Handerkennung mit dem SMARTup

Peter Kämpf

September 2015

Peter.Kaempf@stabilo.com

STABILO International GmbH
Schwanweg 1
90562 Heroldsberg

Zusammenfassung

Um beim Schreiben auf einem Tastfeld die Hand auflegen zu können, wurde ein spezieller Stift entwickelt, der ein gepulstes Muster erzeugt und sich damit von Handkontakten unterscheidet. Dabei war die Abstimmung mit der Elektronik der in iOS-Geräten verwendeten Tastfeldern wichtig. Ein spezieller gestureRecognizer musste geschrieben werden, der die Signale des Stiftes herausfiltert und zu normalen Punktefolgen rekonstruiert. Um eine möglichst genaue und schnelle Erkennung zu gewährleisten, sind einige Besonderheiten zu beachten.

1. Das Konzept

Handschrift hilft beim Lernen neuer Inhalte [1] und verbessert die Kreativität [2]. Jedoch ist die Handschrifteingabe bei der Entwicklung von Tablet-Computern mit Tastfeldern nicht berücksichtigt worden. Mit einem Finger zu schreiben erzeugt unwillkürlich große und unleserliche Buchstaben, jedoch auch mit einem Stift wird die Lesbarkeit nur mit viel Übung besser, denn man ist gezwungen, jeden Kontakt der Schreibhand mit dem Tastfeld zu vermeiden. Daher entstand der Wunsch, die gewohnte Handschrift auf Tastfeldern zu ermöglichen, wobei dies so einfach wie möglich umgesetzt werden sollte. Die im Markt verfügbaren Lösungen, wie Stifte mit Bluetooth [3, 4], wurden diesem Wunsch nicht gerecht: Die geringe Betriebszeit mit einer Batterieladung, der hohe Preis und das mitunter umständliche Paaren standen dem Wunsch nach Einfachheit im Weg. Idealerweise sollte das Muster des Stiftes sich in eindeutiger Weise von dem der Hand unterscheiden. So wurde eine Lösung gewählt, bei der die Körperkapazität kapazitiv an die Elektronik im Inneren des Stiftes angekoppelt ist und abwechselnd auf die Spitze durchgeschaltet und wieder von der Spitze isoliert wird.

2. Anpassung an das iPad-Tastfeld

Zunächst wurden mehrere iPad-Tastfelder vermessen, um die Kapazitätsgrenze zum Registrieren eines Fingerkontakts und deren Streuung zu ermitteln. Der minimale Wert zur Auslösung eines Signals lag dabei knapp über 1 µF und sowohl über der Fläche als auch zwischen einzelnen

Geräten waren die Schwankungen gering. Mit einer so niedrigen Kapazität scheiden die üblichen Halbleiterschalter aus, denn sie haben bereits Eingangskapazitäten ab 3 µF. Die von Griffin [5] verwendete Schaltung mit einem Reed-Relais erzeugt im Betrieb ein störendes Klicken, ist verhältnismäßig groß und hat einen hohen Stromverbrauch, weshalb eine Schaltung mit einer Schaltdiode gewählt wurde. Durch eine spezielle Bauform der Diode konnte die Eingangskapazität der gesamten Schaltung auf 0,7 µF reduziert werden, was einen sicheren Betrieb auf allen iPad-Tastfeldern ermöglicht.

Hier muss angemerkt werden, daß die gängigen Tastfelder für Android-Geräte wesentlich empfindlicher eingestellt sind, weshalb die Anwendung eines gepulsten Stiftes bei ihnen aufgrund der unvermeidlichen Restkapazität der Spitze nicht funktioniert. Während Apple die Unterdrückung von Störsignalen wichtiger zu sein scheint, haben die Hersteller im Android-Bereich sich auf einen Wettbewerb zur Senkung der Auslösegrenze ihrer Tastfelder hinreißen lassen. Das bedeutet aber, dass bei den Tastfeldern von Apple-Geräten eine größere Kontaktfläche erforderlich ist, um ein Signal auszulösen. Dies erfordert auch bei weichen Stiftspitzen mehr Andruck, um die meist sphärisch geformte Spitze genügend plattzudrücken, damit ein Signal ausgelöst wird.

3. Wahl der Betriebsfrequenz

Das Tastfeld aller iPad-Modelle wird im Ruhezustand mit 25 Hz und im Betrieb mit 60 Hz abgetastet. Dabei werden entlang von 39 Streifen aus Indium-Zinn-Oxyd (ITO), einer transparenten, leitfähigen Substanz, Messpulse von 100 µs (Ruhezustand) bzw. 400 µs (Betriebszustand) geschickt. Ein spezieller Schaltkreis, Touchscreen Controller genannt, wertet aus, wie weit diese Messpulse bedämpft werden. Ein quer zu den 39 Streifen liegendes Raster ermöglicht die Auflösung von Berührungen auf Felder von ca. 5x5 mm. Aus der Bedämpfung benachbarter Streifen wird die genaue Position der Berührung interpoliert und vom Betriebssystem an den unter dem berührten Punkt liegenden View und spezielle Software-Module, die GestureRecognizer, gemeldet.

Um den Stift so einfach wie möglich aufzubauen, läuft seine Schaltelektronik mit einer fest eingestellten Frequenz und hat keine Sensorik, um die Messpulse des Tastfeldes zu registrieren. Somit ist die Phasenlage zwischen Schalten und Messen unbestimmt. Um sicher zu sein, dass wenigstens ein Messpuls den Stift nicht wahrnimmt, muss seine Aus-Zeit etwas länger sein als die Periode zwischen zwei Messpulsen. Zudem dauert das Schalten selbst, bis sich die Elektronik stabilisiert hat, knapp 1 ms, so dass in Summe eine Periode der Schaltzustände des Stiftes auf 19 ms festgelegt wurde. Wählt man gleiche An- und Aus-Zeiten, ergibt sich eine Periode von 38 ms und damit eine Frequenz von 26,3 Hz.

Nachteilig ist dabei die maximal mögliche Latenz, bis ein Kontakt des Stiftes mit dem Tastfeld registriert wird. Während diese bei einer normalen Fingereingabe bei der Periode zwischen zwei Messpulsen, also 16,67 ms, liegt, vergrößert sich diese Zeit beim gepulsten Stift um die Dauer des ausgeschalteten Zustandes, also auf 35,67 ms. Daher wurde die Aus-Zeit so kurz wie möglich gewählt. Nach dem Nyquist-Kriterium muss aber die Schaltfrequenz niedriger als die Hälfte der

Abtastfrequenz des Tastfeldes sein, weshalb eine wesentliche Verkürzung der Latenz nicht mehr möglich ist.

Wenn man schon Muster erzeugt, liegt es nahe, verschiedene Muster zuzulassen und sie an eigene Funktionen zu knüpfen. Indem die Aus- und die An-Zeit varriiert werden, lassen sich Muster unterscheiden, die immer noch genügend verschieden zu einem Handkontakt sind, aber weitere Funktionen wie Leuchtmarkieren, Löschen oder einen Wechsel der Stiftfarbe oder Linienbreite signalisieren können.

4. Auslegung des pulsedTouchRecognizers

Im einfachsten Fall sammelt man die Messpulse zwischen einem `touchesBegan` und dem zugehörigen `touchesEnded` und prüft, ob sowohl die maximale Anzahl der Messpulse als auch die Lücke zum vorherigen `touchesEnded` zum Schaltmuster des Stiftes passen. Damit wird aber besonders bei langen Schaltperioden die Latenz zwischen einem Messpuls und der Rückmeldung an den Anwender über das Anzeigen der Linie auf dem Bildschirm vergrößert. Besser ist es daher, bei jedem Messpuls zu prüfen, ob es sich um einen Stiftkontakt handeln kann und nachträglich als Handkontakt identifizierte Punktefolgen wieder zu löschen. Diese Aufgabe übernimmt ein Modul, das wir pulsedTouchRecognizer genannt haben.

Verwendet man mehrere Pulsmuster, muss der `pulsedTouchRecognizer` nicht nur prüfen, ob es sich um eine Stifteingabe handelt, sondern auch das zugehörige Muster erkennen. Für eine Erkennung braucht es mehrere Messpulse, denn aufgrund der gewählten Aus-Zeit kann ein Auszustand des Stiftes von einem oder zwei Messpulsen registriert werden. Wir haben uns für eine Auswertung ab dem Ende des dritten vollständigen Schaltzyklus entschieden, wobei durch die Länge der

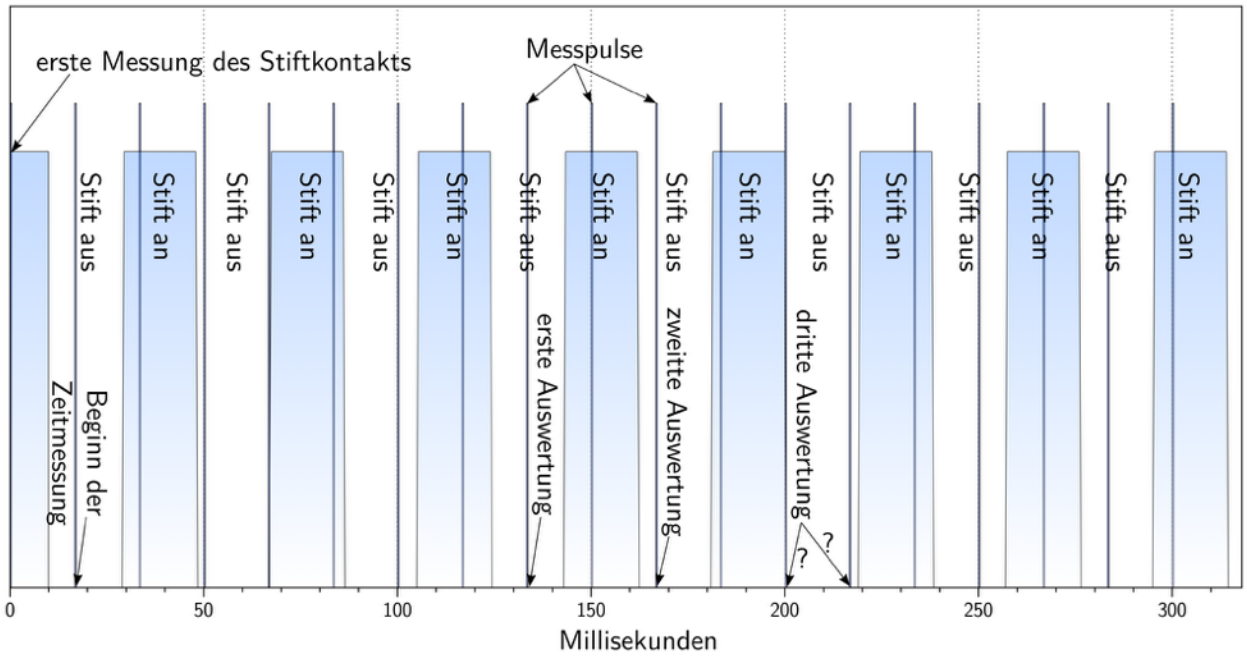


Abbildung 1: Zeitliche Zuordnung von Messpulsen des Tastfeldes und Schaltzuständen des Stiftes beim einfachsten Muster (19 ms an, 19 ms aus).

Aus-Zeit immer noch Ungenauigkeiten möglich sind. Die Abbildung 1 auf der vorhergehenden Seite zeigt dies im Fall der dritten Auswertung des Pulsmusters: Es ist nicht sicher, ob ein `touchesEnded` schon beim ersten Messpuls oder erst beim zweiten innerhalb der Aus-Zeit erzeugt wird.

Eine solche Streuung in der Anzahl der Messpulse pro Zyklus ließe sich erst vermeiden, wenn der Stift die Messpulse des Tastfeldes wahrnehmen kann und sich mit dem Tastfeld synchronisiert. Die dafür erforderlichen zusätzlichen Bauteile an der Stiftespitze heben allerdings deren Eingangskapazität an, wodurch ein Erkennen des Stiftes auch im ausgeschalteten Zustand nicht mehr ausgeschlossen werden kann.

In der praktischen Umsetzung werden einzelne Punktefolgen im `pusedTouchRecognizer` registriert und über einen Vergleich der örtlichen und zeitlichen Nähe zu vorhergehenden Punktefolgen zu Linien verknüpft. Die Rückmeldung an das aufrufende Programm geschieht über den Rückgabewert, ein `NSDictionary`, dessen `key` die Zuordnung neuer Punkte an bestehende Linien erlaubt. Wird ein neuer `key` gemeldet, bedeutet dies den Beginn einer neuen Linie. Das Ende einer Linie kann nur durch das wiederholte Ausbleiben des `key` im Rückgabewert geschlossen werden. Zur Vereinfachung meldet der `touchRecognizer` über eine `NSNotification` ein Ende einer Linie an das aufrufende Programm. Der gleiche Weg wurde gewählt, um ein erkanntes Muster zu kommunizieren.

5. Erfahrung im Betrieb

Besonders bei Berücksichtigung des `majorRadius` einer Berührung auf dem Tastfeld ist der Anteil von falsch eingeschätzten Linien sehr gering, aber besonders bei flüchtigem Handkontakt kommt es immer noch zu kleinräumigen, kurzzeitigen Kontakten, die zuerst angezeigt und dann, etwa 100 bis 200 ms später, wieder gelöscht werden müssen. Ebenso muß eine Anzeige so schnell wie möglich und zuerst im zuletzt erkannten Modus erfolgen, und eine sichere Bestimmung des eventuell neu gewählten Modus erfordert bis zu 250 ms. Daher können neu begonnene Linien nicht gleich in eine Bitmap kopiert werden, sondern müssen zunächst separat in einem transparenten `CALayer` angezeigt werden, bevor sie endgültig übernommen werden können, was den Speicherbedarf bei vielen kurzen Linien dramatisch erhöht. Bei Darstellung mit OpenGL wird es einfacher: Will man eine Linie nachträglich löschen, gibt man einfach ihren Speicherbereich wieder frei.

Bei der Suche nach geeigneten Pulsmustern wurde zunächst sowohl die Aus- als auch die An-Zeit variiert. Damit lassen sich die Perioden kürzer halten, aber die maximale Latenz beim ersten Stiftkontakt steigt bei den Mustern mit längerer Aus-Zeit an. Aber es tritt noch ein weiterer Effekt auf, besonders, wenn die Schaltperiode des Stiftes sich einem Vielfachen der Abtastfrequenz des Tastfeldes nähert: Sinkt die Schwebungsfrequenz zwischen beiden zu sehr ab, kommt es zu langen Folgen, in denen entweder immer nur ein oder immer gleich zwei Messpulse nicht registriert werden, mithin erzeugt das gleiche Schaltmuster je nach Phasenlage mehr oder weniger Aus-Pulse pro Zyklus, was die eindeutige Identifikation eines Musters erschwert.

Stift 130 mit 4 Zyklen

	4 aus	5 aus	6 aus	7 aus	8 aus	9 aus
3 an		1	1	1, 3	3	
4 an	1	1	1, 3	3	3	
5 an	1	1, 3	1, 3	3		
6 an	1, 2	1, 3	3	3		4
7 an	1, 2	2, 3	2	3, 4	4	
8 an	2	2	2, 4	4	4	
9 an	2	2, 4	4	4		
10 an	2, 4	4	4			
11 an	4	4				
12 an	4					

Abbildung 2: Tabelle der Verteilung erkannter und ausgelassener Messpulse von je 200 Eingaben mit vier verschiedenen Schaltmustern nach 4 Schaltzyklen.

In der Tabelle in Abbildung 2 sind auf den Achsen die Anzahl der über 4 Schaltzyklen des Stiftes registrierten (an) und ausgelassenen (aus) Messpulse aufgetragen. Jede Farbe entspricht einem Schaltmuster und Farbverläufe kennzeichnen mehrere Modi für die gleiche Zahl von An- und Aus-Pulsen. Das grüne Schaltmuster hatte eine Periode von 63 ms (entsprechend 15,87 Hz), was bereits zu nah am Vierfachen der Periode des Tastfeldes von 66,67 ms liegt. Die Schwebungsfrequenz zwischen beiden beträgt 0,87 Hz, so dass es über eine Sekunde dauert, bis die Phasenlage zwischen beiden einen vollen Durchlauf hinter sich gebracht hat. Daher gibt es Punktefolgen mit nur wenig nicht registrierten Messpulsen, gefolgt von langen Folgen mit vielen nicht registrierten Messpulsen. Eine schnelle Unterscheidung zwischen zwei gleich langen Schaltmustern nur über die Auszeit wird damit praktisch unmöglich.

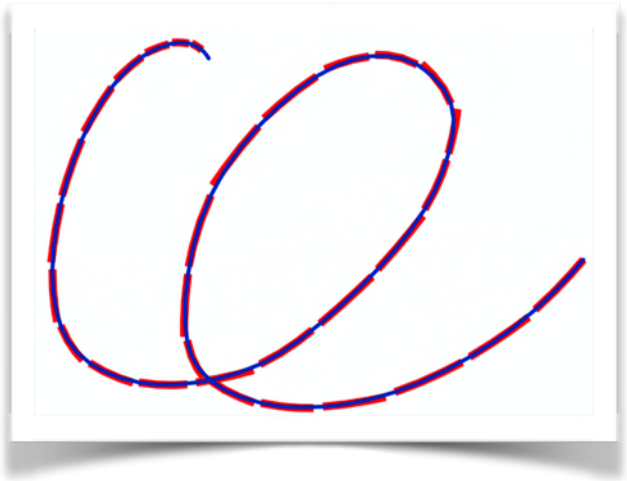
Daher wurden für das Produkt nur Muster gewählt, die eine kurze Aus-Zeit haben und die Unterscheidung erfolgt lediglich über die Dauer der Schaltperiode, praktisch also über die Anzahl der Messpulse pro Schaltzyklus. Zudem wurde die kürzeste An-Zeit auf 22 ms angehoben, wodurch die Schaltperiode nun 41 ms beträgt, was bequem zwischen dem Zwei- und Dreifachen der Messperiode des Tastfelds liegt.

Ein Handkontakt erzeugt bereits typisch drei Berührpunkte, und mit dem Stiftkontakt kann die maximal vom iPad unterstützte Anzahl von Punkten überschritten werden, besonders, wenn Hand- und Stiftkontakt auf der gleichen Elektrodenbahn des Tastfeldes liegen. Im Betrieb äußert sich dieses als ein Verschwinden des Stiftkontaktes bei aufgelegter Hand, wenn man diesen vertikal über die Handposition bewegt. Dieses Verhalten kann auch bei gewöhnlichen Griffeln beobachtet werden und zeigt die Grenzen der projiziert-kapazitiven Berührungsmessung des iPad-Tastfeldes bei Betrieb mit aufliegender Hand auf. Eine solche Stifthaltung ist allerdings sehr ungewöhnlich, da sie ohne eine starke Dehnung des Handgelenkes kaum zu erzeugen ist.

6. Einbindung der iOS-internen gestureRecognizer

Wenn man Klassen von Berührpunkten anhand ihrer seit iOS8 mitgemeldeten Eigenschaft `major–Radius` bildet, kann man Stift-, Finger- und Handkontakte unterscheiden. Bei Erkennen eines zu großen Kontakts meldet der `pulsedTouchRecognizer` den `state` Failed, und die Messwerte können an weitere `gestureRecognizer` gemeldet oder in einer eigenen Klasse von Linien angezeigt werden. Dabei ist es aber wie zuvor bei den verschiedenen Schaltmustern: Will man mehr Varianten unterscheiden, nehmen Falsch-Klassifikationen zu. Es gilt also, einen Kompromiß zwischen der Zuverlässigkeit und der Funktionsvielfalt zu finden.

7. Extrapolation der Linie



Durch die interne Verarbeitung nur jedes zweiten Messpulses ist ein gewisser Versatz zwischen der Position der Stiftspitze und dem Strich auf dem Bildschirm unvermeidlich. Indem der nächste Berührpunkt vorausgesagt wird, kann man die Linie zur Spitze hin verlängern. Parallel kann dieser extrapolierte Punkt zum Testen des nächsten gemessenen Punktes auf Zugehörigkeit zur Linie verwendt werden; in der Realität sind damit geringere Abstände möglich, als wenn man den Abstand zum letzten gemessenen Punkt der Linie für die Zugehörigkeitsprüfung heranzieht. Die Grafik rechts zeigt die Extrapolationen als breitere, rote Striche; die blaue Linie ist die gesplinte Linie aus den tatsächlich gemessenen Punkten. Indem man die Extrapolation verwendet, um den Spline bis zum zuletzt gemessenen Punkt zu verlängern, rückt die Linie beim Zeichnen nah an die Spitze heran, ohne durch das Überschreiben des letzten Abschnitts mit jeder Fortsetzung der Linie zu unruhig zu werden.

Literatur

[1] Marr, Windsor, Cermak: Handwriting Readiness: Locatives and Visuomotor Skills in the Kindergarten Year; ECRP, Vol. 3 No. 1. (http://ecrp.uiuc.edu/v3n1/marr.html)

[2] Berninger, Abbott, Augsburger, Garcia: Comparison of pen and keyboard transcription modes in children with and without learning disabilities affecting transcription (2009). Learning Disability Quarterly, 32, 123-141

[3] Adonit Jot Pro (http://www.adonit.net/jot/pro/)

[4] Wacom Intuos Creative Stylus (http://www.gravis.de/Zubehoer/Eingabegeraete/Eingabestifte/Wacom-Intuos-Creative-Stylus-fuer-iPad-3-4-iPad-mini-Bluetooth-4-0-schwarz.html)

[5] Griffin iMarker (https://store.griffintechnology.com/crayola-imarker)