Developing and Evaluating An Adaptive User Interface For Mobile Devices, Controlled Via Head Gestures

James Whiffing
jw204@bath.ac.uk
University of Bath - Department of Computer Science
Bath, England

ACM Reference Format:
James Whiffing. 2022. Developing and Evaluating An Adaptive User Interface For Mobile Devices, Controlled Via Head Gestures. In *Proceedings of (Conference acronym 'XX).* ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

There are attempts to introduce an additional modal of interaction with smart devices utilising the user's face. These exist on a spectrum with regards to interaction techniques: Using the face as a pointer, typically based on the movement/position of the user's nose; detecting gestures based on the movement/pose of the user's face; and a combination of the two.

A common issue that afflicts many of these systems/approaches is that they don't distinguish between the movement of the phone or the movement of the user's head. For example, the user moving their head to the left, will be treated the same as the user moving the phone to the right, since from the front-facing camera's perspective it looks like the head is moving in the same wayThis reduces the number of 'recognisable' gestures.

We look to explore whether such a system could distinguish between the user moving their head vs the phone being moved.

In order to develop such a system, a data driven approach was taken. As such a study was undertaken to collect the camera feed and IMU/Gyro of the smart device, an IMU within an earable worn by the user, and 3D positioning of the user's head and the smart device via a motion capture stage. With the motion capture data being synced to the IMU/Gyro data and photos, a system could be trained to recognise several gestures and learn to distinguish between whether an observed gesture was due to the phone or the user's head moving.

2 LITERATURE REVIEW

2.1 Head Gestures

Gestures can be classified into 5 classes[6]:

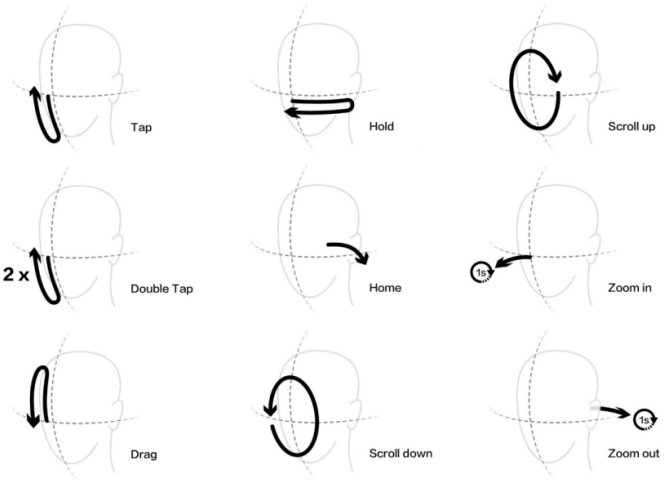

Figure 1: Proposed head gestures and their corresponding actions[12].

Dietic These are gestures that involve pointing, and mapping this to either a specific object or location within the interface.
Manipulative A gesture which indicates intent to manipulate some object.
Semaphoric Gestures which map to a specific action or intent.
Gesticulative Typically accompany speech, but not required to transfer meaning.
Language A substitute to written or verbal language.

For the purposes of this project, we shall be intending to review Dietic, Manipulative, and Semaphoric gestures to support user interaction.

One of the simpler ways to track the movement of a user's head is with an Inertial Measurement Unit (IMU), as this is a physical device that can be used to measure rotational and linear acceleration[1].

An example of this can be seen in the work of Yan et al. who propose 9 gestures (Figure 1) and utilise the IMU embedded within the Hololens[12].

The gestures they propose were derived from a study wherein they asked participants to suggest head movements that they believed corresponded to the action taken. These were then collated by manually into 80 gestures, which were then effectively voted upon by the participants for their respective actions. The gestures

[1]Linear acceleration is typically less accurately tracked compared to angular acceleration

Conference acronym 'XX, N/A, N/A

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

with the most votes for a given action were selected, with some minor adjustments to ensure there were no clashes between actions.

To extract the gestures from the Hololens, the IMU output was segmented via detection of acceleration (20 degrees per second) and deceleration (4 degrees per second), not exceeding 2 seconds.

Feature extraction is performed with Dynamic Time Warping (DTW)[1], followed by a Support Vector Machine (SVM) classifier to classify the observed gesture into one of 9 the categories, or unintentional movement. With just the DTW they were able to achieve 90% accuracy, but with the SVM they were able to boost this to 97%.

To evaluate the head gestures, they compared them with existing hand gestures. They found that head gestures caused more fatigue and generally felt less natural, while being equivalent or better with regards to learnability.

Using a mobile device we won't have access to an IMU on the user's head, however we will be able to try and utilise the same set of gestures, and to use a similar approach to track the phone's movement, which could allow us to try and differentiate between the phone or head being moved.

An alternative approach is to try and extract the face using an RGB camera.

One way to do this was developed by Gorodnichy, which 'finds' the nose under the assumption that it should have the greatest intensity gradient since it should always be closest to the camera, and given it is convex in nature it should be the 'brightest' feature[4].

The tracked point isn't a specific point on the nose (e.g. the tip), but a point that can move across the surface of the nose, based on what is closest to the camera.

They go on to extend this work with the usage of the user's nose to control a pointer on their screen[5]. They extract just the nose since it meets their two requirements for a trackable feature:

(1) It is always visible, presuming the user is facing within 180 degrees towards the camera.
(2) Only one feature should be used to define the cursor, to reduce/eliminate potential jitter.

To click with the 'Nouse' the user blinks twice within short succession. Blinking is determined by reviewing the change in the sequence of 3 frames.

To ensure the system is realtime they use a reduced resolution, and could find that a resolution of 160x120 was robust enough to accurately track nose, and map the cursor to an accurate location on the screen.

They make claims about accuracy and enjoyment, but relevant data not provided. They only seemed to make statements suggesting Nouse was as good as, if not better than, typical mouse control. They claim mouse usage caused wrist ache, but movement of entire head doesn't present neck ache, which was reported in the IMU head tracking describe4d above[12].

Another nose controlled cursor is presented by Varona et al., however they use Haar cascades to extract the region containing the face, within which they use a similar technique as above[4] to extract points for the corners of the nose, or the nostrils[11].

To detect eyes, the system determines the user's skin colour by sampling the pixels within the detected face region. They then presume the eyes will be a different colour, and as such filter based on the extracted skin-tone. They then select the features closest to the nose, that are symmetrical.

A UI is provided with possible actions as buttons. The user moves the cursor to the action they wish to perform, then wink (with either eye) to select it. When they then fixate on part of the screen (move the cursor to a point and keep it stationary), the action will be performed.

Only evaluated for click recognition and accuracy of where the click was performed within a grid of points. However >80% accuracy even for users with no training time, just instructions.

Moving closer to a tool for mobile devices we have the work of Roig-Maimó et al., who use the front faced camera to scroll, using the head angle w/r/t the device as direction of scrolling.[10]

They extend upon the work of Varona et al.[11] the nose of the user and to correlate it's motion to a virtual cursor on the screen.

Selection/tapping is performed via tapping anywhere on the screen, however the tap will actually occur under the virtual cursor.

They evaluated the system by asking users to select elements of varying sizes, phone held in different orientations (portrait vs landscape), and with varying gain applied to the velocity of the cursor in response to head movement.

Elements below 88x88pt[2] were found to be less successfully selected, this is primarily due to the low resolution used for the gesture tracking being unable to be mapped to a finer resolution on the device screen. Potentially increasing the resolution used for the tracking could permit finer accuracy.

They do not distinguish between the user moving the phone, or the user moving their head. It could be seen as a feature, either move the phone or head to scroll, but this would be intersting to try and distinguish, to potentially support additional actions/gestures.

The above systems describe the ability to identify where on a screen the user is looking, or at the very least intends to perform some action, through providing them with a virtual cursor they can manipulate via moving there head, either through rotation or physically moving the head. We can look to extend upon this to understand where the user's attention may be focused.

Some smartphones now also include front-facing depth cameras, a technique for tracking a user's head, and detecting head/facial features is provided by Deepateep and Vichitvejpaisal, wherein they utilised the ARKit Framework for iOS[2].

Objective similar to works described above to control a virtual cursor, however instead of specifically using the nose, they are using the perceived pose of the user's entire head.

Cursor motion is tracked based on head pitch and yaw in the Y and X directions respectively. Requires user to directly face the camera for zero movement of the cursor.

Additionally they combine this control with facial gestures, which perform specific actions, or permit the beginning of specific actions, such as zoom, drag, and tapping. These utilised poses obtained from the eyebrows and mouth. Timings specific to each action, some gestures overloaded based on timing.

A depth camera affords greater accuracy for tracking (particularly for understanding the distance from the screen), and in case

[2]Apple Point, effectively 2 pixels on a retina display, so 88pt == 176px

of iPhone there is consistency with specific hardware. However for the general smartphone population, specifically android, depth cameras aren't standard, and when present can have different hardware. For our project we will presume 3D depth cameras aren't available.

2.2 Adaptive Interfaces

2.2.1 3D Interfaces. A 3D interface presumes that there is a virtual screen through which a portion is visible from the mobile device.

Francone and Nigay approaches this by developing a system which adjusts 3D content on the screen based on the user's perspective/orientation relative to the phone screen/front facing camera[3].

Their system utilises Haar cascades to extract the user's face. The X and Y positions are tracked via the centre of the observed bounding box. Here they intentionally accept both movement of either the head or device. Depth is estimated via the size of the region, however this fails if the user's face starts to go out-of-frame of the camera, as the a portion of the face may still be recognise, resulting in a thin bounding box.

For their interface they tested:

- Displaying 3D interface elements, e.g. adding depth to the interface such that adjusting perspective would permit viewing the sides of elements within the UI.
- A workspace that was too large to fit onto the screen, and to reveal other elements of the interface you could adjust the perspective, rather than say changing the page.

For their evaluation they did not compare with existing techniques, such as touch, but rather asked participants to directly evaluate the usability in isolation.

We can learn from Francone and Nigay regarding the depth axis. To alleviate the issue wherein the face is partially out-of-frame we could adjust the face detection to require the whole face, or to understand how much of a face has been detected. We can also follow a similar evaluation procedure, however it would be beneficial to also compare to an existing interface.

Another approach is developed by Miyazaki and Komuro, however they approach this from the perspective of the phone moving to expose a virtual display[8].

They developed a virtual display (a map within which several 'pins' are present), which could either be viewed as static and hovering in-front of the user, or placed upon a 2D plane in their environment. To move around the virtual display, they simply need to move the device. To do this they utilised Google's Ar toolkit: Tango. The rear camera of the device would be used by Tango to track the device, and to track the position of the virtual display, either with respect to a 2D plane, or the user.

They evaluated these two displays against a touch-only display, which could only be manipulated by touch. They found that the touch only implementation was the slowest for finding various the pins. Both virtual display interfaces were found to require less mental load (like recalling the direction of various points on the map from the current position), and intuitive to use when compared with touch only, though the non-AR instance was found to be more taxing to use.

For our project we can look into also utilising some AR features to help track the phone, and differentiate the movement of the head vs movement of the phone. However we would need to investigate power usage and processing limits if we are trying to perform AR tracking and head tracking, using both front and rear cameras.

2.2.2 Context Aware UI. Where the 3D UI allows a user to move around and adjust the perspective of the UI elements, an alternative approach would be to instead change which UI elements are on screen in reaction to the user's attention or perceived intentions.

An example of this by Pfeuffer et al. is a UI for an AR head mounted display which adjusts the displed content and presented information based on user's gaze[9].

This is primarily controlled via user gaze, specifically dwell-time, however is also informed by the task context. For example in their conversational UI, information about a person is presented around them, in the background, however if the user were to dwell upon specific information, it will bring it to the foreground, and display more (if applicable), until they revert their gaze back upon the person. In the tree of life example (a network graph for a subset of the tree of life is rendered), there is no 'background', instead gaze is used to adjust the level of information for specific nodes. This is also the case for the shopping interface, where gaze permits information to display for specific items, and the ability to interact with said items.

They evaluated their interfaces with different amounts dwell-timerequired to adapt/accept user input, to compare the accuracy and error rate. With shorter dwell times task completion was faster, as expected, but the error rate much higher. They found that a 3 sec dwell-time was least error prone (in a range of 1-4s). However the majority of participants found the dwell-time was too slow for anything beyond 2s.

We can't use this directly, as we won't be using gaze, but we could adapt to work with a cursor controlled by the user's head.

Another interpretation of this is the interface developed by López et al., which is a virtual display that is in the shape of a concave box, from which the visible part of the box is based on the user's perspective.[7]. This is similar to the work of Francone and Nigay, however specifically for extending workspaces, and for supporting adaptive UI elements which react to the user's head position.

One interface is described as being a concave box, from which interface elements are placed. When the user places attention to a particular element, it is brought into the foreground for use, and can then be dismissed for them to select another element. Another interface they describe permits head gestures to be recognised which map to revealing additional UI elements based on user intent. Their example involves a web-page, wherein turning/moving in from the right would reveal a prompt to add the page to the bookmarks, and continuing the motion would open the bookmark dialogue.

The head is extracted via an Adaboost classifier which determines the weighting of a 20x20px neighbourhood of pixels as containing facial features, in similar manner to Haar cascade. head distance is then estimated via change in scale of the extracted region, so no definitive size, but permits a relative scale/distance from original size.

Due to limited Field of View (FOV) of the camera, they added a wide-angle lens to increase the FOV to 160 degrees.

Unfortunately they don't actually implement an instance of any of their proposed interfaces, which is something we can look to achieve and evaluate.

3 DATA COLLECTION METHODOLOGY

3.1 Data Collection Application

3.2 Study

3.3 Data Post-Processing

4 SYSTEM DEVELOPMENT

4.1 Evaluation

5 RESULTS

6 DISCUSSION

7 FURTHER WORK

8 CONCLUSION

REFERENCES

[1] Donald J Berndt and James Clifford. 1994. Using dynamic time warping to find patterns in time series.. In *KDD workshop*, Vol. 10. Seattle, WA, USA:, 359–370.

[2] Chatsopon Deepateep and Pongsagon Vichitvejpaisal. 2020. Facial Movement Interface for Mobile Devices Using Depth-sensing Camera. In *2020 12th International Conference on Knowledge and Smart Technology (KST)*. IEEE, 115–120.

[3] Jérémie Francone and Laurence Nigay. 2011. Using the user's point of view for interaction on mobile devices. In *Proceedings of the 23rd Conference on l'Interaction Homme-Machine*. 1–8.

[4] Dmitry O Gorodnichy. 2002. On importance of nose for face tracking. In *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*. IEEE, 188–193.

[5] Dmitry O Gorodnichy and Gerhard Roth. 2004. Nouse 'use your nose as a mouse' perceptual vision technology for hands-free games and interfaces. *Image and Vision Computing* 22, 12 (2004), 931–942.

[6] Maria Karam et al. 2005. A taxonomy of gestures in human computer interactions. (2005).

[7] Miguel Bordallo López, Jari Hannuksela, Olli Silvén, and Lixin Fan. 2012. Head-tracking virtual 3-D display for mobile devices. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 27–34.

[8] Masashi Miyazaki and Takashi Komuro. 2021. AR Peephole Interface: Extending the workspace of a mobile device using real-space information. *Pervasive and Mobile Computing* 78 (2021), 101489.

[9] Ken Pfeuffer, Yasmeen Abdrabou, Augusto Esteves, Radiah Rivu, Yomna Abdelrahman, Stefanie Meitner, Amr Saadi, and Florian Alt. 2021. ARtention: A design space for gaze-adaptive user interfaces in augmented reality. *Computers & Graphics* 95 (2021), 1–12.

[10] Maria Francesca Roig-Maimó, Javier Varona Gómez, and Cristina Manresa-Yee. 2015. Face Me! Head-tracker interface evaluation on mobile devices. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. 1573–1578.

[11] Javier Varona, Cristina Manresa-Yee, and Francisco J Perales. 2008. Hands-free vision-based interface for computer accessibility. *Journal of Network and Computer Applications* 31, 4 (2008), 357–374.

[12] Yukang Yan, Chun Yu, Xin Yi, and Yuanchun Shi. 2018. Headgesture: hands-free input approach leveraging head movements for hmd devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 4 (2018), 1–23.

A PROJECT PLAN

A.1 Research and Development

A.2 Studies