract

This paper presents an alternative solution for applying MetaHuman facial animations using MediaPipe, providing a versatile option to the Live Link iPhone system. Our approach involves capturing facial expressions with various camera devices, including webcams, laptop cameras, and Android phones, processing the data for landmark detection, and applying these landmarks in Unreal Engine Blueprint to animate MetaHuman characters in real-time. Techniques such as the Eye Aspect Ratio (EAR) for blink detection and the One Euro Filter for data smoothing ensure accurate and responsive animations. Experimental results demonstrate that our system provides a cost-effective and flexible alternative for iPhone non-users, enhancing the accessibility of advanced facial capture technology for applications in digital media and interactive environments. This research offers a practical and adaptable method for real-time facial animation, with future improvements aimed at integrating more sophisticated emotion detection features.

Keywords: *Facial Motion Capture, MetaHuman, MediaPipe, Animation*

1. Introduction

Facial motion capture technology is widely used in film and television production, auxiliary teaching, visual communication, and human-computer interaction, significantly enhancing the efficiency and quality of facial animations in these fields[1]. The captured facial motion data undergoes extensive processing to remove errors and optimize quality, including data stabilization, filtering, and smoothing, ensuring accurate and realistic animations[2]. This technology captures the subtle movements of facial muscles to create lifelike and natural facial animations.

Realistic digital humans have gained popularity, particularly with the introduction of Unreal Engine's MetaHuman Creator application. MetaHumans are fully rigged for animation with a series of manipulation control points and are available for live linking to various performance capture applications. This machine-learning-assisted facial automation is becoming increasingly prevalent within the FX and gaming industries,

Manuscript Received: July. 25. 2024 / Revised: August. 1. 2024 / Accepted: August. 7. 2024
Corresponding Author: sbu1977@dongseo.ac.kr
Tel: *** - **** - ****
Professor, Department of International College, Dongseo University, Korea

with the MetaHuman Creator enabled for real-time motion capture from its early release[3].

In this paper, we aim to explore novel approaches to applying MetaHuman characters using different cameras, such as webcams, laptop cameras, and Android phones, as opposed to the MetaHuman LiveLink motion capture system that relies exclusively on iPhones. By widening the use of MetaHuman face capture to include a variety of devices, we provide more opportunities for users who wish to implement face linking systems. This research presents a variety of ways to achieve face capture, broadening accessibility and enhancing the versatility of MetaHuman facial animations.

2. Background

Motion capture (MoCap) technology in animation production has been around since the 1970s[4], initially drawing both hype and criticism as people began experimenting with computerized motion-recording devices. By the 1990s, facial motion capture systems became valuable tools for analyzing facial muscle movements, with marker-based systems becoming widely applied in the industry[5-7]..

To overcome the limitations of marker-based systems, Markerless Motion Capture (MMC) technology was developed[8]. This innovation eliminates the time-consuming procedure of placing markers, allowing motion capture experiments to be performed more conveniently[9]. MMC technology captures human motion in a more natural and lifelike manner, utilizing portable and low-cost sensors compared to traditional multi-camera systems with physical markers[10]. MMC started with the development of various algorithms and techniques for capturing human motion and 3-D objects using multi-view video, unsynchronized cameras, and image-based joint detection without physical markers.

Live Link Face UE, a commonly used markerless MetaHuman system with an iPhone, produces standardized results similar to those from expensive facial tracking tools. This system broadens research and innovation in the field by making high-quality facial capture more accessible[11].

However, there has been no research so far that can fully replace the iPhone Live Link facial motion capture system with a free or low-cost markerless alternative to MetaHuman. As facial motion capture systems become increasingly popular in various media, it is crucial to develop multiple methods to apply these systems effectively across different devices. This paper aims to explore such alternatives, widening the scope and accessibility of facial motion capture technology.

3. Experiments and Process

3.1 Initial Setup and Programming

The first step in our research involved setting up the face tracker system, which is detailed in the FaceTracker program in Figure 1. The FaceTracker class was designed to handle different input configurations, whether from a server or directly from a phone camera. The system initializes the video capture using OpenCV and configures the settings for optimal facial landmark detection using MediaPipe.

The start_camera() function configures the video capture device, while start_tracking() processes the video frames to detect facial landmarks and calculate feature positions. This function also records a neutral pose when the spacebar is pressed, ensuring baseline accuracy. The send_data() function then transmits the calculated facial feature data to the server for further processing in Unreal Engine.

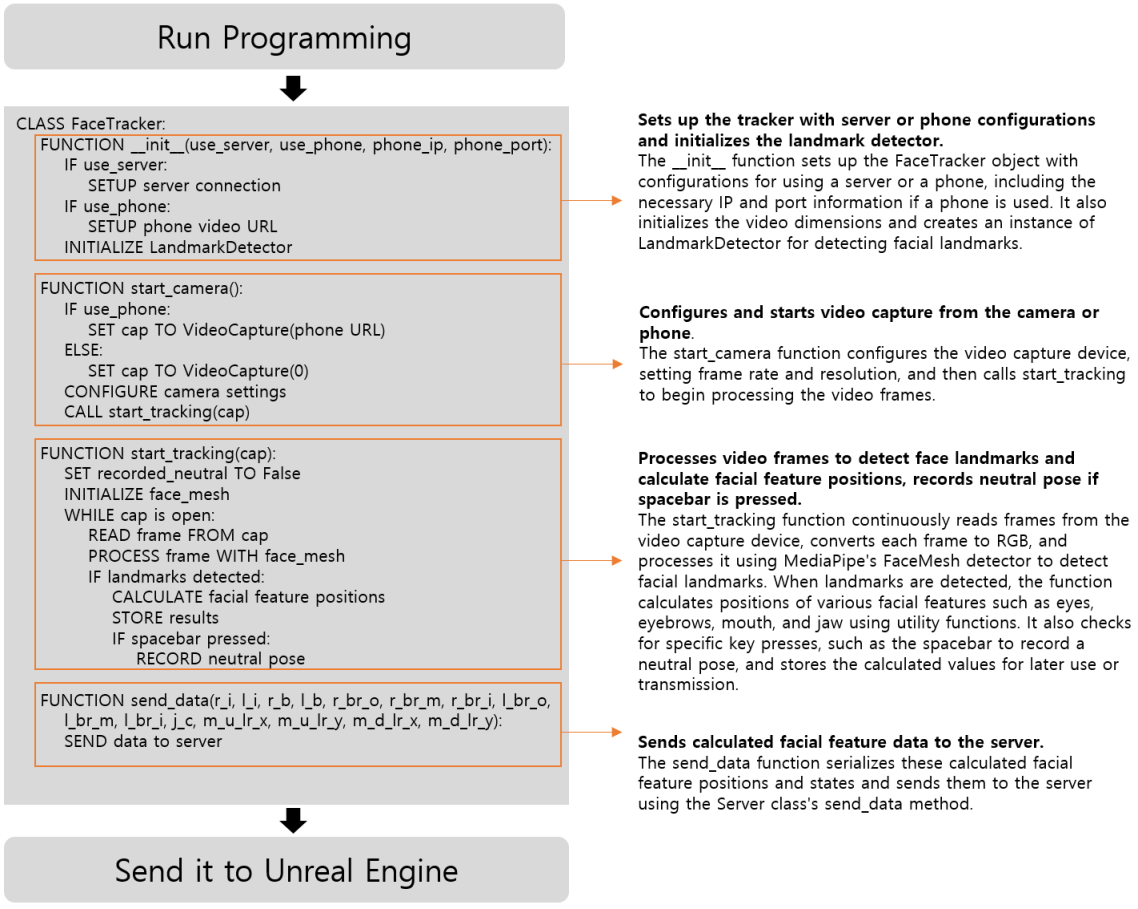

Figure 1. FaceTracker Program Workflow

3.2. Eye Aspect Ratio (EAR) Calculation

The Eye Aspect Ratio (EAR)[12], helps in determining blinks. The EAR is calculated using the distances between specific eye landmarks, as shown in Figure 2. The EAR value remains relatively constant when the eye is open and drops significantly during a blink, making it a reliable metric for blink detection.

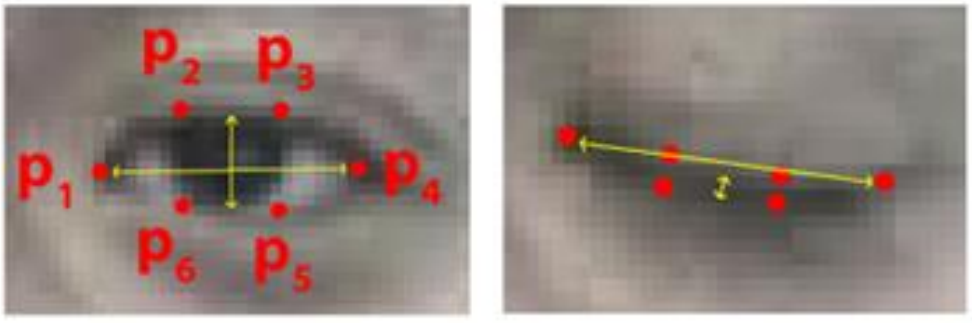

Figure 2. Eye Aspect Ratio (EAR)[12]

The formula for EAR[12] is:

$$EAR = \frac{||p_2 - p_6|| + ||p_3 - p_5||}{2||p_1 - p_4||} \quad (1)$$

Where $p_1, p_2, \ldots, p_6$ are the 2D coordinates of specific eye landmarks. The numerator represents the sum of the vertical distances between the eye landmarks, and the denominator is the horizontal distance. This ratio remains relatively constant when the eye is open and decreases significantly when the eye is closed, making it useful for detecting blinks.

3.3. Smoothing Data with One Euro Filter

To ensure the facial motion data is smooth and realistic, we applied the One Euro Filter algorithm[13], which is based on adaptive exponential smoothing. The filter balances noise reduction and responsiveness to rapid changes by adjusting the cutoff frequency dynamically. Figure 3 illustrates the effect of the One Euro Filter on our data, showing improvement in stability.

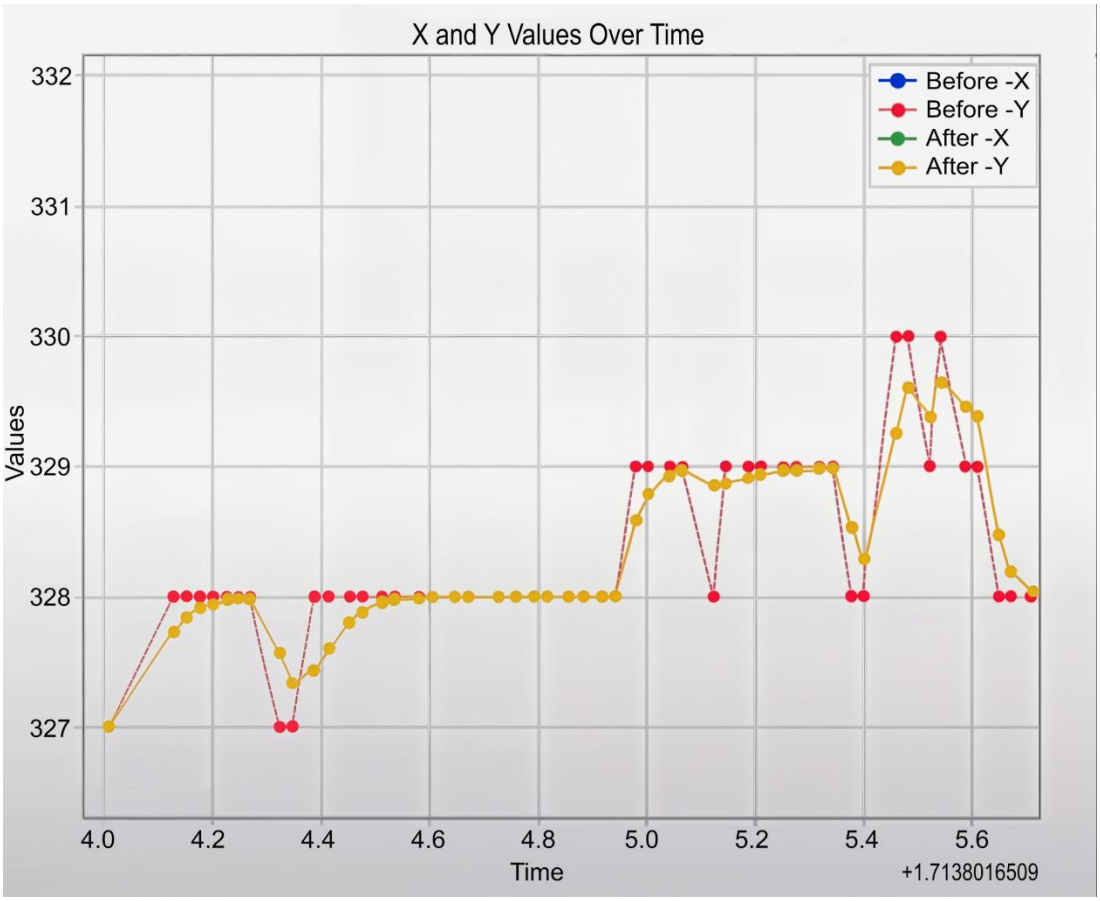

Figure 2. Euro Filter X and Y values over time

The low-pass filter is the core of the One Euro Filter, designed to smooth the input signal by averaging it with its previous values and expressed in Equation (2).

$$lowpass_{\alpha}(x_t, x_{t-1}) = \alpha \cdot x_t + (1 - \alpha) \cdot x_{t-1} \quad (2)$$

x_t is the current signal value at time t, x_{t-1} is the previous signal value, and α is the smoothing factor that determines how much of the new signal value is considered in the smoothed result. It balances between the current and previous values to smooth out rapid changes.

The smoothing factor α determines how much influence the current and previous signal values have on the filtered output. It is dynamically calculated to adapt to the rate of change in the signal. The formula for the smoothing factor is expressed in Equation (3).

$$\alpha = \frac{1}{1+\tau \cdot f_c} \quad (3)$$

τ is the time constant, controlling the responsiveness of the filter. f_c is the cutoff frequency, determining the filter's sensitivity to changes in the signal. A higher cutoff frequency f_c results in a smaller α, making the filter more responsive to changes, while a lower f_c increases α, leading to more smoothing. The smoothing factor adjusts how quickly the filter reacts to changes in the input signal. A higher cutoff frequency makes the filter more responsive, while a lower cutoff frequency results in more smoothing

The dynamic cutoff frequency f_c adjusts based on the rate of change in the signal, allowing the filter to balance between noise reduction and responsiveness. The equation for the dynamic cutoff frequency is as follows (4).

$$f_c = f_{min} + \beta \cdot |dx| \quad (4)$$

f_{min} is the minimum cutoff frequency, ensuring a base level of smoothing. β is a parameter that adjusts the influence of the rate of change. |dx| is the absolute rate of change of the signal, representing how quickly the signal values are changing.

This dynamic adjustment allows the filter to balance between noise reduction and responsiveness to rapid changes. When the signal changes rapidly, the cutoff frequency increases, making the filter more responsive. Conversely, when the signal changes slowly, the cutoff frequency decreases, resulting in more smoothing.

3.4. Integrating Data into Unreal Engine

The final step involved integrating the processed data into Unreal Engine to animate the MetaHuman character. The data flow from MediaPipe and Python to Unreal Engine is depicted in Figure 4. The data is received and parsed into arrays, then used to set various facial feature values. Unreal Engine's Blueprint system updates the MetaHuman character's facial expressions in real-time based on these values.

The Blueprint system processes these values every two frames to ensure smooth and accurate facial animations. This integration allows for a wide range of facial expressions to be captured and animated in real-time, providing a robust alternative to the Live Link iPhone system.

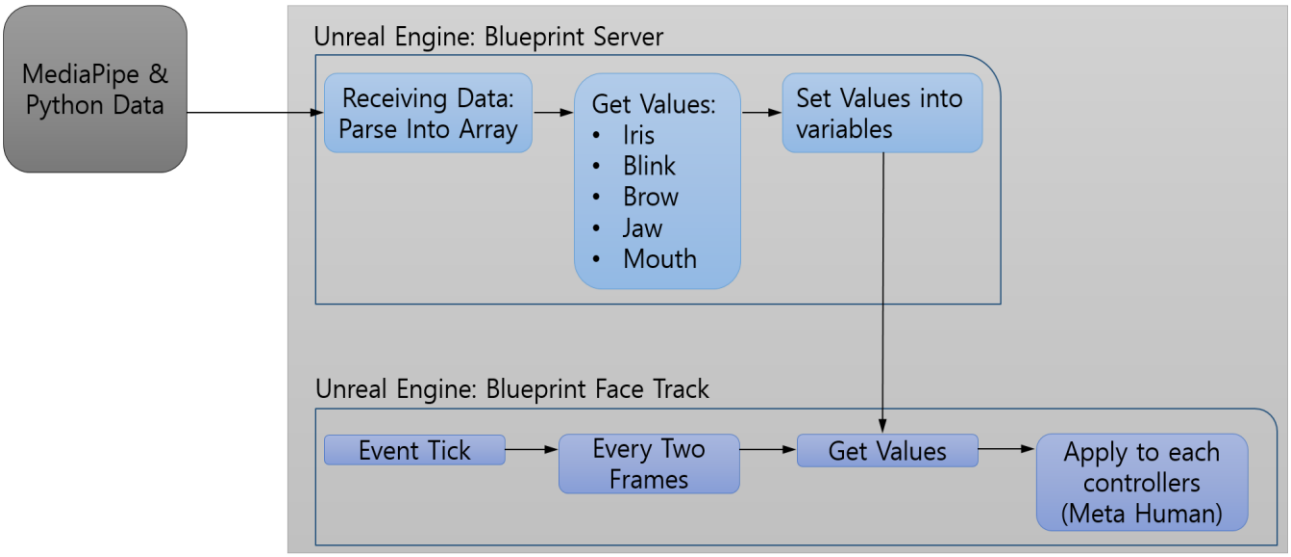

Figure 3. Data to Unreal Engine flow chart

4. Results and Discussion

The implementation of MediaPipe to apply facial landmark detection across various camera devices—such as webcams, laptop cameras, and Android phones—has demonstrated promising results in animating MetaHuman characters in Unreal Engine. Figure 5 illustrate the outcomes of our experiments, where the first and sixed column shows the real-time captured data from a computer webcam, and the subsequent columns depict the applied facial expressions on four different types of MetaHuman characters. We expressed variety of emotions;neutral, look down, look up, blink, mouth open, jaw open, eyebrows up, look left.

Figure 4. Webcam facial expressions with four different MetaHuman character results

Initially, the concept of controlling MetaHuman characters using facial landmarks seemed challenging. The primary difficulty lay in designing a MetaHuman rig compatible with MediaPipe's outputs. The most complex aspect was tracking the iris, given its intricate movements and the need for precise detection when looking up or down. Occasionally, the system misinterpreted eye movements, mistaking closed eyelids for iris movement, which remains an area needing improvement.

The Eye Aspect Ratio (EAR) method proved effective for eyelid detection, but this occasionally conflicted with accurate iris tracking. Eyebrow movements, although functional, are currently limited to simple emotions and movements. The distance-based method used for eyebrow tracking was chosen for its efficiency and ease of implementation, but further refinement is needed for more nuanced expressions.

Mouth movement presented significant challenges in replicating realistic animations. Differentiating between mouth opening and jaw opening requires more sophisticated calculations. A threshold system based on averaged maximum and minimum values from multiple subjects was implemented to accommodate varying facial structures. However, this approach needs further customization to ensure accurate tracking for individual users.

The experiments and results indicate that our method for capturing and applying facial emotions to MetaHuman characters using MediaPipe and Unreal Engine is both effective and practical. It provides a versatile, real-time, and accurate solution that can be used with a variety of camera devices, making it accessible to a wider range of users and applications. Further research could explore the integration of additional features, such as enhanced emotion detection and more complex facial animations, to further improve the system's capabilities.

5. Conclusion

This research successfully demonstrates an alternative approach to applying MetaHuman facial animations using MediaPipe, providing a flexible and accessible solution substitute to the Live Link iPhone system. By utilizing various camera devices such as webcams, laptop cameras, and Android phones, we broaden the scope and accessibility of high-fidelity facial motion capture technology. Our method integrates facial landmark detection through MediaPipe with real-time animation capabilities in Unreal Engine’s Blueprint system, ensuring accurate and responsive facial expressions.

The experimental results show that key techniques, including the Eye Aspect Ratio (EAR) for blink detection and the One Euro Filter for data smoothing, are effective in delivering realistic and lifelike animations. This approach not only maintains the quality of facial animations but also reduces the dependency on specific hardware, making it a cost-effective solution suitable for a wide range of applications in digital media and interactive environments.

While the current system provides a flexible alternative to the Live Link iPhone for MetaHuman facial animation, ongoing efforts are required to perfect the technology and expand its capabilities. The ultimate goal is to create a robust, user-friendly tool that delivers high-fidelity facial animations for various applications in digital media and interactive environments.

References

[1] Yihao Zhang, Xiangzhen He, Yerong Hu, Jia Zeng, Huaiyuan Yang, and Shuaihang Zhou, “Face animation making method based on facial motion capture,” in 2021 IEEE International Conference on

Emergency Science and Information Technology (ICESIT), pp. 84-88, IEEE, 2021. DOI: https://doi.org/10.1109/icesit53460.2021.9696547.

[2] Xiaoting Wang, Lu Wang, and Guosheng Wu, "Body and Face Animation Based on Motion Capture," International Journal of Information Engineering and Electronic Business, Vol. 3, No. 2, pp. 28, 2011. DOI: https://doi.org/10.5815/ijieeb.2011.02.04.

[3] Joel McKim, "Animation without animators: from motion capture to MetaHumans," Animation Studies 2.0, 2022.

[4] Chris Bregler, "Motion capture technology for entertainment [in the spotlight]," IEEE Signal Processing Magazine, Vol. 24, No. 6, pp. 160-158, 2007. DOI: https://doi.org/10.1109/msp.2007.4317482.

[5] Nicole Dagnes, Federica Marcolin, Enrico Vezzetti, François-Régis Sarhan, Stéphanie Dakpé, Fréderic Marin, Francesca Nonis, and Khalil Ben Mansour, "Optimal marker set assessment for motion capture of 3D mimic facial movements," Journal of Biomechanics, Vol. 93, pp. 86-93, 2019. DOI: https://doi.org/10.1016/j.jbiomech.2019.06.012.

[6] Bernd Bickel, Mario Botsch, Roland Angst, Wojciech Matusik, Miguel Otaduy, Hanspeter Pfister, and Markus Gross, "Multi-scale capture of facial geometry and motion," ACM Transactions on Graphics (TOG), Vol. 26, No. 3, pp. 33-es, 2007.

[7] Demetri Terzopoulos and Keith Waters, "Techniques for realistic facial modeling and animation," in Computer Animation'91, pp. 59-74, Springer Japan, 1991. DOI: https://doi.org/10.1007/978-4-431-66890-9_5.

[8] Stefano Corazza, Lars Mündermann, Emiliano Gambaretto, Giancarlo Ferrigno, and Thomas P. Andriacchi, "Markerless motion capture through visual hull, articulated icp and subject specific model generation," International Journal of Computer Vision, Vol. 87, pp. 156-169, 2010. DOI: https://doi.org/10.1007/s11263-009-0284-3.

[9] M. Rahul, "Review on motion capture technology," Global Journal of Computer Science and Technology, Vol. 18, No. 1, pp. 23-26, 2018.

[10] Bradley Scott, Martin Seyres, Fraser Philp, Edward K. Chadwick, and Dimitra Blana, "Healthcare applications of single camera markerless motion capture: a scoping review," PeerJ, Vol. 10, e13517, 2022. DOI: https://doi.org/10.31219/osf.io/2e8nz.

[11] Carlos Vilchis, Sharon Ramírez, Armando Rodríguez, and Miguel Gonzalez, "Driving the future faces: Benchmarking state-of-the-art facial tracking technology for Digital Humans," 2022.

[12] Jan Cech and Tereza Soukupova, "Real-time eye blink detection using facial landmarks," Cent. Mach. Perception, Dep. Cybern. Fac. Electr. Eng. Czech Tech. Univ. Prague, pp. 1-8, 2016.

[13] Géry Casiez, Nicolas Roussel, and Daniel Vogel, "1€ filter: a simple speed-based low-pass filter for noisy input in interactive systems," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2527-2530, 2012. DOI: https://doi.org/10.1145/2207676.2208639.