November 2020

Optical Accessory to Add Touch Capability to a Non-touchscreen Device

D Shin

Shirley Xue

Optical Accessory to Add Touch Capability to a Non-touchscreen Device

ABSTRACT

Certain laptops and other devices do not include built-in touchscreen capability. This disclosure describes an optical accessory that enables such devices to recognize touch inputs. The accessory includes a fisheye lens attached to a holder, and a tilt plane mirror placed within the accessory that reflects incident light towards the camera of the device to which the accessory is attached. A one-time calibration is performed at a setup time of setup and at runtime, camera images that capture the user's finger position relative to the screen are received. A neural finger pose is estimated by detecting fingers and utilizing a skeletal finger pose detector. A touch event is determined based on an Euclidean distance between the tip joint coordinates of real and virtual fingers. The touch coordinate in the optical view is determined and is mapped to a display coordinate using a conformal mapping.

KEYWORDS

- Touch capability
- Periscopic mirror
- Tilt plane mirror
- Fish-eye lens
- Finger position
- Pose estimation
- Conformal mapping
- Hungarian matching algorithm

BACKGROUND

Commodity laptops come with large screen displays without touchscreen capability (such laptops do not have a touch sensing panel underneath the display) due the additional cost. Such laptops rely on other input mechanisms such as trackpads. However, there are several key applications that can benefit from touch capabilities on a laptop screen directly. For example, illustrators and visual content creators can draw accurate images with a stylus or finger by directly touching the screen. Also, a lot of touch shortcuts such as “pinch-to-zoom” or “slide-to-wake” are more intuitive to be performed directly on the screen than indirectly on a trackpad.

DESCRIPTION

This disclosure describes a low-cost optical accessory to enable a touch-based user interface for computing devices that do not include a touchscreen. The optical accessory as described herein can be attached to a computing device such as a laptop and can be activated by the user. Upon activation, the accessory provides touch sensing capabilities even in the absence of explicit additional touch sensing hardware behind the display screen.

The optical accessory includes a 90-degree plane mirror such that to capture user touch activity on the screen using a device camera, e.g., a front-facing camera in the top bezel of a laptop. Captured images are algorithmically translated into touch events that can be used to trigger suitable actions. Activation of the camera and accessory, and the capture of user touch input is performed with specific user permissions, and the user is provided with options to disable the accessory and associated image capture at any time.

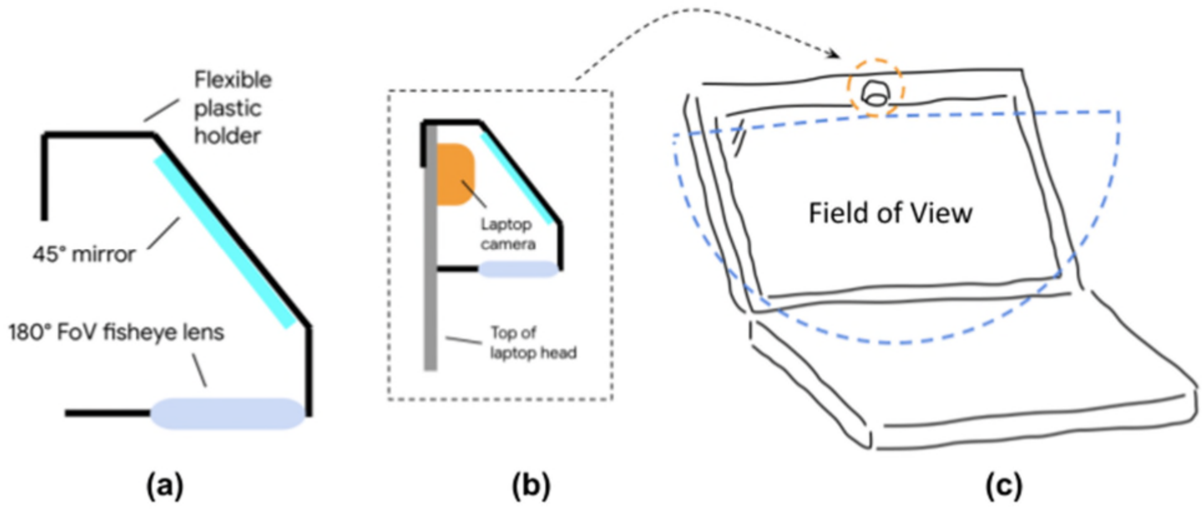


______**Fig. 1: (a) optical accessory; (b) mounted on laptop; (c) field of view**

Fig. 1 depicts an example optical accessory and its use for detection of touch inputs, per techniques of this disclosure. Fig. 1(a) depicts a schematic showing the arrangement of components of the optical accessory. As depicted, the accessory includes a flexible plastic holder that can be clamped on top of a computing device such as a laptop, desktop monitor, etc. A fisheye lens, e.g., with a 180° field-of-view (FoV), is attached to the holder towards the bottom side of the holder. A 45° tilt plane mirror is placed within the accessory such that it reflects incident light from the direction of the fisheye lens towards the device camera.

As depicted in Fig. 1(b), when mounted to a laptop display, the optical accessory enables the camera to receive images of the display in a top-down view (planar view) along the display plane. Fig. 1(c) depicts an example field of view provided by the optical accessory, which covers the entire screen.

When there is no touch event (the user's fingers are not near the display screen), the camera images only include views of the display and are discarded as lacking touch input. When the user's fingers are close to the display screen, the camera images include views of the user's fingers. Accurate touch detection needs determination of whether placement of the user's fingers are indicative of a hover of the user's fingers near the display screen, or whether a touch event has occurred. For example, a user may be pointing to content on the screen, which should not trigger detection of a touch event. Further, while the foregoing description refers to fingers, a touch operation performed with a single finger can also be recognized by the accessory in the described arrangement.

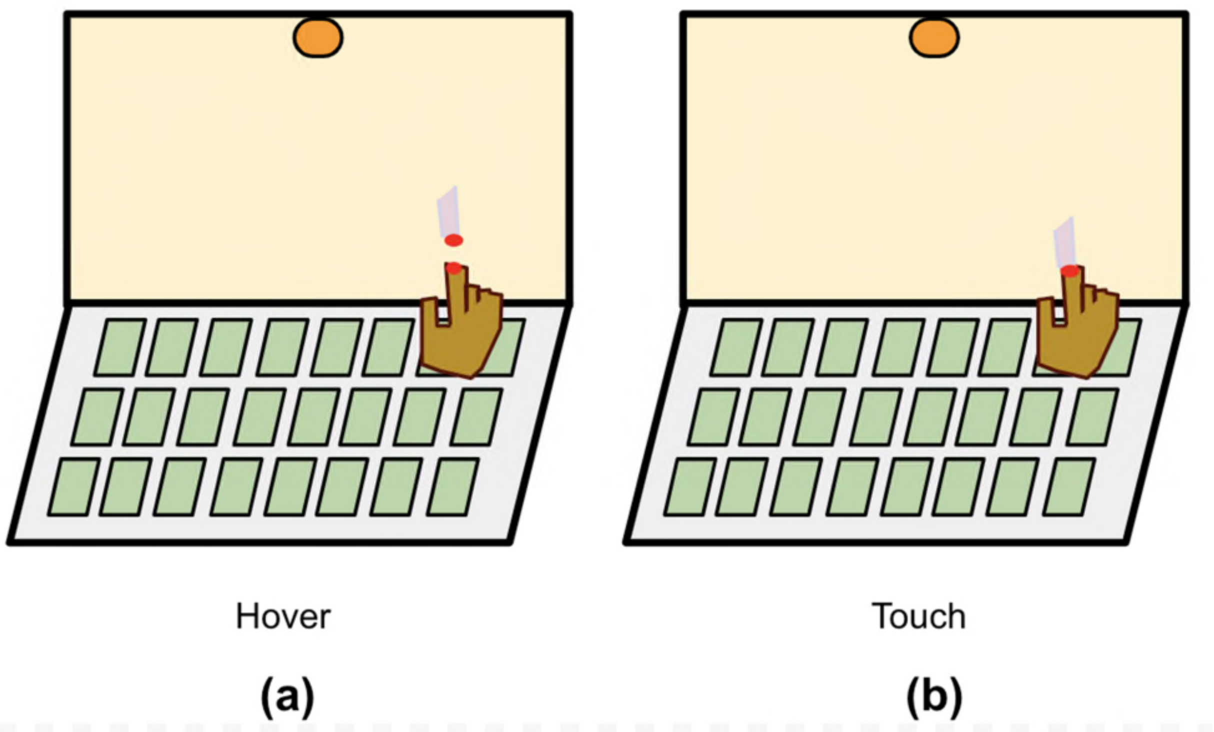
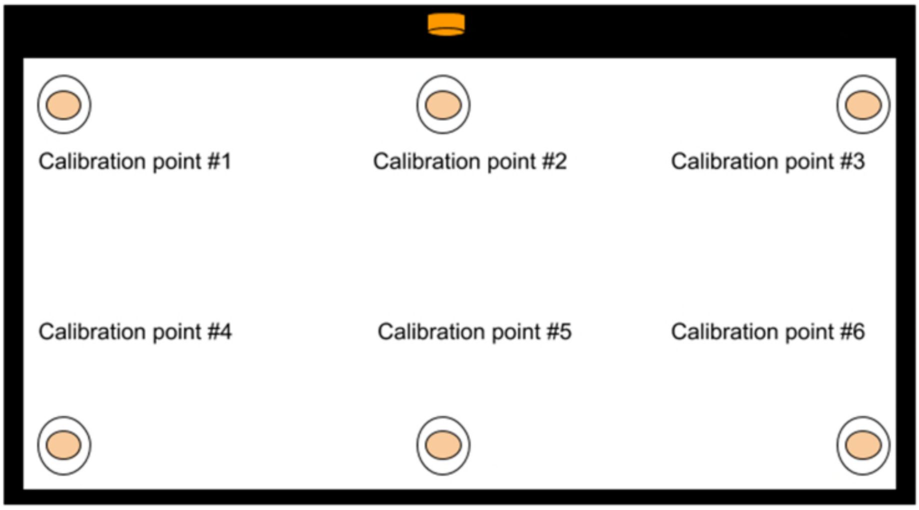
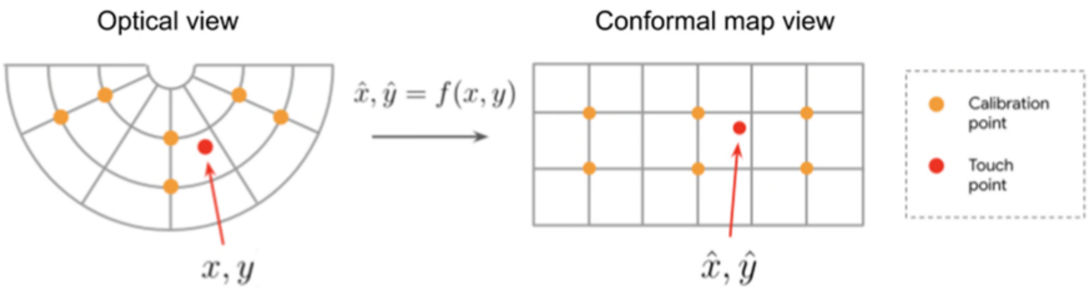


Fig. 2: Hover versus touch

Fig. 2 depicts an example of touch detection, per techniques of this disclosure. The partially-reflective property of the display screen is utilized to determine if a touch event has occurred. As depicted in Fig. 2(a), camera images of a hovering finger reveal two objects: the real finger, as well as its reflection from display. When the user's finger is hovering over the screen, the respective fingertips, indicated by the two red dots in Fig. 2(a), of the actual finger and the reflection do not touch. However, when the user actually touches the screen, the two fingertips meet at a unique point indicated by the single red dot in Fig 2(b). The distance between the real and reflected fingertips is utilized to enable robust touch sensing, even in the absence of a physical touch sensor.



(a)



(b)

Fig. 3: Calibration and conformal mapping

Fig. 3 depicts example calibration of a touch event detection system, per techniques of this disclosure. Calibration is performed by the user (with on-screen guidance provided via software) to associate different user touch locations (points) on the screen with specific pixel points on the computing device. Calibration enables accurate determination of display coordinates based on the estimated touch coordinates.

As depicted in Fig. 3(a), the user is requested to touch six points located near the boundaries of the screen. The fisheye lens is optimized to have maximal field-of-view and thus, has an optical view that is cone-shaped, as depicted in Fig. 3(b). The cone-shaped touch manifold is transformed into a rectangular manifold that describes the display pixels by utilizing a conformal mapping, an angle-preserving undistortion function. Fig. 3(b) depicts the six calibration points and an example touch point before and after the conformal mapping.

The conformal mapping is represented by a complex-log operator that has real-imaginary argument amplitudes as tunable hyperparameters, as provided below:

$$
\boxed{\begin{aligned} &v = \log(\alpha_x x + i \cdot \alpha_y y) \\ &\hat{x} = \Re(v),\ \hat{y} = \Im(v) \end{aligned}}
$$

Tuning of the hyperparameters ensures accurate estimation of the touch coordinates. An optimization that maximizes the log-likelihood given the input-output pairs of the conformal mapping is performed to determine optimal hyperparameters based on the calibration and is saved locally on the computing device for subsequent runtime usage.

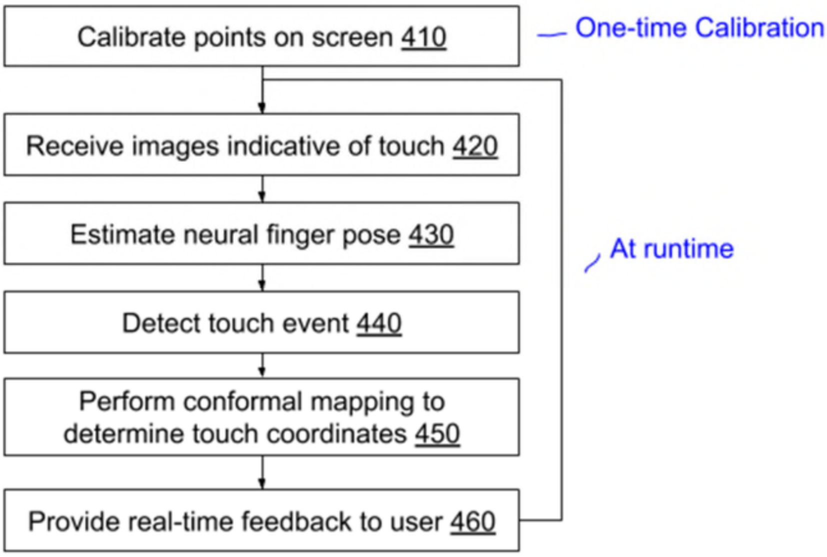


Fig. 4: Optical enablement of Touch event detection

Fig. 4 depicts an example workflow for enablement of touch event detection using an optical accessory, per techniques of this disclosure. A one-time calibration is performed (410) as described above at a time of setup to calibrate points on the screen. At runtime, with user permission, camera images of the field of view of the camera via the optical accessory are received (420). A likely touch event is detected based on the camera images that are indicative of finger proximity to the display screen.

A neural finger pose is estimated (430) by detecting human fingers in the images and by utilizing a skeletal finger pose detector. The resulting image includes labels of different finger joints for both the real finger and the reflected finger. A touch event is determined by calculating (440) the Euclidean distance between the tip joint coordinates of real and virtual (reflected) fingers. If the Euclidean distance meets a predetermined threshold of proximity, it is determined that the user's finger is physically touching the display.

The touch coordinate in the optical view is determined and is mapped to a display coordinate (450) using a conformal mapping. Based on one or more detected display coordinates, e.g., corresponding to operations such as a click, a swipe, etc., corresponding action is performed and feedback is provided (460) to the user via the display screen.

Multi finger touch

In some implementations, multi-touch detection is performed. For example, when two fingers are within a field-of-view, a total of four fingers are detected in the captured image: two real, two virtual (reflected). In this situation, a bipartite graph optimization is performed by utilizing a Hungarian matching algorithm to determine an association map between the real and virtual fingers. The Euclidean distance is determined for each pair of fingers at respective locations to determine touch coordinates for each finger.

Gesture detection

Since images within the device field of view are being processed, techniques of this disclosure can also enable gesture recognition for additional user interface (UI) inputs from the user. This can provide capabilities beyond those provided by common input devices such as a mouse, keyboard, trackpad, etc. For example, a pinch can be utilized to trigger a zoom in/out an object, a circular gesture can be utilized to trigger rotation of an object, etc.

Besides touch input using human fingers, other touch input modalities such as nail touch, knuckle touch, human finger touch angle, etc. can be detected and utilized to encode additional user inputs. For example, a tap on the screen with a fingernail can be utilized to represent the right-click of a mouse, a touch on a screen with a knuckle can be utilized to trigger a screenshot, touching the screen with a different finger angle can be used to manipulate 3D views in a map, games, etc.

Image processing to detect touch input is performed locally on the user's computing device. Images that are captured are discarded immediately after detection of the touch input or determination that no touch input was provided. The user can disable the touch detection system via software and/or by removing the accessory. The user can also select specific situations under which touch detection is enabled or disabled, e.g., particular software applications, particular time periods, etc. If necessary for accurate touch detection, skeletal finger touch parameters may be temporarily stored locally for processing.

CONCLUSION

This disclosure describes an optical accessory that enables computing devices without touchscreen hardware to recognize touch inputs using a built-in camera. The accessory includes a fisheye lens attached to a holder, and a tilt plane mirror placed within the accessory that reflects incident light towards the camera of the device to which the accessory is attached. A one-time calibration is performed at a setup time of setup and at runtime, camera images that capture the user's finger position relative to the screen are received. A neural finger pose is estimated by detecting fingers and utilizing a skeletal finger pose detector. A touch event is determined based on an Euclidean distance between the tip joint coordinates of real and virtual fingers. The touch coordinate in the optical view is determined and is mapped to a display coordinate using a conformal mapping.

REFERENCES

1. GLAMOS: Bring Your Touchless Screens To Life, https://www.kickstarter.com/projects/300948436/glamos-bring-your-touchless-screens-to-life, last accessed 26 October 2020.