stantum
Unlimited Multi-Touch

How to Evaluate Multi-Touch While Standing in a Store

By Jim Meador, Pascal Auriel and Gauthier Chastan

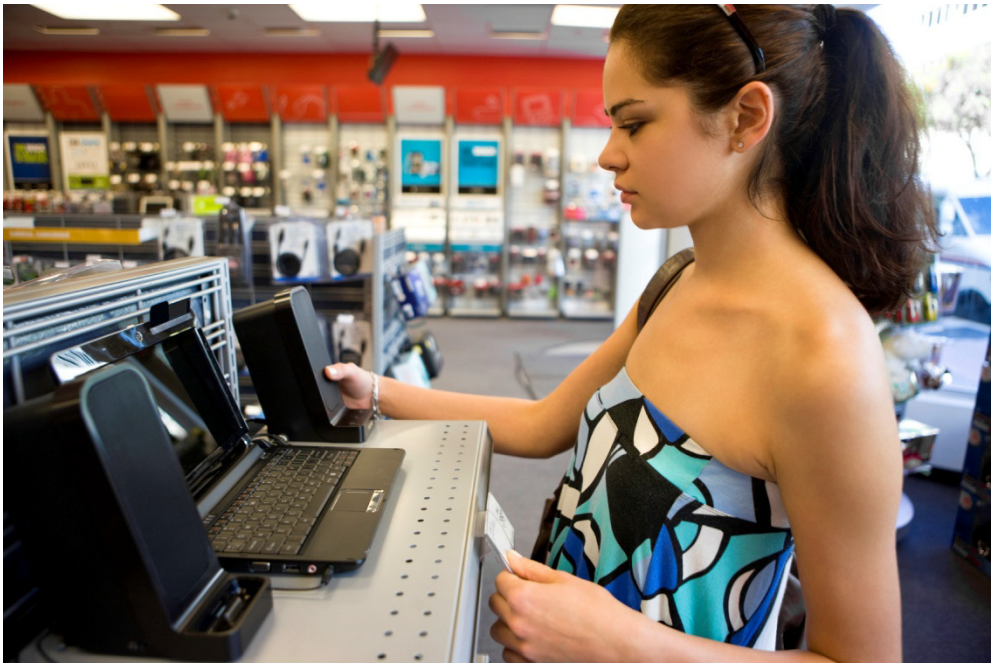

Figure 1: A young woman choosing a new laptop in an electronics store.

I. Introduction

Multi-touch interfaces are becoming increasingly popular for handheld devices and portable electronics. There are a number of technologies that can be used to implement multi-touch functionality. Are they all the same? What are the properties that distinguish a good multi-touch technology from a great one, and how can you test for these properties? This paper describes simple techniques for testing a multi-touch interface, which don't require any special tools or software. We rely on simple test utilities which are usually available for any given system.

Whether you are a professional product reviewer, a technology blogger, or a consumer contemplating a purchase, these tests will enable you to learn more about the strengths and limitations of any given multi-touch system.

Why bother? A good touch screen will be responsive, accurate, and pleasurable to use. A low quality screen may be frustrating, or may even block you from being able to use all the features of your favorite multi-touch enabled software. In today's cost-competitive market, manufacturers will often try to get by with a minimally acceptable solution, so it is up to you

to make sure that the technology you buy will perform to your expectations. Armed with these simple test techniques, you'll be able to gain a lot of information quickly, even while looking at demo units in a store.

Here are the specs that we will test:

Chapt.	*Title*
II a)	Number of simultaneous touches
II b)	Type of contact (finger, stylus, nails...)
II c)	Responsiveness
II d)	Resolution
III a)	Jitter (stability)
III b)	Linearity
III c)	CPU usage
III d)	Pointing accuracy
III e)	Diagonal cursor confusion

Now, let's move on to the first test.

II. Basic Tests

The first tests you'll want to perform are basic tests that reveal the critical characteristics of any device. Later we'll describe some more advanced tests, which may not be important to everyone. In order to do our testing, we'll rely on a drawing application like Microsoft Paint for most of our tests. Systems with Microsoft's Windows 7 OS all include "Paint", but you should be able to find a similar program for any system. We've seen similar programs on Android, Linux, and Apple devices. Of course you'll need to make sure the program is multi-touch capable in order to use it for these tests.

a) How many is "multi"?

One of the first things to know about any multi-touch implementation is how many contact points can be simultaneously detected. Two is the obvious minimum, enough to enable basic multi-touch gestures. Using the drawing application, you can test to see the number of simultaneous points. To test it simply open the application, select a brush color, and try to draw lines with all ten fingers at the same time. The number of lines that actually appear will tell you the number of simultaneous touch points supported by your hardware. Many implementations will stop at two or maybe four. If you get all 10 lines, ask a friend to lend a hand and try to find the limit!

Figure 2: Drawing with 10 fingers on an HP TouchSmart

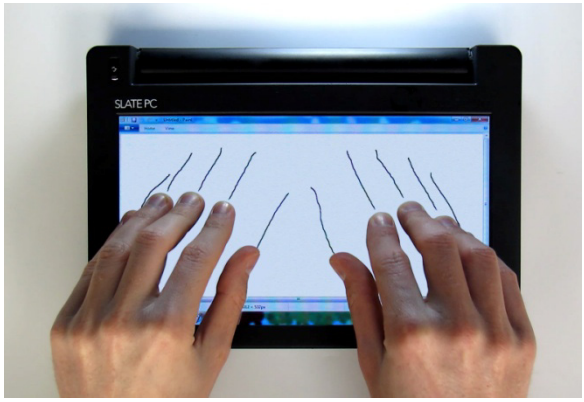

Figure 3: Drawing with 10 fingers on a Stantum Slate Demo

Support for multiple touch points enables more interesting interactions with games, and is also important in applications like handwriting capture, where some of the touch points on the screen, like the palm, may not be intentional, but can block the writing input by using up all the available touch points. Unless you are going to be using only basic gestures, look for an implementation that supports more than two simultaneous contact points.

b) Touch it with this!

All touch interfaces are designed to detect a fingertip, but what about other cases? How about fingernails, or gloved fingers, or a stylus? Using our drawing application again, try drawing with different pointers, keeping track of the results. Try using a glove, your fingernails, a stylus, or a paintbrush. If you live in a cold area and plan to use your device outside, you'll want to try it with your heavy gloves. Fingernails, if supported, can be very handy for very precise pointing tasks without having to pull out a special stylus. This is also important for women with long nails. And speaking of a special stylus, many touch implementations today do require the use of a special "active" stylus, which might be easily misplaced, and can be expensive to replace. Check to see if an ordinary object like the "wrong" end of a pen can work as a stylus. Try a handwriting recognition utility (such as the one that ships with Windows 7) to see if the stylus experience will meet your needs. Testing with a paintbrush can be very revealing both about the types of touch that can be detected, and also about the response to the touch area or pressure. Although it is not commonly supported by drawing applications today, future applications will likely be developed to take advantage of touch pressure to modulate brush size and shape.

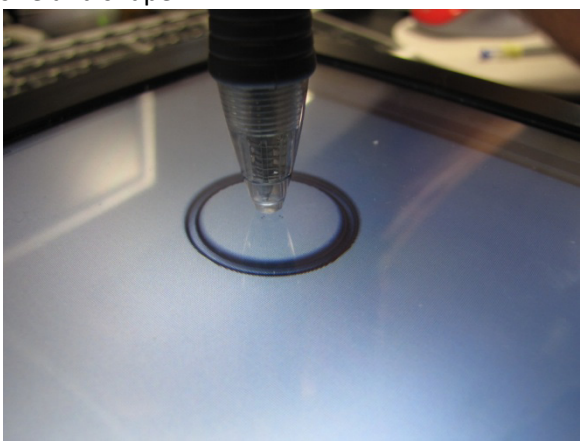

Figure 4: Touch input with a retracted ball point pen

Figure 5: Koreans in this photo are using a sausage product with their iPhones, to avoid taking off their gloves. Credit: Kottke.org

c) Is it responsive enough?

All touch input systems will have some delay between the touch input and the system response. This can be affected by the scan rate of the touch sensor, as well as the overall speed of the system in processing the input and producing an output. A slow response will limit the use for games and other applications that require high performance, and will generally lead to dissatisfaction in using the device. Conversely a fast response increases the usefulness and makes the system more pleasurable to use.

You can measure responsiveness quickly without any equipment. Just draw a straight line across the screen with a drawing application, and note any delay between your finger and the line that appears. Try it again with multiple touch points to see if there is any increase in the delay. Another test is to draw a fast scribble between two horizontal lines. If the touch input

is not being sampled at a high enough rate, the scribble that appears will be lower in amplitude that what you draw (i.e. your scribbles will not reach all the way to the horizontal lines.)

If you want to make a more precise measurement, you can use a video camera. Take a shot of your finger(s) drawing quickly across the screen. You might even place a ruler on the screen. You can then step through the video frame by frame. By knowing the video frame rate, you can see how far your finger moves in one frame, and you can measure the lag of the line that is being drawn.

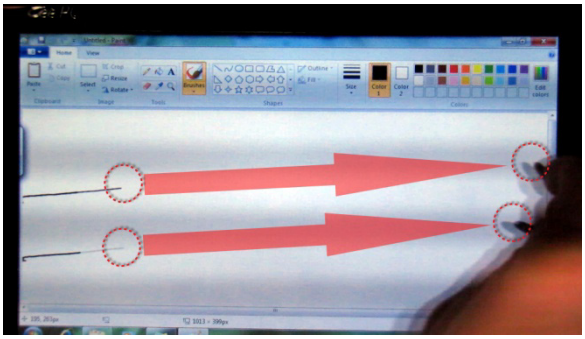

Figure 6: This video capture shows a significant lag when drawing with two contacts on an Asus T91MT

d) How about the resolution?

The ability of a touch interface to locate a precise point on the screen is the resolution, and as with digital cameras and displays, higher resolution is generally better, but only up to a point. Don't expect a touch input system to have the same resolution as a display. Very precise touch input can be obtained by using a "digitizer", but generally touch interfaces do not require such high precision. Most of the graphical objects you will manipulate with touch don't require pixel-level accuracy, and at some point increasing the touch precision does not bring any value. The question is whether a given implementation has enough resolution to support the way you will use it. If you will only be zooming maps or selecting icons, high precision may not be needed. If you need to capture handwriting or use a sensitive touch interface for gaming, precision will be more important.

Figure 7: A smaller circle and closer lines indicate better multi-touch resolution

Usually the best way to evaluate the resolution is with a stylus, which allows the most precise pointing, and helps you to discover the resolutions limits of a given touch system. Using a drawing program, choose a minimal width line, and try to draw the smallest circle you can. You might also try writing a word, using smaller and smaller letters, to see what is the limit where you lose the ability to read it back.

III. Advanced Tests

These tests reveal more subtle performance characteristics of a touch device. They may or may not be critical for every application, but are useful to understand.

a) Hold still for a minute:

The ability of the touch interface to hold a constant, unwavering output for a given input, without jitter or instability, can give an indication of the overall stability and strength of the measuring system. Excessive jitter can be a sign that the touch system is noisy, or picking up a weak signal. To test it, use a drawing program again, selecting a minimal brush. Place one or more contacts down quickly on the touch

screen, and hold them motionless for a few seconds. Then quickly remove them. Any instability in the touch input will be rendered as squiggles at each touch location. A stable touch measurement should give rise to a single tiny dot where each contact was made.

Figure 8: Step 1, keep a finger in contact without moving it

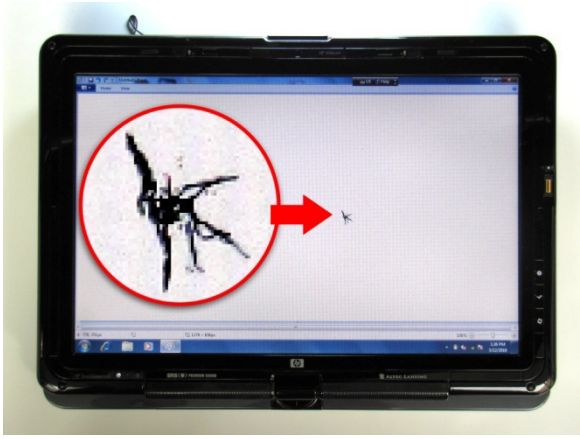

Figure 9: Step 2, note any squiggles that reveal unintended cursor movements. (HP TouchSmart)

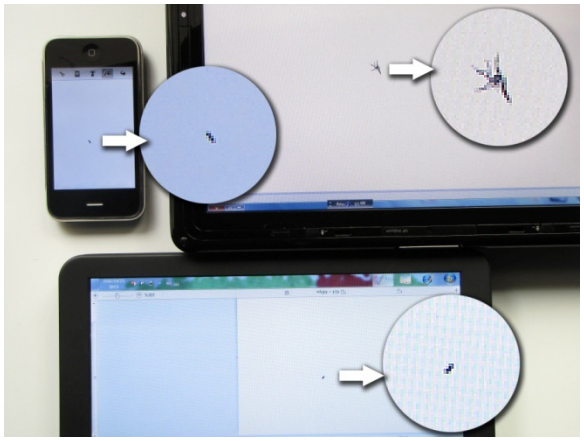

Figure 10: Comparison between Apple's iPhone (left), Stantum's Slate PC (bottom) and HP TouchSmart (top).

b) Line up:

If you move your finger in a straight line on your touch screen, you want a straight line to appear on the display. Some technologies have trouble with this seemingly simple requirement. The test is easy, and it uses the drawing application again. You may want to use a ruler or other straight-edge so that you are sure about your input. Draw a series of straight lines across the full length of the touch surface, from edge to edge. Usually diagonal lines will be more revealing than horizontal or vertical lines, but the test is easy to perform, so you may want to try both. Try moving both slowly and quickly. (Usually the slow lines will be more revealing on any problems).

Figure 11: Drawing a straight line on a Slate PC

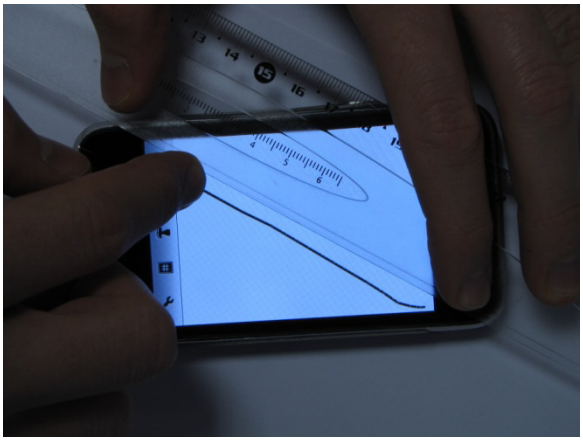

Figure 12: Drawing a straight line on an iPhone

c) Take a load off:

Another factor you may want to consider is the impact that the touch system will have on the overall system performance. Does it bog down the product with a lot of processing requirements, or is it working independently? The test here uses the drawing program again, but also requires an additional indication of the CPU loading. For a PC, there is usually a performance monitor that indicates the CPU loading, such as the Microsoft Windows Performance Monitor. In other environments, look for a similar indicator.

The test is to simply monitor the CPU loading while using the touch interface. You'll probably want to use multiple contact points, but use the same number for both systems in your side-by-side comparison. A good touch system will do its job without excessive loading on the CPU. Some systems rely on extensive post-processing of the raw touch data, which can sap performance and slow down other processes.

It may be hard to draw any conclusions from a single run of this test, since the CPU cycles consumed by the touch system and the drawing application won't be measured separately. The results will be more meaningful if a side-by-side comparison of two different touch systems can be run on the same processor platform. In this case any difference in cycles should be due to the touch input processing, with the cycles needed by the drawing application balancing out on both sides.

d) That hits the spot:

A typical touch screen consists of a transparent sensor which is mounted above a display. Usually touch inputs from the sensor are processed so that objects directly underneath the touch points are redrawn as required. Any misalignment between the sensor and the display will lead to problems, with a touch above one part of the display affecting a nearby graphical object, rather than the desired graphical object. In the case of a soft keyboard, this means typing errors, and other applications might have similar undesirable outcomes. So the test here is simply to touch small buttons on the device and see how often you can hit them.

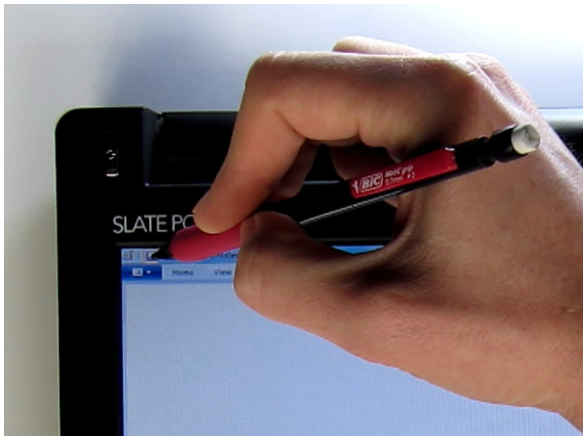

Figure 13: Hitting the save button on Windows 7's paint application with a Slate PC

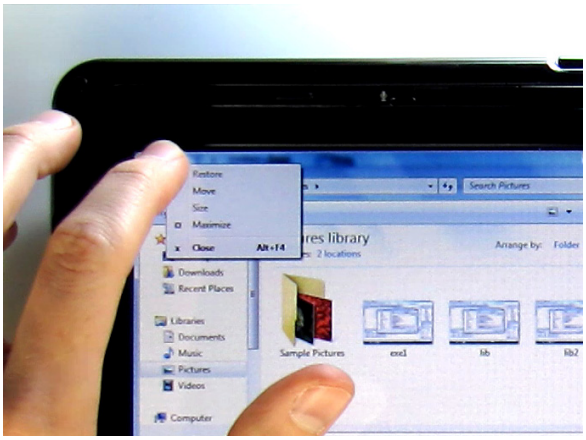

Figure 14: Hitting the save button on Windows 7's paint application with an HP TouchSmart

It is simple to perform this test with the drawing application, just by tapping the touch screen around the whole display area, and making sure that any drawing occurs directly under the touch point. If you want to get more sophisticated, you can cover the touch screen with a thin plastic film, like plastic food wrap, and paint your fingertip with lipstick, or similar material so that you can keep track of exactly where you touched, and compare it to what was drawn.

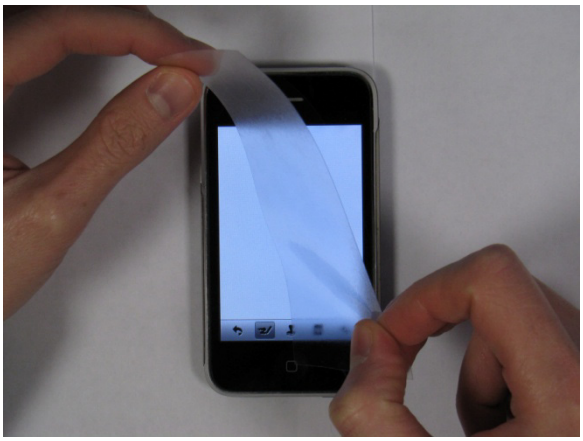

Figure 15: Step 1, adding a transparent overlay

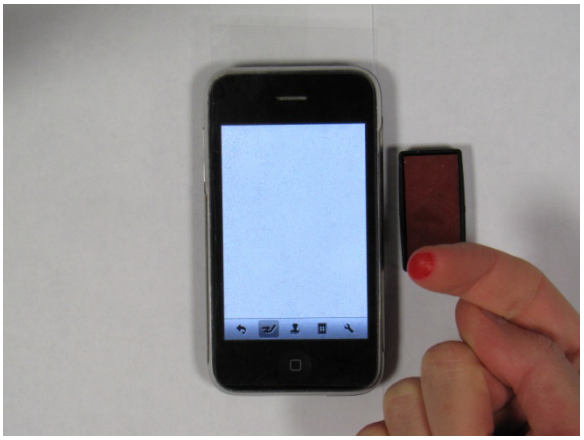

Figure 16: Step 2, inking the finger

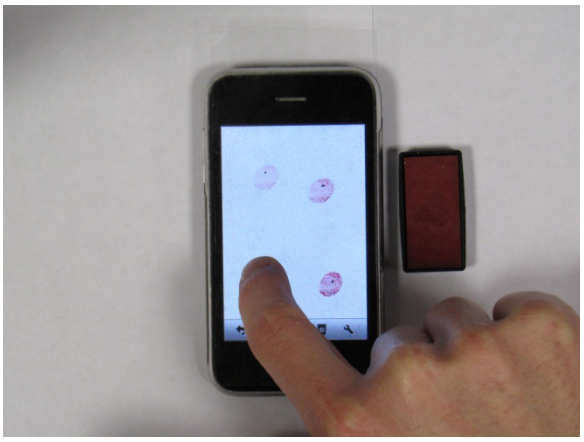

Figure 17: Step 3, the resulting points should be centered with each ink mark

Be sure to work all areas of the screen. Some touch technologies will have particular problems near the edges or in the corners.

e) Diagonal cursor confusion:

Another issue which can affect some touch screens has to do with tracking of two objects as they move close to each other, or pass by each other diagonally. It can arise when two fingers are placed on diagonal corners of the touch screen and then moved horizontally or vertically to opposite corners. In some cases, the fingers' horizontal and/or vertical positions are swapped in the detected output, resulting in ghost

cursors that are symmetric to the locations of the actual contact points.

Use the drawing program again to test for this issue. Place two fingers on diagonal corners of the touch screen, and move them in opposite directions up and down or left and right. Make sure that the drawing follows the two fingers. If the problem occurs you will see ink appearing in the opposite corners from where your fingers really are.

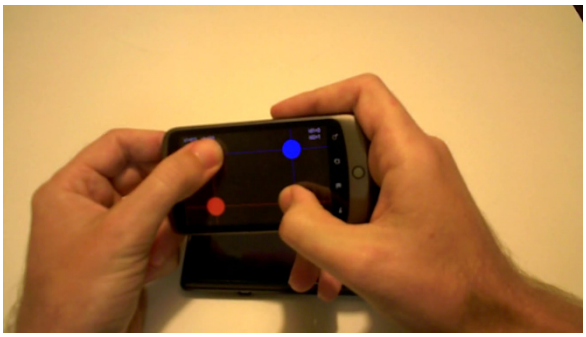

Figure 18: Video snapshot showing symmetric ghost cursors (Nexus One) Credit: androidandme.com

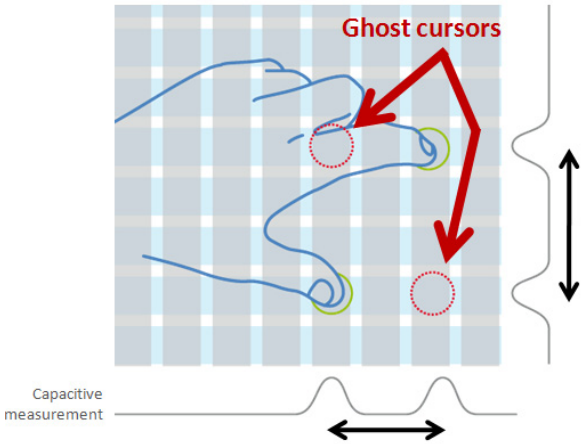

Figure 19: Symmetric dual-touch can only report the distance between two touches, not the real position of each finger.

IV. Conclusions:

We've shown you how to perform a few simple tests that can tell you a lot about the capabilities of any given multi-touch implementation. As these interfaces become more common two things will happen. Software developers will generate new applications that will try to take full advantage of the touch interface. At the same time, hardware vendors will look for ways to deliver touch input performance that is "good enough", but may not be able to unlock the full power of the software. Device reviewers, bloggers, and ordinary users can use the tests we've described here to help distinguish good interfaces from great ones, helping to make the differences clearer, reviews more meaningful, and purchases more satisfying.