doi:10.1017/S1431927617001908

Gesture-Based Control of ImageJ Using Leap Sensors

R Brookes[1], BC Breton[2], DM Holburn[2], and NHM Caldwell[3]

[1.] BT plc, Adastral Park, Martlesham Heath, Ipswich, United Kingdom
[2.] Department of Engineering, University of Cambridge, Cambridge, United Kingdom
[3.] Department of Science and Technology, University of Suffolk, Ipswich, United Kingdom

This work forms part of our ongoing research into enhancing and improving microscopy and image processing technologies, and was one of two projects initiated to explore the potential of gesture-based interface control. This project focused on gesture-based manipulation of the ImageJ package, whose functionality can be extended through macros and plugins. In this case, additional plugins were developed to demonstrate the feasibility of interfacing ImageJ with a Leap Motion device.

The Leap Motion is a motion capture device which utilizes three infrared lights and two infrared cameras to detect movement in X, Y and Z axes in three-dimensional space. The sensors are directed in the Y-axis relative to the surface upon which the unit rests. They have a hemispherical view of approximately 150 degrees from the controller and range of 25-600mm. The cameras capture and process images at a frame rate of 20-200 per second, communicating the calculated three dimensional point reference to a base computer for analysis of differences between the two camera perspectives. A three dimensional image is generated for computational processing by the Leap Engine. A pass on individual frame images identifies fingers, hands and gestures; a second pass on a frame sequence identifies movement and direction. The Leap Motion has an Application Programmer’s Interface enabling third-party software to access both the raw frame data as well as positional, motion and gestural information.

Research in other domains have revealed issues in terms of sampling rate inconsistency, inaccuracies in the underlying mathematical model, and occlusion of hands and fingers all leading to incorrect interpretations of user input [1, 2]. The Leap Motion also has a limited repertoire of recognizable movements and gestures (e.g. swipe, circle, tap, pinch and grab). However the pool of gestures can be expanded by careful choice of which hand and which digits are being used – thus it is possible to distinguish between a gesture performed by one hand with index finger extended versus the same gesture performed by the same hand with the thumb extended.

A number of considerations need to be taken into account when designing gestures: 1) hands and fingers must not overlap or gestures will be occluded, leading to probable errors; 2) transitioning between finger gestures is non-disruptive but transitioning between finger gestures to whole hand gestures (e.g. swiping) can impair accuracy (e.g. by losing location on screen during the transition); and 3) humans cannot be perfectly motionless so Leap sensors cannot be operated at full sensitivity. Similarly there are “natural” mappings of gestures: 1) circle (clockwise and anticlockwise) can imply increase/decrease, select/deselect or object rotation; 2) swipe can imply closure, deletion or moving of an object across the screen; 3) key tap implies selection; and 4) screen tap implies selection of what the user was pointing at on the screen.

The project considered two use cases: a simple threshold plugin using Leap Motion input to set the values for an image thresholding operation, and a region-of-interest plugin using Leap Motion input to

https://doi.org/10.1017/S1431927617001908 Published online by Cambridge University Press

designate the boundaries of the region for later processing.

The threshold plugin was essentially a proof-of-concept to demonstrate that Leap and ImageJ could be successfully connected. It was decided to use a finger circle gesture to increase or decrease the threshold (using the distinction between clockwise and anticlockwise to map for increase versus decrease). The intention had been to utilize fractions of a rotation so that a single rotation could adjust the threshold from minimum value to maximum value. Unfortunately the Leap sensor could only register a single gesture from a rotation; thus significant raising or lowering of a threshold from its current value necessitated many circle gestures.

The region of interest plugin was intended to duplicate the selection of image areas by the Rectangle Selection Tool using keyboard and mouse in ImageJ. A two-hand gesture routine was developed where one hand served to move the cursor and the other was used for “clicking” on points of interest – two hands were required to avoid the trap of gesture transitions losing location accuracy. It was necessary to utilize the Leap’s “Stabilize Tip Position” method to ignore unintended minute hand movements, and to code defensively to prevent the controller switching between hands on consecutive image frames. The Polygon Selection Tool proved to be more amenable to Leap interfacing, so this was used instead of the Rectangle Selection Tool. The final plugin had several gestural mappings. If the user was using both hands and pinching, the user was intending to select a region of interest, and palm positions were used to identify the x and y coordinates and the height and width of the rectangular region. Once a region was selected, then a single hand swipe gesture executed the ImageJ Crop tool, a clockwise circle gesture executed the ImageJ “Zoom to Selection”, and an anticlockwise circle executed a “Zoom Out” operation. Due to using palm positions to generate image coordinates, small regions of interest were more difficult to select than larger regions. This was a consequence of the user’s hands having to be positioned too near the Leap controller for accurate readings. A further difficulty was that the method calculated the x and y coordinates directly from the hand positions. Hence once a selected Region of Interest had been set, it could not be fine-tuned or edited as the selected area would resize to the current hand position coordinates.

In conclusion, the project demonstrated the feasibility of interfacing Leap sensors to control ImageJ behaviour. In terms of practicality, mouse and keyboard are more effective user interface mechanisms for using ImageJ and similar complex packages. Industry trends such as the deprecation of “gestures” in the most recent Leap Motion releases and the switch of interest towards virtual reality and voice as human-computer interactions modes means that it is possible that gesture control of software and instrumentation may be limited to touchscreen gestures scaled up from smartphone to desktop for the foreseeable future [3].

References:

[1] A Bracegirdle, “Investigating the Usability of the Leap Motion Controller: Gesture-Based Interaction with a 3D Virtual Environment”, 2014 Dissertation, University of Canterbury, New Zealand.
[2] L Shao, “Hand movement and gesture recognition using Leap Motion Controller”, 2016 Dissertation, Stanford University, USA.
[3] Leap Motion is a trademark of Leap Motion, Inc. This research was supported by funding from Carl Zeiss Microscopy and in-kind support of the first author by BT plc.

https://doi.org/10.1017/S1431927617001908 Published online by Cambridge University Press