



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 695 34 404 T2** 2006.03.16

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 0 870 223 B1**

(51) Int Cl.⁸: **G06F 3/033** (2006.01)

(21) Deutsches Aktenzeichen: **695 34 404.8**

(86) PCT-Aktenzeichen: **PCT/US95/13306**

(96) Europäisches Aktenzeichen: **95 938 766.3**

(87) PCT-Veröffentlichungs-Nr.: **WO 96/011435**

(86) PCT-Anmeldetag: **06.10.1995**

(87) Veröffentlichungstag
der PCT-Anmeldung: **18.04.1996**

(97) Erstveröffentlichung durch das EPA: **14.10.1998**

(97) Veröffentlichungstag
der Patenterteilung beim EPA: **24.08.2005**

(47) Veröffentlichungstag im Patentblatt: **16.03.2006**

(30) Unionspriorität:
320158 07.10.1994 US

(84) Benannte Vertragsstaaten:
DE, FR, GB

(73) Patentinhaber:
Synaptics, Inc., San Jose, Calif., US

(72) Erfinder:
GILLESPIE, David, Palo Alto, US; ALLEN, P., Timothy, Los Gatos, US; WOLF, Ralph, Santa Clara, US

(74) Vertreter:
**Rechts- und Patentanwälte Lorenz Seidler Gossel,
80538 München**

(54) Bezeichnung: **Verfahren zum kompensieren von unbeabsichtigten Bewegungen bei einer Antipgeste auf einer Berührungssensorfläche**

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

Beschreibung

[0001] Die vorliegende Erfindung betrifft Objektposition erfassende Wandler und Systeme. Insbesondere betrifft die vorliegende Erfindung eine Objektpositionserkennung, die bei Anwendungen wie Bewegung des so genannten Cursors (dt. Einfügemarke) bei Rechnern und anderen Anwendungen brauchbar ist, und speziell die Bewegung des Cursors, bei verbesserten Merkmalen der Randweiterbewegung und Gestenerkennung.

[0002] Es gibt zahlreiche Geräte zur Verwendung als Objektpositionsdetektoren für den Einsatz bei Computersystemen und anderen Anwendungen bzw. wurden vorgeschlagen. Das bekannteste dieser Geräte ist die Computer-„Maus“. Sie ist zwar als positionsanzeigende Vorrichtung äußerst beliebt, doch weist eine Maus mechanische Teile auf und benötigt eine Fläche, auf der ihr Positionsball rollen kann. Weiterhin muss eine Maus für vernünftige Auflösung meist lange Strecken bewegt werden. Schließlich fordert eine Maus vom Anwender, dass er eine Hand von der Tastatur nimmt, um die Cursor-Bewegung auszuführen, wodurch der Hauptzweck beeinträchtigt wird, nämlich das Tippen auf dem Computer.

[0003] So genannte Trackball-Geräte (dt. Steuerkugelgeräte) ähneln Mausgeräten. Ein Hauptunterschied ist aber, dass im Gegensatz zu einem Mausgerät ein Trackball-Gerät keine Fläche erfordert, über die es gerollt werden muss. Trackball-Geräte sind immer noch teuer, haben bewegliche Teile und erfordern wie Mausgeräte ein relativ starkes Berühren. Sie sind ferner groß und passen nicht gut in eine volumenempfindliche Anwendung wie einen Laptop-Computer.

[0004] Es gibt mehrere Berührung erfassende Technologien, die zur Verwendung als Positionsindikator eingesetzt werden können. Positionssensoren mit Widerstandsmembran sind bekannt und werden bei mehreren Anwendungen eingesetzt. Im Allgemeinen leiden sie aber unter schlechter Auflösung, die Sensorfläche ist gegenüber dem Bediener ungeschützt und unterliegt dadurch Verschleiß. Ferner sind Berührungssensoren mit Widerstandsmembran relativ teuer. Eine Strategie mit einer Oberfläche erfordert, dass der Bediener für zuverlässigen Betrieb gegenüber dem Sensor geerdet ist. Das kann bei tragbaren Rechnern nicht garantiert werden. Ein Beispiel für eine Strategie mit einer Oberfläche ist das Produkt UnMouse (™) von MicroTouch, Wilmington, MA. Eine Strategie mit zwei Oberflächen hat eine schlechtere Auflösung und verschleißt unter Umständen recht schnell.

[0005] Widerstandstabletten werden von U.S. Pat. Nr. 4,680,430 für Yoshikawa, U.S. Pat. Nr. 3,497,617 für Ellis und von vielen anderen gelehrt. Der Nachteil all dieser Strategien ist der hohe Stromverbrauch und die hohen Kosten der eingesetzten Widerstandsmembran.

[0006] Akustische Oberflächenwellengeräte (SAW-Geräte) sind potenziell als Positionsindikatoren verwendbar. Diese Sensortechnologie ist aber teuer und spricht auf leichte Berührung nicht an. Ferner sind SAW-Geräte gegenüber Ablagerungen auf den Berührungsflächen empfindlich und haben im Allgemeinen eine schlechte Auflösung.

[0007] Dehnmessstreifen- oder Druckplattenstrategien sind eine interessante Positionserfassungstechnologie, leider aber unter mehreren Nachteilen. Dieses Vorgehen kann piezoelektrische Wandler einsetzen. Ein Nachteil ist, dass das Piezo-Phänomen ein Wechselstromphänomen ist und auf die Bewegungsgeschwindigkeit des Bedieners ansprechen kann. Ferner sind Dehnmessstreifen- oder Druckplattenstrategien relativ teuer, da spezielle Sensoren erforderlich sind.

[0008] Optische Strategien sind ebenfalls möglich, aber aus mehrerlei Gründen relativ eingeschränkt. Alle würden Lichterzeugung erfordern, was externe Komponenten nötig macht und Kosten und Stromverbrauch erhöht. Zum Beispiel verbraucht ein „fingerbrechender“ Infrarot-Matrixpositionsdetektor viel Strom und leidet unter einer relativ schlechten Auflösung.

[0009] Es gibt zahlreiche Versuche, ein Gerät für das Erfassen der Position eines Daumens oder anderen Fingers zur Verwendung als Zeigegerät als Ersatz für eine Maus oder einen Trackball an die Hand zu geben. Wünschenswerte Eigenschaften eines solchen Geräts sind geringer Stromverbrauch, geringe Dicke, hohe Auflösung, niedrige Kosten, schnelle Reaktion und die Fähigkeit, zuverlässig zu arbeiten, wenn der Finger elektrisches Rauschen überträgt oder wenn die Berührungsfläche mit Schmutz oder Feuchtigkeit verunreinigt ist.

[0010] Aufgrund der Nachteile von Widerstandsgeräten wurden viele Versuche unternommen, Zeigertauglichkeit beruhend auf einem kapazitiven Erfassen der Position des Fingers zu bieten. U.S. Pat. Nr. 3,921,166 für Volpe lehrt eine kapazitive Matrix, bei der der Finger die Transkapazität zwischen Zeilen- und Spaltenelek-

troden ändert. U.S. Pat. Nr. 4,103,252 für Bobick setzt vier oszillierende Signale ein, um x- und y-Positionen zwischen vier kapazitiven Elektroden zu interpolieren. U.S. Pat. Nr. 4,455,452 für Schuyler lehrt ein kapazitives Tablett, bei dem der Finger die kapazitive Kopplung zwischen Elektroden dämpft.

[0011] Das U.S. Pat. Nr. 4,550,221 für Mabuth lehrt ein kapazitives Tablett, bei dem die Betriebskapazität gegenüber "virtueller Erde" durch ein oszillierendes Signal gemessen wird. Jede Zeile oder Spalte wird sequentiell abgefragt, und es wird eine rudimentäre Form der Interpolation angewendet, um die Position zwischen zwei Zeilen oder Spalten zu ermitteln. Durch Mitteln über viele Zyklen der oszillierenden Wellenform wird ein Versuch unternommen, das Problem der elektrischen Interferenz anzugehen. Das Problem der Verunreinigung wird dadurch angegangen, dass erfasst wird, wann kein Finger anliegt, und eine periodische Kalibrierung während solcher Zeiten ohne anliegenden Finger durchgeführt wird. U.S. Pat. Nr. 4,639,720 für Rymalski lehrt ein Tablett für das Erfassen der Position eines Eingabestifts. Der Eingabestift ändert die Transkapazitätskopplung zwischen Zeilen- und Spaltenelektroden, die sequentiell abgetastet werden. U.S. Pat. Nr. 4,736,191 für Matzke lehrt eine radiale Elektrodenanordnung unter der Leertaste einer Tastatur, die durch Berührung mit einem Daumen zu aktivieren ist. Dieses Patent lehrt die Verwendung einer Gesamtberührungskapazität als Anzeige des Berührungsdrucks zur Steuerung der Geschwindigkeit der Cursor-Bewegung. Um die Wirkungen elektrischer Interferenz anzugehen, wird gepulstes sequentielles Abfragen eingesetzt.

[0012] U.S. Pat. Nr. 4,686,332 und 5,149,919 für Greanias lehrt einen Eingabestift und ein Fingerdetektionssystem, das zur Anbringung an einer Kathodenstrahlröhre gedacht ist. Als Fingerdetektionssystem wird seine X/Y-Sensormatrix zur Lagefeststellung der beiden Matrixdrähte, die das maximale Signal führen, eingesetzt. Mit einem Codierungsschema ermitteln diese beiden Drähte eindeutig die Lage der Fingerposition bis zur Auflösung der Leiterabstufung. Bei der Eingabestiftdetektion ermittelt Greanias ihn zuerst grob, entwickelt dann einen virtuellen Dipol durch Antreiben aller Leitungen an einer Seite des Objekts in einer Richtung und aller Leitungen an der gegenüberliegenden Seite in die Gegenrichtung. Dies wird dreimal mit verschiedenen Dipolphasen und Signalpolaritäten ausgeführt. Unter der Annahme einer vorgegebenen Matrixreaktion auf das Objekt geben die drei Messungen einen Satz gleichzeitiger Gleichungen, die für die Position gelöst werden können.

[0013] U.S. Pat. Nr. 4,733,222 für Evans lehrt zum ersten Mal ein Kapazitätsberührungsmesssystem, das hochgradig interpoliert. Evans lehrt ein Dreiklemmen-Messsystem, das einen Ansteuer-, Erfassungs- und Elektroden-signalsatz (3 Signale) in seiner Matrix verwendet und die Messung auf die Dämpfungswirkung eines Fingers auf dem Elektrodenknotensignal stützt (nutzt ein kapazitives Teilerphänomen). Evans tastet sequentiell jeden Ansteuersatz ab, um die Kapazität zu messen. Aus den drei größten Reaktionen wird eine Interpolationsroutine zur Ermittlung der Fingerposition angewendet. Evans lehrt auch ein Nullsetzungsverfahren, das als Teil der Messung ein Löschen von Werten „ohne Finger“ zulässt.

[0014] U.S. Pat. Nr. 5,016,008 für Gruaz beschreibt eine berührungsempfindliche Fläche, die ebenfalls Interpolation verwendet. Gruaz benützt einen Ansteuer- und Erfassungssignalsatz (2 Signale) in der Berührungsmatrix und stützt sich wie Evans auf die Dämpfungswirkung eines Fingers zur Modulierung des Ansteuersignals. Die Berührungsmatrix wird sequentiell abgetastet, um die Reaktion jeder Matrixzeile zu lesen. Ein Interpolationsprogramm wählt dann die beiden größten benachbarten Signale in beiden Dimensionen aus, um die Fingerposition zu ermitteln, und ermittelt auf ratiometrische Weise die effektive Position aus diesen 4 Zahlen.

[0015] Gerpheide, PCT-Anmeldung US90/04584, Veröffentlichung Nr. WO91/03039, U.S. Pat. Nr. 5,305,017 wendet bei einem Berührungsflächensystem eine Abwandlung des virtuellen Dipol-Vorgehens von Greanias an. Gerpheide lehrt die Anwendung eines oszillierenden Potentials vorgegebener Frequenz und Phase bei allen Elektroden an einer Seite des virtuellen Dipols und eines oszillierenden Potentials der gleichen Frequenz und entgegengesetzten Phase an denen an der anderen Seite. Elektronische Schaltungen entwickeln ein „Ausgleichssignal“, das Null ist, wenn kein Finger vorhanden ist, und eine Polarität aufweist, wenn ein Finger an einer Seite der Mitte des virtuellen Dipols vorhanden ist, und die entgegengesetzte Polarität, wenn sich der Finger an der gegenüberliegenden Seite befindet. Um zunächst die Position des Fingers zu erhalten, wird der virtuelle Dipol sequentiell über das Tablett abgetastet. Sobald die Lage des Fingers festgestellt ist, wird er durch Bewegungen des virtuellen Dipols hin zum Finger „verfolgt“, sobald sich der Finger mehr als eine Zeile oder Spalte bewegt.

[0016] Da das virtuelle Dipol-Verfahren durch Erzeugen eines Ausgleichssignals arbeitet, das Null ist, wenn sich die Kapazität nicht mit dem Abstand ändert, erfasst es nur den Umfang der Fingerkontaktfläche statt der gesamten Kontaktfläche. Da sich das Verfahren auf eine synchrone Detektion des Erregungssignals stützt, muss es zur Störunterdrückung über lange Zeiträume mitteln und ist daher langsam. Die bei diesem Verfahren

erforderliche Mittelungszeit zusammen mit der Notwendigkeit, sequentiell nach einem neuen Fingerkontakt zu suchen, sobald ein vorheriger Kontakt verloren geht, lässt dieses Verfahren wie die vorherigen auch hinter den Anforderungen an eine schnelle Zeigervorrichtung, die nicht durch elektrische Interferenz beeinträchtigt wird, zurückbleiben.

[0017] Es versteht sich auch, dass alle vorherigen Berührungsflächen-Erfindungen, die Interpolation verwendeten, strenge Konstruktionsanforderungen an ihre Erfassungsfläche stellten. Greanias und Evans verwenden ein kompliziertes und teures Ansteuer-, Erfassungs- und Elektrodenleitungsschema zur Entwicklung ihres Signals. Gruaz und Gerpheide verwenden einen Ansteuer- und Erfassungssatz mit zwei Signalen. Bei der vorliegenden Erfindung erfolgt das Ansteuern und Erfassen an der gleichen Leitung. Dies lässt symmetrische und äquivalente Zeilen- und Spaltenabschnitte zu. Dies lässt wiederum eine unabhängige Kalibrierung aller Signalwege zu, was das Leiterplatten-Layout einfacher und weniger einschränkend macht und einzigartigere Sensortopologien zulässt.

[0018] Die Nachteile der in der vorbekannten Technik beschriebenen Erfindungen und Verfahren können auch auf die Verwendung von nur einem Ansteuer- und Erfassungselektroniksatze, der sequentiell über die Elektroden in dem Tablett multiplex betrieben wird, zurückverfolgt werden. Diese Anordnung war in den Tagen diskreter Komponenten kostengünstig und vermied Offset und Maßstabunterschiede zwischen den Schaltungen.

[0019] Das sequentielle Abtastvorgehen früherer Systeme machte diese auch gegenüber Rauschen empfindlicher. Rauschpegel konnten sich zwischen aufeinander folgenden Messungen ändern, wodurch das gemessene Signal und die in den Interpolationsroutinen verwendeten Annahmen geändert wurden.

[0020] Schließlich nahmen alle früheren Vorgehen eine bestimmte Signalreaktion für die Fingerposition gegenüber einer Matrixposition an. Da die Übertragungskurve gegenüber vielen Parametern sehr empfindlich ist und keine gleichmäßige lineare Kurve ist, wie Greanias und Gerpheide annehmen, sind solche Vorgehen bezüglich des Interpolationsmaßes beschränkt, das sie leisten können.

[0021] Von den so genannten Touchpad-Geräten (Geräten mit berührungsempfindlichem Feld), die derzeit erhältlich sind, enthält nur der Alps/Cirque GlidePoint (™) eine Gestenerkennung. Der GlidePoint unterstützt einfache Tipp-, Doppeltipp- und Ziehgesten zur Simulierung der Bedienung einer primären Maustaste. Er unterstützt nicht mehrfingrige Gesten, es gibt auch keine Gesten für das Simulieren von sekundären Tastenklicks. Zu den im GlidePoint eingesetzten Umsetzungsverfahren sind keine Informationen bekannt. Der GlidePoint hat aber bekanntermaßen Schwierigkeiten mit Doppeltippen, einem Problem, das von der vorliegenden Erfindung angegangen wird. Der GlidePoint weist bei jedem Fingerbewegungsstreichen eine Verzögerung auf, was ein Versuch sein kann, den Cursor während Tippgesten zu stabilisieren. Ferner ist der GlidePoint auf physikalische Schalter oder extrem hohe Verstärkung oder Beschleunigung angewiesen, um Ziehen über lange Strecken zuzulassen.

[0022] Bei einem Erzeugnis mit berührungsempfindlicher Fläche, der UnMouse, ist ein Schalter unter ihrem Widerstandssensor angebracht, so dass der Bediener einfach auf die Fläche drückt, um die Taste zu aktivieren. Abgesehen von der Erfordernis einer zerbrechlichen und komplexen mechanischen Befestigung ist dieses Gerät angeblich auch recht ermüdend für den Bediener.

[0023] Graphiktablets, die durch einen druckempfindlichen Eingabestift an Stelle eines Fingers bedient werden, sind auf dem Gebiet gut bekannt. Diese Geräte verwenden üblicherweise einen Mechanismus wie die „Schiebe“-Geste der vorliegenden Erfindung, um Bedienschalter zu simulieren. Keine anderen Gesten der hier beschriebenen Art wurden bisher bei mit Eingabestift bedienten Tablets gesehen.

[0024] EP-A-0 490 001 offenbart ein Gerät mit berührungsempfindlicher Fläche, bei dem eine Tippgeste erkannt wird, die dem Host-Computer übermittelten X- und Y-Koordinaten eingefroren werden.

[0025] Eine Aufgabe der Erfindung besteht darin, die Erkennung von Gesten eines Fingers oder anderen Objekts eines Berührungssensorfelds so vorzusehen, dass eine unbeabsichtigte Bewegung des Fingers oder anderen Objekts während der Ausführung der Geste kompensiert wird.

[0026] Erfindungsgemäß wird ein Verfahren für das Kompensieren einer unbeabsichtigten seitlichen Bewegung bei Antippen eines Berührungssensorfelds in einem Berührung erfassenden System, welches einem Host X- und Y-Positionsangaben liefert, an die Hand gegeben, wobei es die folgenden Schritte umfasst:

Detektieren des Eintretens eines von einem leitenden antippenden Objekt vorgenommenen Antippens des Berührungssensorfelds; und
 bei Detektieren des Eintretens eines Antippens Senden eines Signals an den Host, welches das Eintreten des Antippens anzeigt; wobei das Verfahren durch folgendes gekennzeichnet ist:
 Senden von X- und Y-Positionsangaben an den Host, welche eine seitliche Bewegung des antippenden Objekts an dem Berührungssensorfeld während des Antippens darstellen; und
 bei Detektieren des Eintretens eines Antippens Senden von weiteren X- und Y-Positionsangaben an den Host, welche eine Bewegung entgegengesetzt zur seitlichen Bewegung des antippenden Objekts an dem Berührungssensorfeld während des Antippens darstellen.

[0027] Daher wird für eine unabsichtliche Bewegung des Fingers oder anderen Objekts während der Ausführung der Antippgeste Ausgleich geboten.

[0028] Die Erfindung wird nachstehend unter Bezug auf die folgende Beschreibung und die Begleitzeichnungen weiter beschrieben. Hierbei zeigen:

[0029] [Fig. 1](#) ein Gesamtschemata des kapazitiven Positionserfassungssystems.

[0030] [Fig. 2a](#) eine Draufsicht auf einen Objektpositionssensorwandler, wobei die Oberflächenschicht des Objektpositionssensors mit einer oberen leitenden Leiterbahnschicht und leitenden Feldern, die mit einer unteren Leiterbahnschicht verbunden sind, gezeigt wird.

[0031] [Fig. 2b](#) eine Bodenansicht des Objektpositionssensorwandlers von [Fig. 2a](#), welche die untere leitende Leiterbahnschicht zeigt.

[0032] [Fig. 2c](#) eine Zusammenschau des Objektpositionssensorwandlers von [Fig. 2a](#) und [Fig. 2b](#), die sowohl die obere als auch die untere leitende Leiterbahnschicht zeigt.

[0033] [Fig. 2d](#) eine Querschnittansicht des Objektpositionssensorwandlers der [Fig. 2a-Fig. 2c](#).

[0034] [Fig. 3](#) ein Blockdiagramm einer Sensorentschlüsselungselektronik, die mit dem Sensorwandler verwendet werden kann.

[0035] [Fig. 4a](#) ein vereinfachtes schematisches Diagramm einer Ladungsintegratorschaltung, die in der vorliegenden Erfindung verwendet werden kann.

[0036] [Fig. 4b](#) ein veranschaulichendes schematisches Diagramm der Ladungsintegratorschaltung von [Fig. 4a](#).

[0037] [Fig. 5](#) ein Zeitverlaufdiagramm des Betriebs der Ladungsintegratorschaltung von [Fig. 4a](#) und [Fig. 4b](#).

[0038] [Fig. 6](#) ein schematisches Diagramm einer veranschaulichenden Filter- und Abtast-Halteschaltung zur Verwendung in der vorliegenden Erfindung.

[0039] [Fig. 7](#) ein detaillierteres Blockdiagramm einer derzeit bevorzugten Anordnung von Analog-Digital-Wandlern zur Verwendung in der vorliegenden Erfindung.

[0040] [Fig. 8](#) ein Blockdiagramm einer veranschaulichenden Recheneinheit, die in der vorliegenden Erfindung verwendet werden kann.

[0041] [Fig. 9](#) ist ein Blockdiagramm einer Kalibrierungseinheit, die mit der Recheneinheit von [Fig. 8](#) verwendet werden kann.

[0042] [Fig. 10](#) ein schematisches Diagramm einer Vorspannung erzeugenden Schaltung, die in der vorliegenden Erfindung brauchbar ist.

[0043] [Fig. 11](#) ein Diagramm der Erfassungsebene, welches das Randweiterbewegungsmerkmal eines Objektpositionssensors veranschaulicht.

- [0044] [Fig. 12](#) ein schematisches Diagramm, das die Hardware-Implementierung der Ermittlung veranschaulicht, ob ein Finger oder anderes Objekt in den Randbereichen der Erfassungsebene vorhanden ist.
- [0045] [Fig. 13](#) ein schematisches Diagramm, das die Hardware-Implementierung des Randweiterbewegungsmerkmals veranschaulicht.
- [0046] [Fig. 14](#) ein detaillierteres Blockdiagramm der Gesteneinheit **20** von [Fig. 1](#).
- [0047] [Fig. 15a](#) bis [Fig. 15e](#) Zeitverlaufdiagramme, die einige der Gesten veranschaulichen, die erkannt werden können.
- [0048] [Fig. 16a](#) und [Fig. 16b](#) Diagramme, die zwei Antippzonenformen veranschaulichen, die auf Sensorfeldern in der vorliegenden Erfindung verwendet werden können.
- [0049] [Fig. 17a](#) bis [Fig. 17f](#) ein Flussdiagramm, das den Betrieb der Antippeinheit von [Fig. 14](#) zeigt.
- [0050] [Fig. 18a](#) bis [Fig. 18c](#) ein Flussdiagramm, das den Betrieb der Zickzackeinheit von [Fig. 14](#) veranschaulicht.
- [0051] [Fig. 19](#) ein Zeitverlaufdiagramm, das eine "Schiebe"-Geste veranschaulicht.
- [0052] [Fig. 20](#) ein Flussdiagramm, das den Betrieb der Schiebeeinheit von [Fig. 14](#) veranschaulicht.
- [0053] [Fig. 21](#) ist ein Blockdiagramm einer veranschaulichenden Anhebe-Sprung-Unterdrückungsschaltung, die bei der Gestenerkennung verwendet werden kann.
- [0054] Die vorliegende Erfindung setzt das von den vorliegenden Anmeldern in früheren Anmeldungen offenbarte Vorgehen fort und gibt einzigartigere Merkmale, die zuvor nicht verfügbar waren, an die Hand. Diese Verbesserungen bieten eine leichter zu integrierende Lösung, verbessertes Ansprechvermögen und größere Rauschunterdrückung, eine höhere Datenerfassungsgeschwindigkeit und geringeren Stromverbrauch. Die vorliegende Erfindung lässt eine kontinuierliche Selbstkalibrierung zur Ausschaltung der Wirkungen von Umgebungänderungen zu und ermöglicht eine verbesserte Cursorsteuerung durch Randweiterbewegung an einer Erfassungsebene.
- [0055] Der Durchschnittsfachmann wird erkennen, dass die folgende Beschreibung der vorliegenden Erfindung nur veranschaulichend und in keiner Weise einschränkend ist. Andere Ausführungen der Erfindung sind für einen solchen Fachmann nahe liegend.
- [0056] Die vorliegende Erfindung bringt in Kombination eine Reihe von einzigartigen Merkmalen zusammen, die neue zuvor nicht mögliche Anwendungen zulassen. Da der in der vorliegenden Erfindung verwendete Objektpositionssensor einen sehr geringen Energiebedarf hat, ist er vorteilhaft für den Gebrauch in batteriebetriebenen oder Niedrigstrom-Applikationen wie Laptops oder tragbaren Computern. Er ist auch eine sehr kostengünstige Lösung, hat keine sich bewegenden Teile (und ist daher praktisch wartungsfrei) und verwendet die vorhandenen Leiterplatten-Leiterbahnen für Sensoren. Die in der vorliegenden Erfindung verwendete Erfassungstechnologie kann in eine Hauptplatine eines Rechners integriert werden, um seine Kosten bei Computeranwendungen noch stärker zu senken. Analog kann der Sensor bei anderen Anwendungen Teil einer bereits vorhandenen gedruckten Schaltung sein.
- [0057] Aufgrund ihrer kleinen Größe und geringen Höhe ist die Sensortechnologie bei Laptop- oder tragbaren Applikationen brauchbar, bei denen Volumen ein wichtiger Aspekt ist. Die Sensortechnologie erfordert Leiterplattenplatz für nur einen einzigen Sensorschnittstellenchip, der direkt mit einem Mikroprozessor verbinden kann, plus die auf der Leiterplatte erforderliche Fläche für die Erfassung.
- [0058] Unter Bezug zunächst auf [Fig. 1](#) wird ein vereinfachtes Blockdiagramm eines kapazitiven Positionserfassungssystems **6** vorgestellt. Das kapazitive Positionserfassungssystem **6** kann die Position eines Fingers **8** oder eines anderen leitenden Objekts nahe einer bzw. unter Berührung einer Erfassungsebene **10** präzise ermitteln. Die Kapazität mehrerer leitender Leitungen, die in einer ersten Richtung (z.B. „X“) verlaufen, wird durch eine X-Eingabeverarbeitungsschaltung **12** erfasst, und die Kapazität mehrerer leitender Leitungen, die in einer zweiten Richtung (z.B. „Y“) verlaufen, wird durch eine Y-Eingabeverarbeitungsschaltung **14** erfasst. Die erfassten Kapazitätswerte werden sowohl in der X-Eingabeverarbeitungsschaltung **12** als auch der Y-Eingabeverarbeitungsschaltung **14** verarbeitet.

beitungsschaltung **14** digitalisiert. Die Ausgaben der X-Eingabeverarbeitungsschaltung **12** und der Y-Eingabeverarbeitungsschaltung **14** werden der Recheneinheit **16** vorgelegt, die die digitalen Angaben nutzt, um digitale Angaben abzuleiten, welche die Position und den Druck des Fingers **8** oder eines anderen leitenden Objekts gegenüber der Erfassungsebene **10** darstellen.

[0059] Die X-, Y- und Z-Ausgaben der Recheneinheit **16** werden zu einer Bewegungseinheit **18** geleitet, die dem Hostrechner die Cursorbewegungsrichtungssignale liefert. Der Durchschnittsfachmann wird erkennen, dass „Host“, so wie der Begriff hier verwendet wird, einen eigenständigen Rechner, zum Beispiel einen IBM oder einen von Apple Computers hergestellten kompatiblen PC oder Rechner, Handbediengeräte, PDAs (= persönliche digitale Assistenten), Fernkommunikationsgeräte oder dergleichen oder jedes andere Gerät oder System bedeutet, das die Ausgabe eines Berührungstabletts als Eingabe nehmen kann.

[0060] Die X-, Y- und Z-Ausgaben der Recheneinheit **16** werden ebenfalls an die Gesteneinheit **20** geleitet, die dazu dient, bestimmte von einem Bediener auf der Erfassungsebene **10** ausgeführte Fingergesten zu erkennen. Ferner kann die Gesteneinheit **20** ein Signal an die Bewegungseinheit **18** erzeugen, um das erfindungsgemäße Randweiterbewegungsmerkmal auf der Grundlage des Zustands der Gestenverarbeitung zu aktivieren.

[0061] Als Sensormaterial kann alles dienen, das die Herstellung einer leitenden X/Y-Feldmatrix zulässt. Dies umfasst nicht nur standardmäßige PC-Platinen, sondern schließt auch flexible PC-Platinen, leitende Elastomermaterialien, durch Siebdruck hergestellte leitende Leitungen und piezoelektrische Kynar-Kunststoffmaterialien ein, ist aber nicht hierauf beschränkt. Dadurch ist es auch bei tragbaren Geräteapplikationen oder bei der Schnittstelle zwischen Mensch und Rechner brauchbar, bei der der Sensor so geformt werden muss, dass er in die Hand passt.

[0062] Der Sensor kann jeder dreidimensionalen Oberfläche angeglichen werden. Kupfer kann in zwei Schichten auf nahezu jede Oberflächenkontur aufgebracht werden, die den Sensor erzeugt. Dadurch lässt sich der Sensor der besten erforderlichen ergonomischen Form einer bestimmten Applikation anpassen. In Verbindung mit dem Merkmal der „leichten Berührung“ macht ihn dies in vielen Anwendungen mühelos zu bedienen. Der Sensor kann auch indirekt verwendet werden, d.h. er kann über der Berührung erfassenden Oberfläche ein Isolierschaummaterial aufweisen, das von einer leitenden Schicht bedeckt ist, und kann zur Detektion eines beliebigen Objekts (nicht nur eines leitenden) verwendet werden, das gegen seine Oberfläche drückt.

[0063] Kleine Sensorflächen sind praktisch, d.h. eine derzeit vorgestellte Ausführung nimmt eine Fläche von etwa 1,5 Zoll (3,81 cm) × 1,5 Zoll (3,81 cm) ein, der Durchschnittsfachmann auf dem Gebiet wird aber erkennen, dass die Fläche für andere Anwendungen skalierbar ist. Die Matrixfläche ist durch entweder Verändern der Matrixleiterbahn-Abstände oder durch Verändern der Anzahl an Leiterbahnen skalierbar. Große Sensorflächen sind praktisch, wenn mehr Informationen erforderlich sind.

[0064] Neben einfachen X- und Y-Positionsangaben liefert die verwendete Sensortechnologie auch Fingerdruckangaben. Diese weitere Informationsgröße kann von Programmen zur Steuerung spezieller Merkmale wie „Strichbreiten“-Moden in Paint-Programmen, spezielle Menüzugriffe, etc. verwendet werden, wobei das Liefern einer natürlicheren sensorischen Eingabe an Rechner möglich wird. Es hat sich auch für das Implementieren von „Mausklick- und Mauszieh“-Moden und für einfache Eingabegesten als brauchbar erwiesen.

[0065] Der Bediener muss nicht einmal die Oberfläche berühren, um die Mindestreaktion zu erzeugen. Dieses Merkmal kann die Ermüdung des Bedieners stark minimieren und eine flexiblere Anwendung ermöglichen.

[0066] Das verwendete Erfassungssystem hängt von einem Wandlergerät ab, das Positions- und Druckangaben bezüglich des den Wandler berührenden Objekts liefern kann. Unter Bezug nun auf [Fig. 2a–2d](#) werden jeweils eine obere, eine untere, eine kombinierte und eine Querschnittsansicht einer derzeit bevorzugten Erfassungsebene **10** gezeigt, die eine Berührungssensormatrix **22** zur Verwendung in der vorliegenden Erfindung umfasst. Da durch diese Ausführung die Kapazität genutzt wird, ist die Oberfläche der Berührungssensormatrix **22** so ausgelegt, dass die kapazitive Kopplung mit einem Finger oder einem anderen leitenden Objekt maximiert wird.

[0067] Eine derzeit bevorzugte Berührungssensormatrix **22** umfasst einen Träger **24** mit einem Satz erster leitender Leiterbahnen **26**, die an einer oberen Fläche **28** desselben angeordnet sind und in einer ersten Richtung verlaufen, um Zeilenpositionen der Matrix zu umfassen. Ein zweiter Satz leitender Leiterbahnen **30** sind an einer unteren Fläche **32** desselben angeordnet und verlaufen in eine zweite Richtung, die vorzugsweise

rechtwinklig zur ersten Richtung ist, um die Spaltenpositionen der Matrix zu bilden. Die oberen und unteren leitenden Leiterbahnen **26** und **30** stehen abwechselnd mit periodischen Erfassungsfeldern **34** in Berührung, welche vergrößerte Flächen umfassen, die in [Fig. 2a–Fig. 2c](#) als Rauten gezeigt werden. Zwar werden die Erfassungsfelder **34** in [Fig. 2a–Fig. 2c](#) als Rauten gezeigt, doch ist jede Form, zum Beispiel Kreise, die deren enges Packen zulässt, für die Zwecke dieser Erfindung gleichwertig. Als willkürliche Gepflogenheit werden hier die ersten leitenden Leiterbahnen **26** als in „X“- oder „Zeilen“-Richtung ausgerichtet bezeichnet und können hier manchmal als „X-Leitungen“ bezeichnet werden und die zweiten leitenden Leiterbahnen **30** werden als in „Y“- oder „Spalten“-Richtung ausgerichtet bezeichnet und können hier manchmal als „Y-Leitungen“ bezeichnet werden.

[0068] Die Anzahl und Beabstandung dieser Erfassungsfelder **34** hängt von der erwünschten Auflösung ab. Bei einer tatsächlichen Ausführung zum Beispiel, die nach den Grundsätzen der vorliegenden Erfindung konstruiert wurde, wird ein Rautenmuster der Erfassungsfelder **34** mit 0,10 Zoll (0,254 cm) von Mitte zu Mitte, die entlang einer Matrix von 15 Zeilen und 15 Spalten von Leitern angeordnet sind, eingesetzt. Jedes zweite Erfassungsfeld **34** in jeder Richtung in dem Feldmuster ist mit leitenden Leiterbahnen an den oberen und unteren Oberflächen **28** und **32** des Trägers **24** jeweils verbunden.

[0069] Der Träger **24** kann eine Leiterplatte sein, eine flexible Leiterplatte oder eine beliebige einer Reihe von erhältlichen Schaltungsverbindungstechnologie-Strukturen. Seine Dicke ist unwesentlich, solange dadurch von den unteren leitenden Leiterbahnen **30** zu deren Erfassungsfeldern **34** auf der oberen Oberfläche **28** Kontakt hergestellt werden kann. Die Leiterplatte mit dem Träger **24** kann unter Verwendung standardmäßiger Branchenverfahren hergestellt sein. Die Dicke der Leiterplatte ist unwesentlich. Verbindungen von den leitenden Feldern **34** zu den unteren Leiterbahnen **30** können unter Verwendung standardmäßiger Durchkontaktierungsverfahren hergestellt werden, die in dem Gebiet der Leiterplatten bekannt sind.

[0070] In einer anderen Ausführung kann der Träger **24** eine Dicke in der Größenordnung von 0,005 Zoll (0,127 mm) bis 0,010 Zoll (0,254 mm) haben. Dann kann auf die Rauten auf der oberen Oberfläche **28** und die Durchkontaktierungen, die mit den Leiterbahnen **30** der unteren Oberfläche verbinden, verzichtet werden, was die Kosten des Systems weiter senkt.

[0071] Eine isolierende Schicht **36** ist über den Erfassungsfeldern **34** an der oberen Oberfläche **28** angeordnet, um einen menschlichen Finger oder ein anderes Objekt davon zu isolieren. Die isolierende Schicht **36** ist vorzugsweise eine dünne Schicht (d.h. etwa 5 mil), um die kapazitive Kopplung groß zu halten, und kann ein Material wie Mylar umfassen, das für seine schützenden und ergonomischen Eigenschaften gewählt wird. Der Begriff „erhebliche kapazitive Kopplung“, wie er hier verwendet wird, soll eine kapazitive Kopplung mit einer Größenordnung von über 0,5 pF bedeuten.

[0072] Es gibt zwei verschiedene kapazitive Wirkungen, die eintreten, wenn sich ein Finger der Berührungssensormatrix **22** nähert. Die erste kapazitive Wirkung ist die Transkapazität oder Kopplung zwischen den Erfassungsfeldern **34** und die zweite kapazitive Wirkung ist die Eigenkapazität oder Kopplung mit virtueller Erde. Die Erfassungsschaltung ist mit der Sensormatrix **22** gekoppelt und reagiert auf Änderungen einer oder beider dieser Kapazitäten. Dies ist wichtig, da sich die relativen Größen der beiden Kapazitäten abhängig vom Bedienerumfeld stark ändern. Die Fähigkeit, Änderungen sowohl der Eigenkapazität als auch der Transkapazität zu detektieren, ergibt ein sehr vielseitiges System mit einem großen Bereich an Anwendungen.

[0073] Nach der bevorzugten Ausführung detektiert ein Positionssensorsystem mit einer Berührungssensormatrix **22** und einer zugehörigen Positionsdetektionsschaltung eine Fingerposition auf einer Matrix von Leiterplatten-Leiterbahnen über die kapazitive Wirkung der Fingernähe zu der Sensormatrix **22**. Das Positionssensorsystem meldet die X,Y-Position eines nahe der Sensormatrix **22** platzierten Fingers mit viel feinerer Auflösung als die Beabstandung zwischen den Zeilen- und Spalten-Leiterbahnen **26** und **30**. Der Positionssensor nach dieser Ausführung meldet auch einen Z-Wert proportional zum Umriss dieses Fingers und somit den Druck anzeigend, mit dem der Finger die Oberfläche der isolierenden Schicht **36** über der Sensormatrix **22** berührt.

[0074] Nach der derzeit bevorzugten Ausführung kann eine sehr empfindliche, auf leichte Berührung ansprechende Detektorschaltung mit Hilfe adaptiver analoger und digitaler VLSI-Verfahren vorgesehen werden. Die Schaltung ist sehr robust und kalibriert Prozess- und systematische Fehler heraus. Die Detektorschaltung verarbeitet die kapazitiven Eingabeinformationen und liefert digitale Informationen, die einem Mikroprozessor direkt übergeben werden können.

[0075] Nach dieser Ausführung ist die Erfassungsschaltung auf einem integrierten Schaltungschip mit einem einzigen Sensorprozessor enthalten. Der Sensor-Prozessorchip kann eine beliebige Anzahl an X- und Y-„Matrix“-Eingängen haben. Die Anzahl an X- und Y-Eingängen muss nicht gleich sein. Die integrierte Schaltung hat einen digitalen Bus als Ausgang. In dem hier in [Fig. 2a–Fig. 2d](#) offenbarten veranschaulichenden Beispiel weist die Sensormatrix **22** 15 Leiterbahnen sowohl in X- als auch Y-Richtung auf. Der Sensor-Prozessorchip hat somit 15 X-Eingänge und 15 Y-Eingänge. Eine tatsächliche Ausführung, die nach den Grundsätzen konstruiert wurde, verwendete **18** Leiterbahnen in X-Richtung und **24** Leiterbahnen in Y-Richtung. Der Durchschnittsfachmann wird erkennen, dass die Größe der Erfassungsmatrix, die in der vorliegenden Erfindung eingesetzt werden kann, willkürlich ist und weitgehend von der gewählten Ausgestaltung diktiert wird.

[0076] Die X- und Y-Matrixknoten werden parallel angesteuert und erfasst, wobei die kapazitive Information von jeder Leitung anzeigt, wie nahe ein Finger sich diesem Knoten befindet. Die abgetasteten Informationen liefern ein Profil der Nähe des Fingers in jeder Dimension. Nach dieser Ausgestaltung wird das Profilverzerrungszentrum sowohl in X- als auch Y-Richtung abgeleitet und ist die Position in dieser Dimension. Die Profilkurve der Nähe wird auch integriert, um die Z-Information zu liefern.

[0077] Es gibt zwei Ansteuer- und Erfassungsverfahren, die bei der Berührungserfassungstechnologie eingesetzt werden. Nach einer ersten und derzeit bevorzugten Ausführung werden die Spannungen an allen X-Leitungen der Sensormatrix gleichzeitig bewegt, während die Spannungen der Y-Leitungen bei einer konstanten Spannung gehalten werden. Als Nächstes werden die Spannungen an allen der Y-Leitungen der Sensormatrix gleichzeitig bewegt, während die Spannungen der X-Leitungen bei einer konstanten Spannung gehalten werden. Dieses Abtastverfahren betont die Messung der Kapazität zur virtuellen Erde, die durch den Finger vorgegeben wird. Der Durchschnittsfachmann wird erkennen, dass die Reihenfolge dieser beiden Schritte etwas willkürlich ist und umgekehrt werden kann.

[0078] Nach einem zweiten Ansteuer-/Erfassungsverfahren werden die Spannungen an allen X-Leitungen der Sensormatrix gleichzeitig in eine positive Richtung bewegt, während die Spannungen der Y-Leitungen in eine negative Richtung bewegt werden. Als Nächstes werden die Spannungen an allen der X-Leitungen der Sensormatrix gleichzeitig in eine negative Richtung bewegt, während die Spannungen der Y-Leitungen in eine positive Richtung bewegt werden. Dieses zweite Ansteuer-/Erfassungsverfahren betont die Transkapazität und nimmt die Betonung weg von der Kapazität zur virtuellen Erde. Wie bei dem ersten Ansteuer-/Erfassungsverfahren wird der Durchschnittsfachmann erkennen, dass die Reihenfolge dieser beiden Schritte etwas willkürlich ist und umgekehrt werden kann.

[0079] Unter Bezug nun auf [Fig. 3](#) wird ein Blockdiagramm einer derzeit bevorzugten Erfassungsschaltung **40** vorgestellt. Dieses Blockdiagramm – sowie die begleitende Offenbarung – betrifft die Erfassungsschaltung **40** in nur einer Dimension (X) und enthält die X-Eingabeverarbeitungsschaltung **12** von [Fig. 1](#). Der Durchschnittsfachmann wird erkennen, dass eine identische Schaltung für das Erfassen der entgegengesetzten Dimension (Y) verwendet werden würde und die Y-Eingabeverarbeitungsschaltung **14** von [Fig. 1](#) beinhalten würde. Ein solcher Fachmann wird weiterhin erkennen, dass die beiden Dimensionen nicht rechtwinklig zueinander sein müssen. Sie können zum Beispiel radial oder von beliebig anderer Art sein, um der Kontur der Berührungssensormatrix und anderen Anforderungen des Systems zu entsprechen. Der Durchschnittsfachmann wird erkennen, dass die hier offenbarte Technologie ebenso auf einen eindimensionalen Fall übertragen werden könnte, bei dem nur ein Satz leitender Leiterbahnen verwendet wird.

[0080] Die Kapazität an jedem Sensormatrixknoten wird durch gleichwertige Kondensatoren **42-1** bis **42-n** wiedergegeben. Die Kapazität der Kondensatoren **42-1** bis **42-n** umfasst die Kapazität der Matrixleiter und hat einen charakteristischen Hintergrundwert, wenn kein Objekt (z.B. ein Finger) sich nahe der Erfassungsebene **10** der Sensormatrix befindet. Wenn sich ein Objekt der Erfassungsebene nähert, steigt die Kapazität der Kondensatoren **42-1** bis **42-n** proportional zur Größe und Nähe des Objekts.

[0081] Die Kapazität an jedem Knoten der Sensormatrix wird gleichzeitig mit Hilfe von Ladungsintegratorschaltungen **44-1** bis **44-n** gemessen. Die Ladungsintegratorschaltungen **44-1** bis **44-n** dienen zum Injizieren von Ladung in die Kapazitäten **42-1** bis jeweils **42-n** und zur Entwicklung einer Ausgangsspannung proportional zur an der entsprechenden X-Matrixleitung erfassten Kapazität. Daher werden die Ladungsintegratorschaltungen **44-1** bis **44-n** als bidirektionale Verstärkersymbole gezeigt. Jeder Ladungsintegratorschaltung **44-1** bis **44-n** wird durch eine Vorspannung erzeugende Schaltung **46** eine Betriebsvorspannung zugeführt.

[0082] Der Ausdruck „proportional zu Kapazität“, so wie er hier verwendet wird, bedeutet, dass das erzeugte Spannungssignal eine monotone Funktion der erfassten Kapazität ist. In der hier beschriebenen Ausführung

ist die Spannung direkt und linear proportional zu der erfassten Kapazität. Der Durchschnittsfachmann wird erkennen, dass andere monotone Funktionen, einschließlich aber nicht ausschließlich umgekehrte Proportionalität und nichtlineare Proportionalität, wie logarithmische oder exponentielle Funktionen, eingesetzt werden könnten, ohne von den hier offenbarten Grundsätzen abzuweichen. Ferner könnten Strom erfassende ebenso wie Spannung erfassende Verfahren eingesetzt werden.

[0083] Nach einem derzeit bevorzugten Ansteuer-/Erfassungsverfahren, das in der vorliegenden Erfindung verwendet wird, werden die Kapazitätsmessungen gleichzeitig über allen Eingängen in einer Dimension ausgeführt, um ein Problem zu überwinden, das bei allen vorbekannten Vorgehen, die individuelle Eingänge abtasten, inhärent ist. Das Problem bei dem vorbekannten Vorgehen ist, dass es gegenüber hoher Frequenz und starkem Amplitudenrauschen (starkes dv/dt -Rauschen), das mit der Schaltung über das berührende Objekt gekoppelt ist, empfindlich ist. Ein solches Rauschen kann das Fingerprofil verzerren, da aufgrund einer Änderung des Rauschpegels Rauschen in einem späteren Abtastzyklus, nicht aber in einem früheren auftritt.

[0084] Dieses Problem wird durch „Aufnahme eines Schnappschusses“ aller Eingänge gleichzeitig in X- und Y-Richtung (oder umgekehrt) gelöst. Da das injizierte Rauschen proportional zu der Stärke des Fingersignals über allen Eingängen ist, ist es daher um das Fingerzentrum herum symmetrisch. Da es um das Fingerzentrum herum symmetrisch ist, beeinträchtigt es nicht die Fingerposition. Ferner übt der Ladungsverstärker eine Differentialmessfunktion aus, um Gleichtaktrauschen weiter zu unterdrücken.

[0085] Aufgrund der Natur der Ladungsintegratorschaltungen **44-1** bis **44-n** ändern sich deren Ausgänge im zeitlichen Verlauf und haben nur kurze Zeit die erwünschte Spannungsausgabe. Wie derzeit bevorzugt wird, werden die Filterschaltungen **48-1** bis **48-n** als zwischen Abtasten und Halten geschaltete Kondensatorfilter implementiert.

[0086] Die erwünschte Spannung wird von den Filterschaltungen **48-1** bis **48-n** erfasst. Gesteuert durch die Steuerschaltung **56** filtern die Filterschaltungen **48-1** bis **48-n** jegliches Hochfrequenzrauschen aus dem erfassten Signal aus. Dies wird durch Wählen Kondensators für den Filter, der viel größer als die Ausgangskapazität der Ladungsintegratorschaltungen **44-1** bis **44-n** ist, verwirklicht. Ferner wird der Durchschnittsfachmann erkennen, dass die geschalteten Kondensatorfilterschaltungen **48-1** bis **48-n** die erwünschten Spannungen erfassen und diese speichern.

[0087] Die in Spannungsform aus den Kapazitätsmessungen erhaltenen Kapazitätsinformationen werden digitalisiert und in digitalem Format verarbeitet. Dementsprechend werden die von den Filterschaltungen **48-1** bis **48-n** gespeicherten Spannungen in Abtast-Halteschaltungen **50-1** bis **50-n** gespeichert, so dass der Rest der Schaltung gleichzeitig genommene Eingabedaten verarbeitet. Die Abtast-Halteschaltungen **50-1** bis **50-n** können als herkömmliche Abtast-Halteschaltungen konfiguriert werden, wie sie auf dem Gebiet bekannt sind.

[0088] Die abgetasteten analogen Spannungen an den Ausgängen der Abtast-Halteschaltungen **50-1** bis **50-n** werden durch Analog-Digital-Wandler **52** (A/D) digitalisiert. Wie derzeit bevorzugt wird, lösen die A/D-Wandler **52** die Eingangsspannung zu einem 10 Bit breiten digitalen Signal auf (eine Auflösung von einem Teil in 1.024), wenngleich der Durchschnittsfachmann erkennen wird, dass andere Auflösungen eingesetzt werden können. Die A/D-Wandler **52** können herkömmliche Wandler mit schrittweiser Annäherung sein, wie sie auf dem Gebiet bekannt sind.

[0089] Bei der verwendeten Ladungsintegratorschaltung liegt der Hintergrundwert (kein Objekt vorhanden) der Ausgänge des Ladungsintegrators bei etwa 1 Volt. Die sich aus dem Vorhandensein eines Fingers oder anderen Objekts ergebende ΔV liegt typischerweise bei etwa 0,4 Volt. Der Spannungsbereich der A/D-Wandler **52** sollte daher in dem Bereich von etwa 1 – 2 Volt liegen.

[0090] Ein wichtiger Aspekt sind die Mindest- und Höchstspannungsbezugspunkte für die A/D-Wandler **52** (V_{\min} und V_{\max}). Es wurde festgestellt, dass Rauschen Positionszittern verursacht, wenn diese Bezugsspannungen feste Punkte sind. Eine Lösung dieses Problems, die in der vorliegenden Erfindung eingesetzt wird, ist das dynamische Erzeugen der V_{\min} - und V_{\max} -Bezugsspannungen aus den Bezugskapazitäten **42-Vmin** und **42-Vmax**, die von den Ladungsintegratorschaltungen **44-Vmin** und **44-Vmax** erfasst und von den Filterschaltungen **48-Vmin** und **48-Vmax** verarbeitet und in den Abtast-Halteschaltungen **50-Vmin** und **50-Vmax** gespeichert werden. Auf diese Weise ist jedes Gleichtakt-Rauschen, das bei Abtasten der Signale von der Sensor-matrix vorliegt, auch in den V_{\min} - und V_{\max} -Bezugsspannungswerten vorhanden und neigt zu Aufheben. Der Durchschnittsfachmann wird erkennen, dass die Bezugskapazitäten **42-Vmin** und **42-Vmax** entweder diskrete Kondensatoren oder extra Leiterbahnen in der Sensor-matrix sein können.

[0091] Die V_{\min} -Bezugsspannung wird von einem Kondensator mit einem Wert gleich der in der Sensormatrix ohne Vorliegen eines Objekts niedrigsten zu erwartenden Kapazität erzeugt (etwa 12 pF unter der Annahme einer 2 Quadratzoll großen Sensormatrix). Die V_{\max} Bezugsspannung wird von einem Kondensator mit einem Wert gleich der in der Sensormatrix bei Vorliegen eines Objekts höchsten zu erwartenden Kapazität erzeugt (etwa 16 pF unter der Annahme einer 2 Quadratzoll großen Sensormatrix).

[0092] Die Ausgaben der A/D-Wandler **52** liefern Eingaben für die Recheneinheit **16**. Wie unter Bezug auf [Fig. 8](#) eingehender offenbart wird, besteht die Funktion der Recheneinheit **16** darin, das gewichtete Mittel der Signale an den einzelnen Erfassungsleitungen sowohl in X- als auch Y-Richtung in der Berührungssensormatrix **22** zu berechnen. Dadurch wird die Recheneinheit **16** gemeinsam von der X-Eingabeverarbeitungsschaltung **12** und der Y-Eingabeverarbeitungsschaltung **14** benutzt, wie in [Fig. 1](#) gezeigt wird.

[0093] Die Steuerschaltung **56** von [Fig. 3](#) lenkt den Betrieb der restlichen Schaltung. Da das System diskret abgetastet und in seinem Betrieb parallelisiert wird, ist die Steuerschaltung **56** vorhanden, um den Signalfloss zu bewältigen. Die von der Steuerschaltung **56** ausgeführten Funktionen können herkömmlicherweise über eine auf dem Gebiet allgemein als Zustandsmaschine oder Mikroregler bekannte Vorrichtung entwickelt werden.

[0094] Nun werden der Aufbau und die Arbeitsweise der einzelnen Blöcke von [Fig. 3](#) offenbart. Unter Bezug nun auf [Fig. 4a](#), [Fig. 4b](#) und [Fig. 5](#) wird eine typische Ladungsintegratorschaltung beschrieben. Die Ladungsintegratorschaltung **44** wird in [Fig. 4a](#) als vereinfachtes schematisches Diagramm und in [Fig. 4b](#) als veranschaulichendes schematisches Diagramm gezeigt. Der Zeitablauf des Betriebs der Ladungsintegratorschaltung **44** wird in [Fig. 5](#) gezeigt. Diese Zeitsignale werden von dem Steuerblock **56** geliefert.

[0095] Die Ladungsintegratorschaltung **44** beruht auf dem grundlegenden physikalischen Phänomen der Verwendung von Strom zur Ladung eines Kondensators. Wird der Kondensator über eine konstante Zeit durch einen konstanten Strom geladen, dann wird an dem Kondensator eine Spannung erzeugt, die umgekehrt proportional zur Kapazität ist. Die zu ladende Kapazität ist die Sensormatrixleitungskapazität **42** parallel zu einem internen Kondensator. Dieser interne Kondensator enthält die interessierende Spannung.

[0096] Unter Bezug zunächst auf [Fig. 4a](#) wird ein vereinfachtes schematisches Diagramm einer veranschaulichenden Ladungsintegratorschaltung **44** gezeigt. Ein Ladungsintegratorschaltungs-Eingangsknoten **60** ist mit einer der X- (oder Y-) Leitungen der Sensormatrix verbunden. Ein erster unterbrechender Schalter **62** ist zwischen dem Ladungsintegratorschaltungs-Eingangsknoten **60** und V_{DD} , dem positiven Versorgungsleiter, angeschlossen. Ein zweiter unterbrechender Schalter **64** ist zwischen dem Ladungsintegratorschaltungs-Eingangsknoten **60** und Erde, dem negativen Versorgungsleiter, angeschlossen. Eine Quelle positiven Konstantstroms **66** ist mit V_{DD} , dem positiven Versorgungsleiter, und mit dem Ladungsintegratorschaltungs-Eingangsknoten **60** und durch einen ersten Stromquellenschalter **68** verbunden. Eine Quelle negativen Konstantstroms **70** ist mit Erde und mit dem Ladungsintegratorschaltungs-Eingangsknoten und durch einen zweiten Stromquellenschalter **72** verbunden. Es ist offensichtlich, dass eine andere hohe oder niedrige Spannungsversorgung an Stelle von V_{DD} und Erde verwendet werden könnten.

[0097] Ein erster interner Kondensator **74** ist zwischen V_{DD} und dem Ausgangsknoten **76** der Ladungsintegratorschaltung **44** angeschlossen. Ein Schalter für die Speicherung positiver Spannung **78** ist zwischen dem Ausgangsknoten **76** und dem Eingangsknoten **60** angeschlossen. Ein zweiter interner Kondensator **80** ist mit einer seiner Platten durch einen Schalter **82** mit Erde und durch einen Schalter **84** mit dem Ausgangsknoten **76** der Ladungsintegratorschaltung **44** verbunden und mit der anderen seiner Platten durch einen Schalter zur Speicherung negativer Spannung **86** mit dem Eingangsknoten **60** und durch einen Schalter **88** mit V_{DD} verbunden. Die Kapazität des ersten und des zweiten internen Kondensators **74** und **80** sollte ein kleiner Bruchteil (d.h. etwa 10%) der Kapazität der einzelnen Sensormatrixleitungen sein. In einer typischen Ausführung beträgt die Sensormatrixleitungskapazität etwa 10 pF und die Kapazität der Kondensatoren **74** und **80** sollte bei etwa 1 pF liegen.

[0098] Nach der derzeit bevorzugten Ausführung besteht das genutzte Vorgehen aus einer Differentialmessung für zusätzliche Störfestigkeit, deren Vorteil ist, dass jegliches Niederfrequenz-Gleichtaktrauschen beseitigt wird. Für die folgende Diskussion sollte angenommen werden, dass alle Schalter offen sind, sofern sie nicht als geschlossen angegeben werden. Zuerst wird die Sensormatrixleitung vorübergehend durch Schalter **62** mit V_{DD} kurzgeschlossen, Schalter **78** wird geschlossen, was den Kondensator **74** parallel mit der Kapazität der Sensorleitung schaltet. Dann wird die parallele Kondensatorkombination mit einem Konstantstrom von der Stromquelle **70** durch Schalter **72** über eine feste Zeitlänge entladen. Am Ende der festen Zeitlänge wird der

Schalter **78** geöffnet, wodurch die Spannung an der Sensormatrixleitung an dem Kondensator **74** gespeichert wird.

[0099] Die Sensorleitung wird dann vorübergehend durch Schalter **64** mit Erde kurzgeschlossen und die Schalter **82** und **86** werden geschlossen, um den Kondensator **80** parallel mit der Kapazität der Sensorleitung zu schalten. Der Schalter **68** wird geschlossen und die parallele Kondensatorkombination wird über eine feste Zeitlänge gleich der festen Zeitlänge des ersten Zyklus mit Konstantstrom von der Stromquelle **66** geladen. Am Ende der festen Zeitlänge wird der Schalter **86** geöffnet, wodurch die Spannung an der Sensormatrixleitung an dem Kondensator **80** gespeichert wird.

[0100] Die erste und die zweite gemessene Spannung werden dann gemittelt. Dies wird durch Öffnen von Schalter **82** und Schließen der Schalter **88** und **84** verwirklicht, was den Kondensator **80** parallel mit Kondensator **74** schaltet. Da die Kondensatoren **74** und **80** die gleiche Kapazität haben, ist die sich ergebende Spannung über ihnen gleich dem Mittel der Spannungen über jedem einzelnen. Dieses Endergebnis ist der Wert, der dann zu der geeigneten der Filterschaltungen **48-1** bis **48-n** weitergegeben wird.

[0101] Das Niederfrequenzrauschen, insbesondere 50/60 Hz und dessen Oberschwingungen, verhält sich wie eine Gleichstromkomponente, die in der einen Messung addiert und in der anderen subtrahiert. Werden die beiden Ergebnisse addiert, mittelt diese Rauschkomponente auf Null. Der Betrag der Störunterdrückung ist eine Funktion, wie schnell in Folge die beiden entgegengesetzten Auf- und Entladezyklen ausgeführt werden, wie hier offenbart wird. Einer der Gründe für die Wahl dieser Ladungsintegratorschaltung ist, dass sie das schnelle Vornehmen von Messungen zulässt.

[0102] Unter Bezug zunächst auf [Fig. 4b](#) wird ein umfassenderes schematisches Diagramm einer veranschaulichenden Ausführung der Ladungsintegratorschaltung **44** des vereinfachten Diagramms von [Fig. 4a](#) gezeigt. Der Eingangsknoten **60** wird durch Durchgangsgatter **90** und **92**, die die Schalter **62** und **64** von [Fig. 4a](#) ersetzen, mit V_{DD} und Erde verbunden gezeigt. Durchgangsgatter **90** wird durch ein Signal ResetUp gesteuert, das seinem Steuereingang vorlegt wird, und Durchgangsgatter **92** wird durch ein Signal ResetDn gesteuert, das seinem Steuereingang vorgelegt wird. Der Durchschnittsfachmann wird erkennen, dass die Durchgangsgatter **90** und **92** ebenso wie alle andere Durchgangsgatter, die durch das gleiche Symbol in [Fig. 4b](#) dargestellt werden, herkömmliche CMOS-Durchgangsgatter sein können, wie sie auf dem Gebiet bekannt sind. Die hier verwendete Konvention ist, dass das Durchgangsgatter aus ist, wenn sein Steuereingang niedrig gehalten wird, und dass es ein ist und eine Verbindung niedriger Impedanz bietet, wenn sein Steuereingang hoch gehalten wird.

[0103] Die P-Kanal-MOS-Transistoren **94** und **96** sind als Stromspiegel konfiguriert. Der P-Kanal-MOS-Transistor **94** dient als Stromquelle **66** und Durchgangsgatter **98** dient als Schalter **68** von [Fig. 4a](#). Der Steuereingang von Durchgangsgatter **98** wird durch ein Signal StepUp gesteuert.

[0104] Die N-Kanal-MOS-Transistoren **100** und **102** sind als Stromspiegel konfiguriert. Der N-Kanal-MOS-Transistor **100** dient als Stromquelle **70** und Durchgangsgatter **104** dient als Schalter **72** von [Fig. 4a](#). Der Steuereingang von Durchgangsgatter **104** wird durch ein Signal StepDn gesteuert. P-Kanal-MOS-Transistor **106** und N-Kanal-MOS-Transistor **108** sind in Reihe mit P-Kanal-MOS-Stromspiegeltransistor **96** und N-Kanal-MOS-Stromspiegeltransistor **102** angeordnet. Das Steuergatter des P-Kanal-MOS-Transistors **106** wird durch ein Freigabesignal EN angesteuert, das den P-Kanal-MOS-Transistor **106** einschaltet, um die Stromspiegel einzuschalten. Diese Vorrichtung dient als Stromsparvorrichtung, so dass die Ladungsintegratorschaltung **44** abgeschaltet werden kann, um Strom zu sparen, wenn sie nicht im Einsatz ist.

[0105] Das Gatter des N-Kanal-MOS-Transistors **108** wird durch eine Bezugsspannung V_{bias} angesteuert, die den Strom durch die Stromspiegeltransistoren **96** und **102** setzt. Die Spannung V_{bias} wird durch eine Servo-Rückkopplungsschaltung, wie sie unter Bezug auf [Fig. 10](#) eingehender offenbart wird, gesetzt. Der Durchschnittsfachmann wird verstehen, dass diese Ausführung das Erfolgen einer Kalibrierung in Echtzeit (über eine lang dauernde, konstante Rückkopplung) zulässt, wodurch alle langfristigen Wirkungen aufgrund Sensorumfeldbedingungen beseitigt werden. In einer derzeitigen Ausführung der Erfindung ist V_{bias} allen Ladungsintegratorschaltungen **44-1** bis **44-n** und **44-Vmax** und **44-Vmin** zueigen.

[0106] Zu beachten ist, dass eine entsprechende Bemessung der N-Kanal-MOS-Transistoren **102** und **108** einen Temperatenausgleich bieten kann. Dies wird durch Ausnutzen der Tatsache verwirklicht, dass der Schwellwert des N-Kanal-MOS-Transistors **108** temperaturabhängig sinkt, während die Mobilität der beiden N-Kanal-MOS-Transistoren **102** und **108** temperaturabhängig sinkt. Das Sinken des Schwellwerts hat die Wir-

kung eines Anhebens des Stroms, während die Mobilitätsabnahme die Wirkung eines Senkens des Stroms hat. Durch geeignetes Bemessen der Vorrichtung können sich diese Wirkungen über einen erheblichen Teil des Betriebsbereichs gegenseitig aufheben.

[0107] Der Kondensator **74** hat eine mit V_{DD} verbundene Platte und die andere Platte ist mit dem Ausgangsknoten **76** und dem Eingangsknoten **60** durch Durchgangsgatter **110** verbunden, das in [Fig. 4a](#) als Schalter **78** gezeigt ist. Der Steuereingang von Durchgangsgatter **110** wird durch das Steuersignal S_{Up} gesteuert. Eine Platte des Kondensators **80** ist durch Durchgangsgatter **112** (Schalter **86** in [Fig. 4a](#)) mit Eingangsknoten **60** und durch Durchgangsgatter **114** (Schalter **82** in [Fig. 4a](#)) mit V_{DD} verbunden. Der Steuereingang von Durchgangsgatter **112** wird durch das Steuersignal S_{Dn} angesteuert und der Steuereingang von Durchgangsgatter **114** wird durch das Steuersignal Ch_{Up} angesteuert. Die andere Platte des Kondensators **80** ist durch N-Kanal-MOS-Transistoren **116** (Schalter **82** in [Fig. 4a](#)) mit Erde und durch Durchgangsgatter **118** (Schalter **84** in [Fig. 4a](#)) mit Ausgangsknoten **76** verbunden. Der Steuereingang von Durchgangsgatter **118** wird durch das Steuersignal $Share$ angesteuert.

[0108] Unter Bezug nun auf [Fig. 4a](#), [Fig. 4b](#) und das Zeitverlaufdiagramm von [Fig. 5](#) kann der Betrieb der Ladungsintegratorschaltung **44** während eines Abtastzyklus beobachtet werden. Zuerst wird das Steuersignal EN (Freigabe) durch Gehen auf Ov auf aktiv. Dies schaltet die Stromspiegel ein und schaltet die Lade- und Entladestromquellen, die MOS-Transistoren **94** und **100**, ein. Das Steuersignal $Reset_{Up}$ ist zu diesem Zeitpunkt aktiv hoch, was den Eingangsknoten **60** (und die Sensorleitung, mit der er verbunden ist) mit V_{DD} kurzschließt. Das Steuersignal S_{Up} ist zu diesem Zeitpunkt ebenfalls aktiv hoch, was den Kondensator **74** und den Ausgangsknoten **76** mit dem Eingangsknoten **60** verbindet. Diese Anordnung garantiert, dass der folgende Entladeteil des Betriebszyklus immer von einem bekannten Gleichgewichtszustand aus einsetzt.

[0109] Der Entladeprozess beginnt, nachdem das Steuersignal $Reset_{Up}$ auf inaktiv wechselt. Das Steuersignal $Step_{Dn}$ wird aktiv, was den MOS-Transistor **100**, die Entladestromquelle, mit dem Eingangsknoten **60** und seiner zugehörigen Sensorleitung verbindet. $Step_{Dn}$ ist eine festgelegte Zeitdauer aktiv und die Quelle negativen Konstantstroms entlädt die kombinierte Kapazität der Sensorleitung und des Kondensators **74**, wodurch dessen Spannung während dieser Zeit gesenkt wird. $Step_{Dn}$ wird dann abgeschaltet. Kurze Zeit später wird das Steuersignal S_{Up} inaktiv, wobei die gemessene Spannung an dem Kondensator **74** gespeichert wird. Dies beendet den Entladezyklus.

[0110] Als Nächstes wird das Steuersignal $Reset_{Dn}$ aktiv und schließt die Sensorleitung mit Erde kurz. Gleichzeitig werden die Steuersignale S_{Dn} und Ch_{Dn} aktiv und schließen den Kondensator **80** zwischen Erde und der Sensorleitung an. Der Kondensator **80** wird gegenüber der Erde entladen, was garantiert, dass der folgende Aufladezyklus immer bei einem bekannten Zustand einsetzt.

[0111] Der Aufladezyklus beginnt, nachdem das Steuersignal $Reset_{Dn}$ inaktiv wird und das Steuersignal $Step_{Up}$ aktiv wird. Zu diesem Zeitpunkt wird die Stromladequelle, der MOS-Transistor **94**, mit der Sensorleitung verbunden und liefert einen Konstantstrom, um die Sensorleitung durch Anheben der Spannung an dieser zu laden. Das Steuersignal $Step_{Up}$ ist eine feste Zeitdauer lang aktiv (vorzugsweise gleich der Zeit für den vorstehend erwähnten Zyklus), was das Laden der Kapazität zulässt, und wird dann ausgeschaltet. Das Steuersignal S_{Dn} wird dann inaktiv, was die gemessene Spannung über dem Kondensator **80** belässt.

[0112] Nun beginnt der Mittelungszyklus. Zuerst wird die Spannung an dem Kondensator **80** pegelverschoben. Dies erfolgt durch Inaktivwerden des Steuersignals Ch_{Dn} , was eine Platte des Kondensators **80** potentialfrei lässt. Dann wird das Steuersignal Ch_{Up} aktiv, was die zweite Platte des Kondensators mit V_{DD} verbindet. Dann wird das Steuersignal $Share$ aktiv, was die erste Platte des Kondensators **80** mit dem Ausgangsknoten **76** verbindet, wodurch die Kondensatoren **74** und **80** parallel geschaltet werden. Dies hat die Wirkung einer Mittlung der Spannungen über den beiden Kondensatoren, wodurch Gleichtaktrauschen wie vorstehend beschrieben heraussubtrahiert wird. Diese mittlere Spannung ist dann auch am Ausgangsknoten **76** verfügbar.

[0113] Der Durchschnittsfachmann wird erkennen, dass der Umgebungswechselstrom und ein anderes Niederfrequenzstörungenbeseitigungsmerkmal, das beim Mitteln der Spannungen inhärent ist, die in den Entlade- und Ladezyklen erhalten werden, äußerst effektiv ist, wenn die beiden Zyklen sehr zeitnah zueinander ausgeführt werden. Die Signale Ch_{Dn} und Ch_{Up} sollten bezüglich einander innerhalb einer Zeit umgesetzt werden, die weit kürzer als ein Viertel des Zeitraums des aufzuhebenden Rauschens ist, um diesen Vorteil s Merkmal der vorliegenden Erfindung zu nutzen.

[0114] Es wurden zwei verschiedene Ansteuer-/Erfassungsverfahren offenbart. Der Durchschnittsfachmann

wird mühelos erkennen, dass die unter Bezug auf [Fig. 4a](#), [Fig. 4b](#) und [Fig. 5](#) offenbarte Ladungsintegratorschaltung **44** so anpassbar ist, dass sie nach jeder hierin offenbarten Abtastmethode arbeiten kann.

[0115] Wie aus dem Verständnis des Betriebs der Ladungsintegratorschaltung **44** klar ist, ist deren Ausgangsspannung nur kurze Zeit verfügbar und unterliegt Umgebungsrauschen. Um die Wirkungen des Rauschens zu minimieren, wird eine geschaltete Kondensator-Filterschaltung **48** verwendet. Unter Bezug nun auf [Fig. 6](#) wird ein schematisches Diagramm einer veranschaulichenden geschalteten Kondensator-Filterschaltung **48** gezeigt, die in der vorliegenden Erfindung verwendet werden kann. Der Durchschnittsfachmann wird diese geschaltete Kondensator-Filterschaltung erkennen, welche einen Eingangsknoten **120**, ein Durchgangsgatter **122** mit einem von einem Steuersignal Sample angesteuerten Steuereingang, einen zwischen dem Ausgang des Durchgangsgatters **126** und einer festgelegten Spannung wie Erde angeschlossenen Kondensator **124** sowie einen Ausgangsknoten mit der gemeinsamen Verbindung zwischen dem Kondensator **124** und dem Ausgang des Durchgangsgatters **122** umfasst. Bei einer typischen Ausführung hat der Kondensator **116** eine Kapazität von etwa 10 pF.

[0116] Wie für einen Durchschnittsfachmann nachvollziehbar ist, ist die geschaltete Kondensator-Filterschaltung **48** zum Teil eine Abtast-Halteschaltung und hat eine Filterzeitkonstante, die K mal der Abtastdauer ist, wobei K das Verhältnis des Kondensators **124** zur Summe der Kondensatoren **74** und **80** der Ladungsintegratorschaltung **44** von [Fig. 4a](#) und [Fig. 4b](#) ist, mit denen er verbunden ist. Die geschaltete Kondensator-Filterschaltung **48** reduziert das Injizieren von Rauschen im System weiter. Bei der bevorzugten Ausführung ist $K=10/2=5$. Der Durchschnittsfachmann wird erkennen, dass andere Arten von Filterschaltungen wie RC-Filter in der vorliegenden Erfindung verwendet werden können.

[0117] Unter Bezug nun auf [Fig. 7](#) wird ein detaillierteres Blockdiagramm einer derzeit bevorzugten Anordnung der A/D-Wandler **52** von [Fig. 3](#) vorgestellt. Es gibt weniger A/D-Wandler **52** als Leitungen in der Berührungssensormatrix, und die Eingänge zu den A/D-Wandlern **52** werden multiplex betrieben, um jeden der einzelnen A/D-Wandler unter mehreren Leitungen in der Berührungssensormatrix gemeinsam zu nutzen. Die Anordnung in [Fig. 7](#) ist bei der Verwendung der Layoutfläche der integrierten Schaltung effizienter als das Vorsehen einzelner A/D-Wandler für jede Eingangsleitung.

[0118] In der in [Fig. 7](#) dargestellten Ausführung werden 24 leitende Leitungsleiterbahnen für die Sensormatrix **10** von [Fig. 2a-Fig. 2d](#) angenommen. Wie in [Fig. 7](#) gezeigt wird, werden die Ausgänge der Abtast-Halteschaltungen **50-1** bis **50-24** den analogen Dateneingängen des analogen Multiplexer **130** zugeführt. Der analoge Multiplexer **130** hat sechs Ausgänge, wovon jeder den Eingang eines einzelnen A/D-Wandlers **52-1** bis **52-6** ansteuert. Die interne Anordnung des analogen Multiplexers **130** ist so, dass vier verschiedene der Eingänge mit jedem der Ausgänge multiplex verbunden werden. Der analoge Multiplexer **130** wurde vom Konzept her mit sechs internen Multiplexerblöcken **132-1** bis **132-6** gezeichnet.

[0119] Bei dem in [Fig. 7](#) gezeigten Beispiel sind von den Abtast-Halteschaltungen **50-1** bis **50-4** genommene Eingänge mit dem Ausgang des internen Multiplexerblocks **132-1** multiplex verbunden, der den A/D-Wandler **52-1** ansteuert. Analog sind von den Abtast-Halteschaltungen **50-5** bis **50-8** genommene Eingänge mit dem Ausgang des internen Multiplexerblocks **132-2** multiplex verbunden, der den A/D-Wandler **52-2** ansteuert, von den Abtast-Halteschaltungen **50-9** bis **50-12** genommene Eingänge sind mit dem Ausgang des internen Multiplexerblocks **132-3** multiplex verbunden, der den A/D-Wandler **52-3** ansteuert, von den Abtast-Halteschaltungen **50-13** bis **50-16** genommene Eingänge sind mit dem Ausgang des internen Multiplexerblocks **132-4** multiplex verbunden, der den A/D-Wandler **52-4** ansteuert, von den Abtast-Halteschaltungen **50-17** bis **50-20** genommene Eingänge sind mit dem Ausgang des internen Multiplexerblocks **132-5** multiplex verbunden, der den A/D-Wandler **52-5** ansteuert, und von den Abtast-Halteschaltungen **50-21** bis **50-24** genommene Eingänge sind mit dem Ausgang des internen Multiplexerblocks **132-6** multiplex verbunden, der den A/D-Wandler **52-6** ansteuert.

[0120] Der analoge Multiplexer **130** hat einen Satz von Steuereingängen, die schematisch durch Bus **134** dargestellt werden. In der in [Fig. 7](#) gezeigten veranschaulichenden Ausführung sind alle der internen Multiplexer **132-1** bis **132-6** Multiplexer mit vier Eingängen, und daher kann der Steuerbus **134** einen 2-Bit-Bus für eine Eins-Aus-Vier-Auswahl umfassen. Der Durchschnittsfachmann wird erkennen, dass die Anordnung von [Fig. 7](#) lediglich eine einer Reihe spezifischer Lösungen der Aufgabe einer A/D-Umwandlung von 24 Kanälen ist und dass andere zufrieden stellende, gleichwertige Anordnungen möglich sind.

[0121] Bei einem unkomplizierten Entschlüsselungsschema geben die Multiplexer **132-1** bis **132-6** in Folge die an ihren ersten bis vierten Eingängen anliegenden analogen Spannungen jeweils weiter zu den Eingängen

der A/D-Wandler **52-1** bis **52-6**. Nachdem sich die analogen Werte in den Eingängen der A/D-Wandler **52-1** bis **52-6** stabilisiert haben, wird an einer gemeinsamen A/D-Steuerleitung **136** ein Befehl CONVERT (= Umwandeln) geltend gemacht, um den A/D-Umwandlungsprozess einzuleiten.

[0122] Nach Beendigung des A/D-Umwandlungsprozesses wird der die Eingangsspannung darstellende digitale Wert in Registern **138-1** bis **138-6** gespeichert. Wie derzeit bevorzugt können die Register **138-1** bis **138-6** jeweils ein Zweiwort-Register umfassen, so dass ein Wort aus den Registern **138-1** bis **138-6** zur Recheneinheit **16** ausgelesen wird, während ein zweites Wort in die Register **138-1** bis **138-6** geschrieben wird, um die Systemgeschwindigkeit zu erhöhen. Die Auslegung solcher Register **138-1** bis **138-6** ist in dem Gebiet üblich.

[0123] Unter Bezug nun auf [Fig. 8](#) wird ein detaillierteres Blockdiagramm der Recheneinheit **16** vorgestellt. Der Durchschnittsfachmann wird nachvollziehen, dass die Recheneinheit **16** Informationen aus sowohl der X- als auch der Y-Dimension verarbeitet, d.h. von der X-Eingabeverarbeitungsschaltung **12** und der Y-Eingabeverarbeitungsschaltung **14** von [Fig. 1](#).

[0124] Vor der Offenbarung der strukturellen Konfiguration der Recheneinheit **16** ist es hilfreich, das bevorzugte Verfahren zu verstehen, nach dem die Zentroidposition eines Objekts nahe der Sensormatrix **22** erfindungsgemäß ermittelt wird.

[0125] Die Objektposition kann in jeder Richtung durch Beurteilen des gewichteten Mittels der an der einzelnen Erfassungsleitung der Sensormatrix **10** gemessenen Kapazitäten ermittelt werden. In der folgenden Diskussion wird die X-Richtung verwendet, der Durchschnittsfachmann wird aber erkennen, dass die Diskussion ebenso auf die Ermittlung des gewichteten Mittels in der Y-Richtung zutrifft. Wie gut bekannt ist, kann das gewichtete Mittel wie folgt ermittelt werden:

$$X \text{ position} = \frac{\sum_{i=0}^n i \times \Delta C_i}{\sum_{i=0}^n \Delta C_i}$$

[Gleichung 1]

wobei $\Delta C_i = C_i - C_{0i}$ ist, C_i die derzeit an i-ten Leiterbahn gemessene Kapazität ist und C_{0i} der an der gleichen Leiterbahn zu einem vergangenen Zeitpunkt gemessene Wert ist, bei dem kein Objekt vorlag. Bezüglich dieser vergangenen und derzeitigen Kapazitätsmessungen kann die Position ausgedrückt werden als:

$$X \text{ position} = \frac{\sum_{i=0}^n i \times (C_i - C_{0i})}{\sum_{i=0}^n (C_i - C_{0i})}$$

[Gleichung 2]

[0126] Unter Nutzung der distributiven Eigenschaft der Multiplikation gegenüber der Addition wird dieser Ausdruck als gleichwertig gesehen mit:

$$X_{\text{position}} = \frac{-\sum_{i=0}^n (i \times C_{0i}) + \sum_{i=0}^n (i \times C_i)}{-\sum_{i=0}^n (C_{0i}) + \sum_{i=0}^n (C_i)}$$

[Gleichung 3]

wobei die negativen Terme sowohl im Zähler als auch im Nenner Offsets sind und den Hintergrundwert der Kapazitäten bei Nichtvorhandensein eines Objekts darstellen. Wenn der Term O_N zur Darstellung des Zähleroffsets verwendet wird und der Term O_D zur Darstellung des Nenneroffsets verwendet wird, kann Gleichung 3 wie folgt umgeschrieben werden:

$$X_{\text{position}} = \frac{-O_N + \sum_{i=0}^n (i \times C_i)}{-O_D + \sum_{i=0}^n (C_i)}$$

[Gleichung 4]

[0127] Unter Bezug nun auf [Fig. 8](#) ist ersichtlich, dass die Recheneinheit **16** X-Zähler- und Nenner-Akkumulatoren **150** und **152** sowie Y-Zähler- und Nenner-Akkumulatoren **154** und **156** umfasst. Die Quelle der Rechengrößendaten für die X-Zähler- und Nenner-Akkumulatoren **150** und **152** und die Y-Zähler- und Nenner-Akkumulatoren **154** und **156** sind die Register **138-1** bis **138-6** in jeder (X und Y) Richtung der Sensormatrix **22** von [Fig. 1](#). Die X- und Y-Nenner-Akkumulatoren **152** und **156** addieren die digitalen Ergebnisse aus den A/D-Umwandlungen. Die X- und Y-Zähler-Akkumulatoren **150** und **154** berechnen die gewichtete Summe der Eingabedaten an Stelle der direkten Summe. Die Akkumulatoren **150**, **152**, **154** und **156** können als Hardwareelemente konfiguriert werden oder als Software, die auf einem Mikroprozessor läuft, wie für den Durchschnittsfachmann problemlos verständlich ist. Wie aus einer Prüfung von [Fig. 8](#) hervorgeht, berechnen die Zähler-Akkumulatoren **150** und **154** den Ausdruck von Gleichung 4:

$$\sum_{i=0}^n i \times C_i$$

[Gleichung 5]

und die Nenner-Akkumulatoren **152** und **156** berechnen den Ausdruck von Gleichung 4:

$$\sum_{i=0}^n C_i$$

[Gleichung 6]

[0128] Der Inhalt der X- und Y-Zähler- und Nenner-Offsetregister **158**, **160**, **162** und **164** wird von den in den Akkumulatoren **150**, **152**, **154** und **156** gespeicherten Ergebnissen in den Addierern **166**, **168**, **170** und **172** subtrahiert. Addierer **166** subtrahiert den im X-Zähler-Offsetregister **158** gespeicherten Offset O_{NX} . Addierer **168** subtrahiert den im X-Nenner-Offsetregister **160** gespeicherten Offset O_{DX} . Addierer **170** subtrahiert den im Y-Zähler-Offsetregister **162** gespeicherten Offset O_{NY} . Addierer **172** subtrahiert den im Y-Nenner-Offsetregister **164** gespeicherten Offset O_{DY} . Die Zähler/Nenner-Paare werden durch Divisionsblöcke **174** und **176** dividiert, um die X- und Y-Positionsdaten zu erzeugen, und das X- und Y-Zählerpaar wird von Block **178** verwendet, um Z-Achsendaten (Druck) zu erzeugen. Die durch Block **178** ausgeführte Funktion wird hier später offenbart. Die Offsets O_{DX} , O_{NX} , O_{DY} und O_{NY} werden von dem Akkumulatorinhalt abgefragt, wenn dies von der Kalibrierungseinheit **180** angeordnet wird.

[0129] Der Durchschnittsfachmann wird problemlos erkennen, dass die Architektur des vorliegenden Sys-

tems auf vielerlei Weise verteilt werden kann, wovon mehrere die Verfügbarkeit eines Mikroprozessors erfordern, sei es in einem Hostrechner, mit dem das System verbunden ist, oder an anderer Stelle zwischen der hierin beschriebenen integrierten Schaltung und einem Hostrechner. Es werden Ausführungen erwogen, bei denen die kumulierten Zähler- und Nennerwerte, die die Additionsterme darstellen, einem solchen Mikroprozessor zusammen mit den O_N - und O_D -Offsetwerten zur Verarbeitung zugeführt werden oder bei denen die gesamte Verarbeitung durch einen auf dem Gebiet bekannten programmierten Mikroprozessor verwirklicht wird.

[0130] Zunächst werden die Zähler- und Nenner-Akkumulatoren **150**, **152**, **154** und **156** während des Einschaltens des Systems auf Null gesetzt. Wenn die multiplex betriebenen A/D-Wandler wie in [Fig. 7](#) gezeigt verwendet werden, werden die digitalisierten Spannungsdaten in dem ersten Wort des Registers **138-1** (die die Spannung am Ausgang der Abtast-Halteschaltung **50-1** darstellen) zu der Summe im Akkumulator addiert und das Ergebnis wird in dem Akkumulator gespeichert. Nacheinander werden die in dem ersten Wort der Register **138-2** bis **138-6** gespeicherten digitalisierten Spannungswerte (die die Spannung an den Ausgängen der Abtast-Halteschaltungen **50-5**, **50-9**, **50-13**, **50-17** bzw. **50-21** darstellen) zu den Summen in den Akkumulatoren addiert und die Ergebnisse werden in den Akkumulatoren gespeichert. Wie bereits erwähnt können die A/D-Wandler **51-2** bis **52-6** zu diesem Zeitpunkt die an den Ausgängen der Abtast-Halteschaltungen **50-2**, **50-6**, **50-10**, **50-14**, **50-18** und **50-22** anliegenden Spannungen umwandeln und die digitalisierten Werte in den zweiten Wörtern der Register **138-1** bis jeweils **138-6** speichern.

[0131] Als Nächstes werden nacheinander die in den zweiten Worten der Register **138-1** bis **138-6** gespeicherten digitalisierten Spannungswerte (die die Spannung an den Ausgängen der Abtast-Halteschaltungen **50-2**, **50-6**, **50-10**, **50-14**, **50-18** bzw. **50-22** darstellen) zu der Summe in dem Akkumulator addiert und das Ergebnis wird in dem Akkumulator gespeichert.

[0132] Als Nächstes werden nacheinander die in den ersten Wörtern der Register **138-1** bis **138-6** gespeicherten digitalisierten Spannungswerte (die die Spannung an den Ausgängen der Abtast-Halteschaltungen **50-3**, **50-7**, **50-11**, **50-15**, **50-19** bzw. **50-23** darstellen) zu der Summe in dem Akkumulator addiert und das Ergebnis wird in dem Akkumulator gespeichert, gefolgt von den in den zweiten Wörtern der Register **138-1** bis **138-6** gespeicherten digitalisierten Spannungswerten (die die Spannung an den Ausgängen der Abtast-Halteschaltungen **50-4**, **50-8**, **50-12**, **50-16**, **50-20** bzw. **50-24** darstellen).

[0133] Zu diesem Zeitpunkt halten die Akkumulatoren die Summen aller einzelnen digitalisierten Spannungswerte. Die in den O_N - und O_D -Offsetregistern **158**, **160**, **162** und **164** gespeicherten digitalen Werte werden jetzt jeweils von den in den Zähler- und Nenner-Akkumulatoren gespeicherten Werten subtrahiert. Die Divisionsoperation in den Teilern **174** und **176** beendet dann die Berechnung des gewichteten Mittels.

[0134] Die Divisionsoperation kann auch durch einen externen Mikroprozessor vorgenommen werden, der die in den Akkumulatoren gespeicherten Werte holen oder die Akkumulationen selbst vornehmen kann. Da die O_N - und O_D -Offsetwerte derzeit von einem externen Mikroprozessor abgeleitet werden, ist der zusätzliche Verarbeitungsaufwand, der einem solchen externen Mikroprozessor durch diese Divisionsoperation auferlegt wird, minimal. Alternativ kann ein eigens vorgesehener Mikroprozessor auf dem Chip enthalten sein, um diese Verarbeitungsaufgaben handzuhaben, ohne von der hierin offenbarten Erfindung abzuweichen.

[0135] Die obige offenbarte Verarbeitung erfolgt innerhalb von etwa 1 Millisekunde und kann wiederholt ausgeführt werden. Die derzeitigen Mausforderungen aktualisieren die Positionsangaben **40** mal pro Sekunde, und dadurch kann die erfindungsgemäße Vorrichtung bei dieser Wiederholungsgeschwindigkeit problemlos betrieben werden.

[0136] Aufgrund der Art des verwendeten Verfahrens besteht die Möglichkeit, zusätzliche Störsicherheit zu bieten, ohne dass zusätzliche Hardware in dem erfindungsgemäßen System erforderlich wird. Zwar versteht sich, dass nach Ausführen der oben offenbarten Sequenz die Akkumulatoren gelöscht und der Prozess wiederholt werden kann, doch dürfen die Werte ggf. auch in den Akkumulatoren bleiben. Wenn dies geschieht, kann eine Mittlungsfunktion implementiert werden, um Rauschen weiter auszufiltern. Eine Reihe von Abtastungen werden genommen und durch die Akkumulatoren laufen gelassen, um diese am Ende der Abtastungssequenz zu löschen. Wie derzeit bevorzugt wird, werden **25** Abtastungen verarbeitet, bevor ein einziges Divisionsergebnis zur Verwendung durch das System genommen wird, wodurch die Wirkungen transientser Störungsspitzen im System stark reduziert werden. Der Durchschnittsfachmann wird erkennen, dass es sich bei Anzahl der vor dem Löschen der Akkumulatoren genommener Abtastungen um eine Gestaltungsentscheidung handelt, die von Faktoren wie Datenerfassungsgeschwindigkeiten, Datenverarbeitungsgeschwindigkeiten etc. bestimmt wird.

[0137] Es ist bevorzugt, ein zusätzliches Filtern der von den Divisionsblöcken **174** und **176** der Recheneinheit **16** von **Fig. 8** erzeugten X- und Y-Positionsdaten vorzusehen. Das Filtern erfolgt bevorzugt zwischen der Recheneinheit **16** und den Bewegungs- und Gesteneinheiten **18** und **20** von **Fig. 1**. Die X- und Y-Koordinaten werden als unabhängige Zahlen separat gefiltert. Jeder Filter ist ein mittelndes Register, das ein „gleitendes Mittel“ berechnet, wie auf dem Gebiet bekannt ist. Wenn das Vorhandensein eines Fingers erstmals detektiert wird, wird das Filterregister mit dem aktuellen Quotienten initialisiert. Bei den folgenden Abtastungen wird der neue Quotient mit dem Filterregisterwert gemittelt, um einen neuen Filterregisterwert zu erzeugen. Bei der derzeit bevorzugten Ausführung werden die Werte gleichmäßig im Mittel gewichtet, wenngleich verschiedene Gewichtungen verwendet werden können, um eine stärkere oder schwächere Filterung vorzusehen. Die Folge von Werten im Filterregister dienen als X- und Y-Koordinaten, die von den Bewegungs- und Gesteneinheiten **18** und **20** von **Fig. 1** verwendet werden.

[0138] Das System ist an sich ändernde Bedingungen, zum Beispiel eine Bauteilalterung, eine sich ändernde Kapazität aufgrund von Feuchtigkeit und Verunreinigung der Berührungsfläche etc. anpassbar. Ferner minimiert es wirksam Umgebungsrauschen. Diese Wirkungen werden in dreifacher Weise berücksichtigt. Zuerst werden die Offset-Werte O_N und O_D dynamisch aktualisiert, um sich ändernde Bedingungen zu berücksichtigen. Zweitens wird eine Servo-Rückkopplungsschaltung vorgesehen, um die Vorspannung zu ermitteln, die zum Setzen der Vorspannung der Ladungsintegratorschaltungen **44-1** bis **44-n** genutzt wird. Drittens werden wie bereits hierin offenbart die Bezugsspannungspunkte für V_{max} und V_{min} der A/D-Wandler **52** ebenfalls dynamisch geändert, um die Signal-Rausch-Spanne zu erhöhen.

[0139] Unter Bezug nun auf **Fig. 9** wird ein Blockdiagramm einer Kalibrierungseinheit **150**, die mit der Recheneinheit von **Fig. 8** verwendet werden kann, vorgestellt. Die Kalibrierungseinheit **150** führt einen Algorithmus aus, um die Zähler- und Nenner-Offsetwerte festzulegen, indem sie versucht zu ermitteln, wann sich kein Finger oder anderes leitendes Objekt in der Nähe der Berührungssensormatrix **22** befindet.

[0140] Wie bereits offenbart stellen die O_N - und O_D -Offsetwerte die Grundlinienwerte der Matrixkapazitäten bei Nichtvorhandensein eines Objekts dar. Diese Werte werden auch aktualisiert, da Grundlinienwerte, die zu niedrig oder zu hoch sind, die Wirkung haben, die scheinbare Position des Objekts abhängig von dem Vorzeichen des Fehlers zu verschieben. Diese Werte werden durch Auswahl der abgelesenen Werte bei Nichtvorhandensein eines Objekts an der Sensormatrix **22** ermittelt. Da es keine externe Möglichkeit gibt, zu „wissen“, wann kein Objekt an der Sensormatrix **22** vorhanden ist, wird ein Algorithmus verwendet, um diese Offsetwerte zu ermitteln und dynamisch zu aktualisieren. Wenn die Kalibrierungseinheit einen Z-Wert sieht, der für die Z-Werte bei Nichtvorhandensein eines Fingers typisch zu sein scheint, weist sie die Offset-Register (**158**, **160**, **162** und **164** von **Fig. 8**) an, aus den aktuellen Werten der Akkumulatoren neu zu laden. Nach einer derzeit bevorzugten Ausführung beruht die Entscheidung zur Aktualisierung der Offsetwerte auf dem Verhalten der Sensormatrix **22** in nur einer von X- oder Y-Richtung, wenn aber die Entscheidung getroffen wird, werden alle vier Offsets (O_{NX} , O_{DX} , O_{NY} und O_{DY}) aktualisiert. In anderen Ausführungen der Erfindung kann die Entscheidung zur Aktualisierung für jede Richtung entsprechend den hierin festgelegten Kriterien einzeln getroffen werden.

[0141] Der Kalibrierungsalgorithmus arbeitet durch Überwachen von Änderungen bei einem gewählten der Nenner-Akkumulatorwerte. Erfindungsgemäß wurde festgestellt, dass die Empfindlichkeit gegenüber Kapazitätsänderungen eines der Sätze leitender Leitungen in der Sensormatrix **22** größer als die Empfindlichkeit gegenüber Kapazitätsänderungen des anderen der Sätze leitender Leitungen in der Sensormatrix **22** ist. Die Erfahrung legt nahe, dass der Satz leitender Leitungen mit der größeren Empfindlichkeit gegenüber Kapazitätsänderungen derjenige ist, der sich physikalisch über dem Satz leitender Leitungen in der anderen Richtung befindet und daher am nächsten zur Berührungsfläche der Sensormatrix **22** liegt. Der obere Satz leitender Leitungen neigt dazu, den unteren Satz leitender Leitungen von kapazitiven Änderungen, die über der Oberfläche der Sensormatrix **22** eintreten, teilweise abzuschirmen.

[0142] Der Fingerdruck wird durch Addieren der an den Erfassungsleitungen gemessenen Kapazitäten erhalten. Dieser Wert ist nach Subtrahieren des Offsets O_D bereits im Nenner-Akkumulator vorhanden. Ein Finger ist vorhanden, wenn der Druck einen geeigneten Schwellwert übersteigt. Diese Schwelle kann experimentell gewählt werden und ist eine Funktion von Oberflächenmaterial und Schaltkreiszeitsteuerung.

[0143] Der Schwellwert kann so eingestellt werden, dass er dem Geschmack des einzelnen Bedieners gerecht wird.

[0144] Der von der Vorrichtung gemeldete Druck ist eine einfache Funktion $f(X_D, Y_D)$ der Nenner der X- und

Y-Richtungen, wie in Block **178** von [Fig. 8](#) implementiert. Mögliche Funktionen umfassen das Wählen eines bevorzugten Nennerwerts oder das Addieren der Nenner. In einer derzeit bevorzugten Ausführung wird der kleinere der beiden Nenner gewählt. Diese Wahl hat die wünschenswerte Wirkung, dass ein Sinken des Drucks unter den Schwellwert verursacht wird, wenn sich der Finger leicht weg von dem Rand der Berührungsfläche bewegt, wo die X-Sensoren gültige Daten erzeugen, die Y-Sensoren aber nicht, oder umgekehrt. Dies dient als elektronische Einfassung, die an Stelle einer mechanischen Einfassung am Rand der Sensorfläche treten kann.

[0145] In dem Beispiel von [Fig. 8](#) wird der Y-Nenner für das Überwachen gewählt, weil er am empfindlichsten ist. Der gewählte Nenner wird für die Zwecke des Kalibrierungsalgorithmus mit Z bezeichnet. Der aktuelle, gespeicherte Offsetwert für diesen Nenner wird mit O_z bezeichnet.

[0146] Das Ziel des Kalibrierungsalgorithmus ist es, allmähliche Veränderungen des ruhenden Z-Werts zu verfolgen, dabei aber sicherzustellen, dass nicht auf den Finger kalibriert wird und auch auf keine sich aus dem Rauschen ergebenden plötzlichen Spitzen kalibriert wird. Wie für den Durchschnittsfachmann aus der folgenden Offenbarung ersichtlich wird, könnte der Kalibrierungsalgorithmus in digitaler oder analoger Hardware oder in Software implementiert werden. In einer aktuellen Ausführung, die von den Erfindern tatsächlich getestet wird, wird er in Software implementiert.

[0147] Wenn Z-Werte in der Kalibrierungseinheit eintreffen, werden sie durch Filter **182** geleitet. Ein Verlaufspuffer **184**, der in Verbindung mit dem Filter **182** arbeitet, hält ein „gleitendes Mittel“ der jüngsten Z-Werte. Wenn ein neuer Z-Wert eintrifft, wird das aktuelle gleitende Mittel F_z entsprechend folgender Formel aktualisiert:

$$\text{neu } F_z = \alpha(\text{alt } F_z) + (1-\alpha)Z \quad \text{[Gleichung 7]}$$

wobei α ein konstanter Faktor zwischen 0 und 1 und typischerweise nahe 1 ist und Z der aktuelle Z-Wert ist. In der bevorzugten Ausführung ist alpha in etwa 0,95. Beabsichtigt wird, dass sich F_z so langsam ändert, dass es allmählichen Veränderungen folgt, ohne durch kurze Störungen von Z stark beeinträchtigt zu werden.

[0148] Der Filter **182** empfängt ein Signal ENABLE von der Steuereinheit **186**. Das gleitende Mittel F_z wird anhand neuer Z-Werte nur aktualisiert, wenn ENABLE aktiviert ist. Ist ENABLE deaktiviert, bleibt F_z konstant und wird von dem aktuellen Z nicht beeinflusst.

[0149] Der Verlaufspuffer **184** hält die mehreren jüngsten Werte für F_z fest. In der vorliegenden Ausführung hält der Verlaufspuffer **184** die beiden vorhergehenden F_z -Werte fest. Der Verlaufspuffer **184** könnte als Schieberegister, Ring oder analoge Verzögerungsleitung implementiert werden. Wenn der Verlaufspuffer **184** ein Signal REWIND von der Steuereinheit **186** empfängt, stellt er das aktuelle gleitende Mittel F_z auf den ältesten gespeicherten Wert zurück. Das ist, als wäre der Filter **182** „rückwirkend“ für eine Zeitdauer entsprechend der Tiefe des Verlaufspuffers **184** deaktiviert. Der Zweck des Verlaufspuffers ist es, eine solche rückwirkende Deaktivierung zuzulassen.

[0150] Das aktuelle gleitende Mittel F_z wird von Absolutdifferenzeinheiten **188** und **190** sowie dem Komparator **192** mit dem aktuellen Z-Wert und dem aktuellen Offset O_z verglichen. Die Absolutdifferenzeinheit **188** subtrahiert die Werte Z und F_z und gibt den absoluten Wert ihrer Differenz aus. Die Absolutdifferenzeinheit **190** subtrahiert die Werte O_z und F_z und gibt den Absolutwert ihrer Differenz aus. Der Komparator **192** aktiviert das Signal UPDATE, wenn die Ausgabe der Absolutdifferenzeinheit **188** kleiner als die Ausgabe der Absolutdifferenzeinheit **190** ist, d.h. wenn F_z näher zu Z als zu O_z ist. Das Signal UPDATE neigt dazu, aktiv zu sein, wenn sich der mittlere Wert von Z zu einem neuen Ruhewert verschiebt. Es neigt dazu, nicht aktiviert zu werden, wenn Z eine kurze Abweichung weg von seinem normalen Ruhewert aufweist. Die Filterkonstante α ermittelt die Länge einer Abweichung, die für diesen Zweck als „kurz“ gilt.

[0151] Eine Subtrahiereinheit **194** ist ein einfacher Subtrahierer, der die mit Vorzeichen versehene Differenz zwischen Z und O_z berechnet. Dieser Subtrahierer ist eigentlich mit Subtrahierer **172** in [Fig. 8](#) redundant und kann daher bei der eigentlichen Implementierung mit diesem verschmolzen werden. Die Ausgabe C_z dieses Subtrahierers ist der kalibrierte Z-Wert, eine Schätzung des Fingerdrucks. Dieser Druckwert wird durch Komparatoren **196** und **198** mit einem positiven und negativen Z-Schwellwert verglichen. Diese Schwellwerte werden als Z_{TH} und $-Z_{TH}$ gezeigt, wenngleich sie nicht wirklich von gleicher Größenordnung sein müssen.

[0152] Wenn das Drucksignal C_z größer als Z_{TH} ist, wird das Signal FINGER aktiviert, was das mögliche Vor-

handensein eines Fingers anzeigt. Der von der Kalibrierungseinheit verwendete Schwellwert Z_{TH} ist ähnlich dem vom restlichen System zur Detektion des Vorhandenseins des Fingers verwendeten Schwellwert oder kann einen anderen Wert haben. In der vorliegenden Ausführung wird die Kalibrierung Z_{TH} etwas unter dem Haupt- Z_{TH} angesetzt, so dass die Kalibrierungseinheit bezüglich des Vorhandenseins eines Fingers eine konservative Wahl trifft.

[0153] Wenn das Drucksignal C_Z kleiner als $-Z_{TH}$ ist, wird das Signal FORCE aktiviert. Da O_Z gleich dem Ruhewert von Z bei Nichtvorhandensein eines Fingers sein soll und ein Finger nur die Sensorkapazität und somit den Wert von Z erhöhen kann, impliziert ein weitgehend negativer C_Z -Wert, dass sich die Vorrichtung inkorrekt auf einen Finger kalibriert haben muss, der gerade weggenommen wurde. Die Kalibrierungslogik **200** nutzt diese Tatsache, um eine Neukalibrierung zu erzwingen, da nun der Finger nicht mehr vorhanden ist.

[0154] Die Steuerlogik **186** ist dafür verantwortlich zu verhindern, dass ein gleitendes Mittel F_Z durch Z -Werte beeinflusst wird, die bei Vorhandensein eines Fingers eintreten. Die Ausgabe ENABLE ist im Allgemeinen aus, wenn das Signal FINGER richtig ist, und ein, wenn das Signal FINGER falsch ist. Wenn aber FINGER von falsch zu richtig wechselt, pulst die Steuereinheit auch das Signal REWIND. Wenn FINGER von richtig zu falsch wechselt, wartet die Steuereinheit kurze Zeit (vergleichbar mit der Tiefe des Verlaufspuffers), bevor ENABLE aktiviert wird. Dadurch wird verhindert, dass das gleitende Mittel Z folgt, sobald ein Finger vorhanden ist sowie kurze Zeit vor und nach dem Vorhandensein des Fingers.

[0155] Die Kalibrierungslogik **200** erzeugt ein Signal RECAL aus den Ausgaben der drei Komparatoren **192**, **196** und **198**. Wenn RECAL aktiviert wird, werden die Offsetregister O_N und O_D aus den aktuellen Akkumulatorwerten neu geladen. RECAL wird aus der folgenden Logikgleichung erzeugt:

RECAL=FORCE oder (UPDATE und nicht FINGER).

[Gleichung 8]

[0156] Ferner sieht die Kalibrierungslogik **200** vor, dass RECAL aktiviert wird, sobald das System erstmals initialisiert wird, möglicherweise nach einem kurzen Zeitraum, um das Stabilisieren der Ladungsintegratoren und anderer Schaltungen abzuwarten.

[0157] Aus den Beschreibungen der Steuerlogik **186** und der Kalibrierungslogik **200** geht für den Durchschnittsfachmann hervor, dass diese Blöcke mühelos mit Hilfe herkömmlicher Logik im Rahmen einer einfachen und routinemäßigen Logikausgestaltung konfiguriert werden können.

[0158] Es sollte für jeden Durchschnittsfachmann offensichtlich sein, dass der beschriebene Kalibrierungsalgorithmus nicht spezifisch für das bestimmte System von Ladungsintegratoren und Akkumulatoren der vorliegenden Erfindung ist. Vielmehr könnte er in jedem Berührungssensor eingesetzt werden, der Nähe- oder Druckdaten erzeugt, wobei es erwünscht ist, einen Kalibrierungspunkt zu halten, der den Zustand des Sensors wiedergibt, wenn kein Finger oder Störerauschen vorhanden ist.

[0159] Unter Bezug nun auf [Fig. 10](#) wird in schematischer Diagrammform eine Vorspannung erzeugende Schaltung **46** gezeigt. Nach einer derzeit bevorzugten Ausführung sind alle Vorspannungstransistoren **108** ([Fig. 4b](#)) der Ladungsintegratorschaltungen **44-1** bis **44-n** mit ihren Gattern mit einer einzigen Vorspannungsquelle verbunden, wenngleich ein Durchschnittsfachmann erkennen wird, dass andere Anordnungen möglich sind. Es gibt eine Reihe von Möglichkeiten, wie die von den Ladungsintegratorschaltungen **44-1** bis **44-n** geforderte Vorspannung erzeugt werden kann.

[0160] Wie bei Prüfung von [Fig. 10](#) ersichtlich wird, ist die Vorspannung erzeugende Schaltung **46** ein übergedämpftes Servosystem. Eine Bezugsquelle, die die Stromquellenfunktion einer typischen der Ladungsintegratorschaltungen **44-1** bis **44-n** approximiert, umfasst einen Kondensator **202**, wobei eine seiner Platten geerdet ist. Die andere seiner Platten ist durch ein erstes Durchgangsgatter **204** mit der V_{DD} -Stromzufuhr und durch ein zweites Durchgangsgatter **208** mit einem Stromquellentransistor **206** verbunden. Eine Filterschaltung **210**, die identisch zu den Filterschaltungen **48-1** bis **48-n** ist und durch das gleiche Signal wie die Filterschaltungen **48-1** bis **48-n** gesteuert wird, ist angeschlossen, um die Spannung an dem Kondensator **202** in gleicher Weise abzutasten, wie die Filter- und Abtast-/Halteschaltungen **48-1** bis **48-n** die Spannungen an den Sensorleiterkapazitäten in der Sensormatrix **22** abtasten.

[0161] Die Ausgabe der Filterschaltung **210** wird dem nichtinvertierenden Eingang eines schwachen Steilheitsverstärkers **212** mit einem Vorspannungsstrom in dem Bereich von etwa 0,1–0,2 μA zugeführt. Der invertierende Eingang des Steilheitsverstärkers **212** ist mit einer festgelegten Spannung von etwa 1 Volt verbunden,

die zum Beispiel durch eine Diode **214** und einen Widerstand **216** erzeugt wird. Der Ausgang des Steilheitsverstärkers **212** wird von einem Kondensator **218** und auch von einem Kondensator **220** durch das Durchgangsgatter **222** nebengeschlossen. Der Kondensator **220** wird viel größer als der Kondensator **218** gewählt. In einer typischen Ausführung der vorliegenden Erfindung kann der Kondensator **218** etwa 0,2 pF und der Kondensator **220** etwa 10 pF haben.

[0162] Der Kondensator **220** ist mit dem Gatter des N-Kanal-MOS-Transistors **224** verbunden, dessen Drain mit dem Drain und Gatter des P-Kanal-MOS-Transistors **226** verbunden ist und dessen Source mit dem Drain und Gatter des N-Kanal-MOS-Transistors **228** verbunden ist. Die Source des P-Kanal-MOS-Transistors **226** ist mit V_{DD} und die Source von N-Kanal-MOS-Transistor **228** ist mit der Erde verbunden. Die gemeinsame Drain-Verbindung der Transistoren **224** und **228** ist der Vorspannungsausgangsknoten.

[0163] Ein optionales Durchgangsgatter **230** kann zwischen einer festgelegten Spannungsquelle (z.B. etwa 2 Volt) und dem Kondensator **220** angeschlossen sein. Das Durchgangsgatter **230** kann zum Initialisieren der Vorspannung erzeugenden Schaltung **46** bei Einschalten durch Laden des Kondensators **220** auf die festgelegte Spannung verwendet werden.

[0164] Die Filterschaltung **210** nimmt während jedes Abtastzeitraums eine neue Abtastung vor. Wenn sich die neue Abtastung von der vorherigen Abtastung unterscheidet, ändert sich die Ausgangsspannung des Steilheitsverstärkers **212** und beginnt, den Kondensator **218** auf eine neue Spannung zu laden bzw. zu entladen. Das Durchgangsgatter **222** wird für kurze Zeit (d.h. etwa 1 μ s) eingeschaltet und die Spannungen an den Kondensatoren **218** und **220** versuchen, sich selbst zu mitteln. Aufgrund der starken Größendifferenz zwischen den Kondensatoren **218** und **220** kann der Kondensator **218** nicht genügend Ladung liefern, um die Spannung während des Zeitraums auszugleichen, in dem das Durchgangsgatter **222** offen ist. Diese Anordnung verhindert große Änderungen der Vorspannung von Zyklus zu Zyklus.

[0165] Der Kondensator **202** sollte weitgehendst wie eine der Sensormatrixleitungen aussehen und hat einen Wert gleich der Hintergrundkapazität einer typischen Sensorleitung (d.h. ohne nahes Objekt oder vorhandene Kapazitätskomponente). Der Kondensator **202** kann in verschiedenerelei Weise gebildet werden. Der Kondensator **202** kann eine extra Sensorleitung in einem Teil der Sensormatrix umfassen, die so ausgelegt ist, dass sie einer der aktiven Sensorleitungen nahe kommt, aber durch eine Erdungsebene etc. von Fingerkapazität abgeschirmt ist. Alternativ kann der Kondensator **202** ein in der integrierten Schaltung ausgebildeter oder damit verbundener Kondensator sein und einen so gewählten Wert haben, dass er einer typischen Sensorleitung entspricht. Diesbezüglich ähnelt die die den Kondensator **202** und die Filterschaltung **210** umfassende Signalquelle ein wenig der Schaltung für das Erzeugen der V_{max} - und V_{min} -Bezugsspannungen, da sie eine typische Sensorleitung nachahmt.

[0166] Als andere Alternative kann eine der tatsächlichen Sensorleitungen verwendet werden, um die Vorspannung zu setzen. Die gemessene Spannung an den beiden Endpunkt-Sensorleitungen kann verglichen werden, und diejenige mit dem niedrigsten Wert kann aufgrund der Theorie gewählt werden, dass, wenn sich ein Finger oder ein anderes Objekt nahe der Sensormatrix befindet, er nicht an Sensorleitungen vorhanden ist, die sich an den gegenüberliegenden Kanten der Matrix befinden.

[0167] Nach einer anderen Ausgestaltung kann ein „Randweiterbewegungs“-Merkmal implementiert werden, wenn der erfindungsgemäße Objektpositionssensor an Stelle einer Maus als Computercursor-Steuergerät verwendet wird. Bei der Verwendung von Computermäusen oder anderen Cursor-Steuergeräten ergibt sich ein praktisches Problem, wenn versucht wird, ein Objekt über eine lange Strecke auf einem Computerbildschirm zu bewegen. Auf dieses Problem trifft man, wenn eine kleine Mausmatte mit einer Computermaus verwendet wird oder wenn ein Objektpositionssensor der hierin beschriebenen Art eine kleine Berührungssensorfläche hat.

[0168] Bei Berührungssensorapplikationen ist dieses Problem insbesondere während einer „Zieh“-Bewegung akut. Wenn der Bediener den Finger hebt, um einen zweiten Strich zu beginnen, endet die Ziehwirkung auf dem Bildschirm vorzeitig. Das Randweiterbewegungsmerkmal der vorliegenden Erfindung trägt dazu bei, die Notwendigkeit der Verwendung von „Ruder“- oder Mehrfachstreichen des Fingers zur Bewegung über eine große Strecke auf dem Bildschirm zu umgehen.

[0169] Eine vorbekannte Lösung für das Problem des Ziehens über eine lange Strecke besteht darin, ein Beschleunigungsmerkmal vorzusehen, d.h. eine "ballistische" Kurve, bei der die Verstärkung als Funktion der Fingergeschwindigkeit variiert, was dem Bediener das – wenn auch unbeholfene – Bewegen über lange Strecken

mit Hilfe wiederholter Fingerwischbewegung erlaubt. Dieses Verfahren kann mit jeder Zeigevorrichtung variabler Geschwindigkeit verwendet werden, zum Beispiel mit einer Maus auf einer Mausmatte beschränkter Größe. Typische Maustreibersoftware umfasst ein einstellbares Beschleunigungsmerkmal (manchmal unter einer irreführenden Bezeichnung wie „Mausgeschwindigkeit“).

[0170] Nach einer derzeit bevorzugten Ausführung wird das Randweiterbewegungsmerkmal des Objektpositionssensors durch die Bewegungseinheit **18** von [Fig. 1](#) implementiert und funktioniert durch Ausbilden von zwei Zonen in der Erfassungsebene **10**, die die Berührungssensormatrix **22** enthält. Wie in [Fig. 11](#) gezeigt wird, ist die Erfassungsebene **10** vorzugsweise in eine innere Zone **240** mit dem größten Teil des mittleren Teils der Oberfläche der Erfassungsebene **10** und in eine äußere Zone **242**, die typischerweise eine schmale Randfläche am Rand der Sensormatrix enthält, unterteilt. Die Mitte der Erfassungsebene **10** kann in einem kartesischen Koordinatensystem als Ursprung ($X_{\text{Mitte}}, Y_{\text{Mitte}}$) beschrieben werden. Der Durchschnittsfachmann wird erkennen, dass die innere und äußere Zone aber eine beliebige Form haben könnten.

[0171] Somit wird in [Fig. 11](#) die innere Zone **240** durch die obere Strichlinie Y_0 , die rechte Strichlinie X_0 , die untere Strichlinie $-Y_0$ und die linke Strichlinie $-X_0$ gebildet. Die äußere Zone **242** ist der Bereich zwischen den äußeren Kanten der Erfassungsebene **10**, die durch $Y_{\text{max}}, -Y_{\text{max}}, X_{\text{max}}$ und $-X_{\text{max}}$ gebildet werden, und den Außengrenzen der inneren Zone **240**, die durch $Y_0, X_0, -Y_0$ und $-X_0$ gebildet werden.

[0172] Nach dieser Ausgestaltung werden Fingerbewegungen in der inneren Zone **240** in üblicher Weise in Bewegungsereignisse umgewandelt, die an den Hostrechner zu senden sind. Wie auf dem Gebiet bekannt ist, kann die übliche Weise der Mitteilung einer Mausbewegung an einen Hostrechner auch zur Mitteilung einer Fingerbewegung an einen Hostrechner eingesetzt werden. Nachdem die Fingerposition wie hierin offenbart bestimmt wurde, lautet die dem Hostrechner mitgeteilte Information:

$$\Delta X = A(X_{\text{aktuell}} - X_{\text{alt}}) \quad \text{[Gleichung 9]}$$

$$\Delta Y = A(Y_{\text{aktuell}} - Y_{\text{alt}}) \quad \text{[Gleichung 10]}$$

wobei ΔX die Änderung der X-Position des Fingers ist, ΔY die Änderung der Y-Position des Fingers ist, X_{aktuell} die aktuelle X-Position des Fingers ist und X_{alt} die zuletzt gemeldete X-Position des Fingers ist, Y_{aktuell} die aktuelle Y-Position des Fingers und Y_{alt} die zuletzt gemeldete Y-Position des Fingers ist und A ein "Verstärkungsfaktor" ist, dem man häufig bei Mauscursor-Steuerapplikationen begegnet.

[0173] Typischerweise nimmt der Hostrechner ($\Delta X, \Delta Y$) Ereignisse und bewegt den Cursor um den angezeigten Betrag in jeder Achse, wodurch die Fingerposition auf dem Bildschirm rekonstruiert wird, wenn die aufeinanderfolgenden ΔX - und ΔY -Werte kumuliert werden. Insofern ist das ein übliches Cursorsterverhalten, bei dem die Randweiterbewegung nicht berücksichtigt wird.

[0174] Wenn der Finger in der äußeren Zone **242** gemeldet wird, kann das Randweiterbewegungsmerkmal aktiviert werden. Die Ermittlung, ob sich der Finger in der äußeren Zone befindet, ist eine einfache Ermittlung:

$$[-X_0 < X_{\text{aktuell}} < X_0] \text{ ist FALSCH, ODER } [-Y_0 < Y_{\text{aktuell}} < Y_0] \text{ ist FALSCH} \quad \text{[Gleichung 11]}$$

[0175] Unter Bezug nun auf [Fig. 12](#) wird in schematischer Diagrammform eine Schaltung **244** für das Vornehmen einer Ermittlung gezeigt, ob sich ein Finger in der äußeren Zone **242** befindet. [Fig. 12](#) zeigt eine Hardware-Ausführung für das Ermitteln, ob ein Finger in der äußeren Zone **242** ist, doch wird der Durchschnittsfachmann mühelos erkennen, dass diese Ermittlung problemlos durch Vornehmen einer Reihe von gleichwertigen Softwareroutinen ausgeführt werden könnte. Solche Softwareroutinen gehen offensichtlich und direkt aus den hierin beschriebenen Funktionen hervor.

[0176] Die Schaltung **244** enthält digitale Komparatoren **246**, **248**, **250** und **252**, die durch herkömmliche Logik direkt implementiert werden können. Der Komparator **246** gibt ein Signal „Richtig“ aus, wenn die Größe X_{aktuell} an einem seiner Eingänge größer als die festgelegte Größe X_0 ist, die seinem anderen Eingang geboten wird. Der Komparator **248** gibt ein Signal „Richtig“ aus, wenn die Größe X_{aktuell} an einem seiner Eingänge kleiner als die festgelegte Größe $-X_0$ ist, die seinem anderen Eingang geboten wird. Der Komparator **250** gibt ein Signal „Richtig“ aus, wenn die Größe Y_{aktuell} an einem seiner Eingänge größer als die festgelegte Größe Y_0 ist, die seinem anderen Eingang geboten wird. Der Komparator **252** gibt ein Signal „Richtig“ aus, wenn die Größe Y_{aktuell} an einem seiner Eingänge kleiner als die festgelegte Größe $-Y_0$ ist, die seinem anderen Eingang geboten wird.

[0177] Die Ausgänge der Komparatoren **246**, **248**, **250** und **252** werden durch ein ODER-Gatter **254** miteinander ODER-verbunden. Wie für den Durchschnittsfachmann nachvollziehbar ist, ist die Signalausgabe FingerOuter des ODER-Gatters **254** nur richtig, wenn die notwendigen Bedingungen von Gleichung 11 erfüllt sind.

[0178] Es wird derzeit bevorzugt, dass der Randweiterbewegungsaspekt von einem Bediener selektiv aktiviert oder deaktiviert werden kann. Wenn das Randweiterbewegungsmerkmal aktiviert ist und der Finger in der äußeren Zone gemeldet wird, wie vorstehend erläutert wurde, wird den gemeldeten ($\Delta X, \Delta Y$) Ereignissen eine zweite Komponente hinzugefügt:

$$\Delta X = A(X_{\text{aktuell}} - X_{\text{alt}}) + S(X_{\text{aktuell}} - X_{\text{Mitte}}) \quad \text{[Gleichung 12]}$$

$$\Delta Y = A(Y_{\text{aktuell}} - Y_{\text{alt}}) + S(Y_{\text{aktuell}} - Y_{\text{Mitte}}) \quad \text{[Gleichung 13]}$$

wobei X_{Mitte} die X-Koordinate der Mitte der Matte ist, Y_{Mitte} die Y-Koordinate der Mitte der Matte ist und S ein Multiplikationsfaktor für die Geschwindigkeit ist. S sollte so gewählt werden, dass die Bewegung des Cursors bei einer komfortablen Geschwindigkeit auf dem Bildschirm erfolgt.

[0179] Wenn der Finger zum Beispiel ein gutes Stück nach rechts gehalten wird (so dass $X_{\text{aktuell}} > X_0$), dann neigt der Cursor dazu, bei einer konstanten Geschwindigkeit nach rechts zu "gleiten", die von dem multiplizierenden Geschwindigkeitsfaktor S in den Gleichungen 12 und 13 gesetzt wurde. Dieser Faktor kann nach dem persönlichen Geschmack eines Bediener eingestellt werden.

[0180] Wenn die Sensormatrix **22** verschiedene Maße in X und Y hat, ist es hilfreich, die Parameter des multiplizierenden Geschwindigkeitsfaktors S in X- und Y-Richtung so zu setzen, dass sie im gleichen Verhältnis differieren wie die Mattenmaße, so dass ein am linken oder rechten Rand der Sensormatrix gehaltener Finger die gleiche Cursorgeschwindigkeit erzeugt wie ein am oberen oder unteren Rand gehaltener Finger. In der derzeit bevorzugten Ausführung der Sensormatrix gibt es 24 X-Leiterbahnen und 18 Y-Leiterbahnen. Da X 4/3 breiter als Y ist (24 Leiterbahnen gegenüber 18 Leiterbahnen), wird daher der X multiplizierende Geschwindigkeitsfaktor S_x so festgelegt, dass er $\frac{3}{4}$ so groß wie der multiplizierende Geschwindigkeitsfaktor S_y ist.

[0181] Die Gleitgeschwindigkeit des Cursors während der Randweiterbewegung ist eindeutig eine direkte Funktion der Distanz des Fingers von der Mitte der Matte, und die Gleitrichtung ist gleich der Richtung des Fingers von der Mitte. Wenn die äußere Zone die bevorzugte „Randbreiten“-Form hat, wie sie in [Fig. 11](#) gezeigt wird, dann ist der Finger immer in etwa gleich weit weg von der Mitte, wenn die Randweiterbewegung aktiviert ist (innerhalb eines Faktors der Quadratwurzel von $2 = 1,41$, unter der Annahme einer quadratischen Matte). Dadurch ergibt sich die psychologische Wirkung, dass die Randweiterbewegung eine konstante Gleitgeschwindigkeit mit sich bringt, bei der die Richtung durch die Position um den Rand der Sensormatrix festgelegt wird.

[0182] Die Variation der Quadratwurzel von 2 kann durch Dividieren der Randweiterbewegungsterme in den Gleichungen (12 und 13) durch einen normalisierenden Faktor folgender Form aufgehoben werden:

$$\sqrt{(X_{\text{aktuell}} - X_{\text{Mitte}})^2 + (Y_{\text{aktuell}} - Y_{\text{Mitte}})^2} \quad \text{[Gleichung 14]}$$

doch ist dies ein rechenintensiver Schritt, der zur Lösung eines Problems eingesetzt wird, das für den durchschnittlichen Bediener kaum wahrnehmbar ist; daher kann er übergangen werden.

[0183] Das Randweiterbewegungsmerkmal kann verwirrend sein, wenn der Bediener es nicht erwartet. Da die Randweiterbewegung in Verbindung mit der Ziehgeste sehr hilfreich ist, wird derzeit bevorzugt, dass sie nur während eines Ziehens eintritt, d.h. nur wenn die Gestenlogik praktisch „die Maustaste gedrückt hält“. Die Ziehgeste und andere Gesten werden durch die Gesteneinheit **20** von [Fig. 1](#) implementiert.

[0184] Zu Zeiten, da die Randweiterbewegungsfunktion nicht erwünscht ist, „geht“ die äußere Zone **242** „weg“ (d.h. wird ignoriert) und die innere Zone **240** weitet sich effektiv so aus, dass sie die gesamte Erfassungsebene **10** bedeckt. Es hat sich gezeigt, dass dies in der Praxis weit weniger verwirrend ist, wahrscheinlich weil der Bediener sich des Cursorsteuergeräts während einer Ziehgeste wahrscheinlicher bewusster ist als während einfacher Cursorbewegungen.

[0185] Unter der Annahme der bevorzugten Zonengrenzform von [Fig. 11](#) kann der folgende Algorithmus verwendet werden, um das Randweiterbewegungsmerkmal zu implementieren:

```

IF NOT ( $-X_0 < X_{\text{aktuell}} < X_0$  UND  $-Y_0 < Y_{\text{aktuell}} < Y_0$ )
AND (optional) eine Ziehgeste abläuft, DANN
Let  $eX = S_x(X_{\text{aktuell}} - X_{\text{Mitte}})$ 
Let  $eY = S_y(Y_{\text{aktuell}} - Y_{\text{Mitte}})$ 
ELSE
Let  $eX = eY = 0$ .
END IF

```

[0186] Als Nächstes werden die Bewegungsterme dX und dY aus dem regulären Algorithmus berechnet:

d.h. Let $dX = A(X_{\text{aktuell}} - X_{\text{alt}})$
 Let $dY = A(Y_{\text{aktuell}} - Y_{\text{alt}})$

[0187] Schließlich wird das sich ergebende Paket ($\Delta X = dX + eX$, $\Delta Y = dY + eY$) zum Hostrechner übertragen. Der Durchschnittsfachmann wird erkennen, dass durch die obige Gleichung eine lineare Proportionalität beschrieben wird. „Proportionalität“, so wie sie hier verwendet wird, bedeutet, dass das erzeugte Signal eine monotone Funktion ist. Der Durchschnittsfachmann wird erkennen, dass andere monotone Funktionen, einschließlich aber nicht ausschließlich umgekehrte Proportionalität und nichtlineare Proportionalität wie logarithmische oder exponentielle Funktionen, eingesetzt werden könnten, ohne von den hier offenbarten Grundsätzen abzuweichen.

[0188] Eine Hardware-Implementierung dieses Algorithmus wird in [Fig. 13](#) in schematischer Diagrammform veranschaulicht. Die Schaltung **256** wird zwar nur in X-Richtung implementiert gezeigt, der Durchschnittsfachmann wird aber erkennen, dass eine identische Schaltung auch in Y-Richtung eingesetzt werden könnte. Dieser Fachmann wird auch unmittelbar die vollständige Gleichwertigkeit der Implementierung der Hardwarelösung von [Fig. 13](#) als Softwareroutine erkennen.

[0189] Die Randweiterbewegungsschaltung **256** umfasst eine Subtrahiererschaltung **258**, in der der vorherige Wert von X_{aktuell} , der in Verzögerung **260** gespeichert ist, von dem derzeitigen Wert von X_{aktuell} abgezogen wird. Die Ausgabe der Subtrahiererschaltung **258** wird dem Multiplizierer **262** vorgelegt, der das Ergebnis mit dem Verstärkungsfaktor „A“ multipliziert. Die Ausgabe des Multiplizierers **262** ist der Term dX .

[0190] Der Term X_{aktuell} wird ebenfalls der Subtrahiererschaltung **264** vorgelegt, in der der Wert von X_{Mitte} von dem derzeitigen Wert von X_{aktuell} abgezogen wird. Die Ausgabe der Subtrahiererschaltung **264** wird dem Multiplizierer **266** vorgelegt, der das Ergebnis mit dem Verstärkungsfaktor „S“ multipliziert.

[0191] Ein UND-Gatter **268** mit zwei Eingängen hat seine Eingangsterme, den Wert FingerOuter von der Schaltung von [Fig. 12](#) und den Wert MotionEnable, das ein ein- und ausschaltbares Aktivierungssignal für das Randweiterbewegungsmerkmal der vorliegenden Erfindung ist. Wenn sowohl FingerOuter als auch MotionEnable richtig sind, ist der Schalter **270** so konfiguriert, dass er die Ausgabe des Multiplizierers **266** zur Addiererschaltung **272** weiterleitet. Wenn entweder FingerOuter oder MotionEnable falsch ist, dann ist der Schalter **270** so konfiguriert, dass er den Wert Null an den Addierer **272** weiterleitet. Die Ausgabe des Schalters **270** ist der eX -Term. Die Ausgabe des Addierers **272** wird zum Hostrechner als ΔX weitergeleitet. Das Signal MotionEnable kann durch den Bediener gesteuert werden, z.B. durch ein Steuerfeld. Alternativ kann es durch die Gesteneinheit gesteuert werden, wie eingehender offenbart wird.

[0192] In einer alternativen Form kann der Term dX durch den Term eX ersetzt werden und analog für dY und eY , wenn sich der Finger in der „äußeren“ Zone befindet, statt die beiden Terme in dieser Zone zu addieren. Dies führt zu einer „reineren“ Randweiterbewegung, die für den Bediener schwieriger zu lenken ist. Bedientests haben gezeigt, dass sich die oben gezeigte $dX + eX$ -Form besser anfühlt und leichter zu bedienen ist.

[0193] Eine andere Alternative, die funktionell ist, aber sich als weniger wünschenswert herausgestellt hat, ist eine etwas breitere äußere Zone. Die Gleitgeschwindigkeit wird dann proportional zur Strecke des Fingers in die äußere Zone statt der Strecke von der Mitte der Matte verändert. Wenn der Finger in die äußere Zone vordringt, beginnt dadurch die Gleitgeschwindigkeit bei Null und nimmt bis zu einem vernünftigen Grenzwert zu, wenn der Finger den Rand der Matte erreicht. Das Ergebnis ist ein gleichmäßigerer Übergang zwischen der Randweiterbewegung und normalem Verhalten. Es ist nicht schwierig, die obigen Formel zu abzuändern, dass dieses alternative Verhalten erzeugt wird. Diese Variante wurde von den Erfindern erprobt, da der Übergang in den Modus der Randweiterbewegung zu abrupt erschien; Tests zeigten, dass diese Abruptheit aber bei typischer Verwendung sogar von Vorteil ist. Der gleichmäßige Übergang ist schwerer zu „erfühlen“ und führt letztlich dazu, dass er den Bediener stärker, nicht weniger überrascht. Der Durchschnittsfachmann wird erken-

nen, dass eine Lösung zwischen den beiden beschriebenen Verfahren ebenfalls eingesetzt werden kann, um einen weniger abrupten Übergang zu erzeugen.

[0194] Eine alternative Lösung für das Problem des Ziehens über lange Strecken besteht darin, ein „arretierendes“ Ziehen vorzusehen, wie es hierin offenbart wird.

[0195] Das Randweiterbewegungsmerkmal wird vorteilhaft mit einer oder mit mehreren Fingergesten eingesetzt, die von einem Bediener auf der Sensormatrix ausgeführt werden können und von dem System erkannt werden. Von besonderem Interesse sind die Grundtipp- und Ziehgesten. Die Tippgeste ist analog zum Klicken der Maustaste auf einer herkömmlichen Maus, und das Konzept des Ziehens von Objekten ist allen Mausbedienern bekannt.

[0196] Zeigegeräte wie Mäuse enthalten typischerweise eine oder mehrere Maustasten. Der Bediener kann Zeigen und auf eine Taste klicken, um ein Objekt auf dem Bildschirm auszuwählen, oder er kann die Taste gedrückt halten und die Maus bewegen, um ein Objekt über den Bildschirm zu bewegen. Zeigegeräte mit Berührungssensor können „Gesten“ anbieten, die spezielle Fingerbewegungen sind, die Maustastenaktionen ohne Notwendigkeit physikalischer Schalter simulieren. (Da Gesten für Anfänger oder behinderte Bediener schwierig sein können, ist es bevorzugt, auch physikalische Schalter vorzusehen.) In der folgenden Diskussion sollte das Wort „Finger“ wie vorstehend beschrieben so verstanden werden, dass es einen Eingabestift oder ein anderes leitendes Objekt umfasst.

[0197] Unter Rückbezug auf [Fig. 1](#) prüft nach einer anderen erfindungsgemäßen Ausgestaltung die Gesteneinheit **20** die (X,Y,Z) Daten, die von der Recheneinheit **16** erzeugt werden, um ein oder mehrere „virtuelle Maustasten“-Signale zu erzeugen, die zusammen mit den (ΔX , ΔY) Signalen zum Host geschickt werden sollen.

[0198] [Fig. 14](#) ist ein eingehenderes Blockdiagramm der Gesteneinheit **20** von [Fig. 1](#). Die Gesteneinheit **20** kann eine Vielzahl von Gesten unterstützen. Die Gesteneinheit **20** umfasst eine Antippeinheit **280**, eine Zickzackeinheit **282**, eine Schiebeeinheit **284** und eine Tastensteereinheit **286**.

[0199] Durch die Gesteneinheit **20** kann eine Reihe physikalischer Schalter unterstützt werden. In dem veranschaulichenden Beispiel von [Fig. 14](#) stammen zwei Eingängen A und B der Tastensteereinheit **286** von physikalischen Schaltern. Diese Schalter können an dem Touchpad-Modul selbst angebracht oder außen vorgesehen werden. Es kann eine beliebige Anzahl an Schaltern oder gar keiner vorgesehen werden. Die Eingänge A und B haben zwei Zustände: Logik „0“ und Logik „1“. Der Durchschnittsfachmann wird erkennen, dass an Stelle von mechanischen Schaltern die Schaltsignale durch spezielle Berührungssensoren implementiert werden könnten, die durch Ladungsintegratoren ähnlich den Einheiten **44** betrieben werden, die in Schwellwertkomparatoren einspeisen, um digitale Signale zu bilden.

[0200] Die Antippeinheit **280**, die Zickzackeinheit **282** und die Schiebeeinheit **284** prüfen die Folge der (X,Y,Z) Abtastungen, um nach verschiedenen Arten von Gesten zu suchen. Die Ausgänge all dieser Einheiten plus die Schaltsignale werden in der Tastensteereinheit **286** kombiniert, um die tatsächlichen Tastendrucksignale zu erzeugen, die dem Host geschickt werden. In dem hier offenbarten veranschaulichenden Beispiel simuliert das Touchpad ein Zeigegerät mit drei Tasten (links, Mitte, rechts). Das System von [Fig. 14](#) könnte offensichtlich erweitert werden, um andere Gesten als die hier beschriebenen zu unterstützen oder um im Interesse der Einfachheit weniger Gesten zu unterstützen.

[0201] Die Tastensteereinheit **286** kann ein beliebiges von mehreren bekannten Verfahren für das Kombinieren mehrerer Signale verwenden. Zum Beispiel kann unter den verschiedenen Quellen eine Prioritätseinstellung festgelegt werden oder jeder Tastenausgang (links, Mitte oder rechts) kann aktiviert („angeklickt“, „gedrückt“ oder „gedrückt gehalten“) werden, sobald eine der Quellen diese Taste anzeigt. Ein spezielles Verfahren für das Kombinieren dieser Signale ist ein Routinenauslegungsdetail, das von einer bestimmten Systemkonfiguration abhängig ist und problemlos von einem Durchschnittsfachmann implementiert werden kann.

[0202] In einer derzeit bevorzugten Ausführung überführt die Tastensteereinheit **286** sowohl Schalter als auch Gesten in die am häufigsten verwendeten virtuellen Tasten, was dem Bediener ein Höchstmaß an Flexibilität gibt. In einer anderen Ausführung können Schalter und Gesten in verschiedene virtuelle Tasten überführt werden, so dass ein größere Anzahl virtueller Tasten abgedeckt werden kann, ohne auf exotische Gesten zurückzugreifen. Oder dem Bediener kann eine Auswahl von Überführungen angeboten werden.

[0203] Auf dem Gebiet ist es bekannt, das Verarbeiten zusätzlicher Tastenschaltungen, beispielsweise Doppelklicken, Auswählen häufig verwendeter Menüpositionen, etc. an Stelle ihrer normalen Rolle als Maustasten zuzulassen. Analog könnte die Tastensteereinheit **286** oder die Hostsoftware einige der hier beschriebenen Gesten in Softwarebefehle überführen, statt Maustasten zu simulieren. Eine solche Verarbeitung und Überführung liegt eindeutig im Bereich des Könnens des Durchschnittsfachmanns.

[0204] Die Antippeinheit **280** decodiert die einfachsten Gesten, einschließlich Tippbewegungen, Ziehen, Hüpfen und Antippzonen. Diese Gesten werden in [Fig. 15a](#) bis [Fig. 15e](#) als Zeitverlaufdiagramme veranschaulicht. In jeder der [Fig. 15a](#) bis [Fig. 15e](#) werden zwei Signale gegen die Zeit aufgetragen gezeigt; eines ist das analoge „Z“-Signal (Fingerdruck), das andere ist das digitale „Aus“-Signal (virtuelles Tastendrücker). Die verschiedenen relevanten Zeitspannen werden mit den Bezeichnungen „t1“ bis „t21“ dargestellt.

[0205] Die einfache "Antipp"-Geste ist ein schnelles Tippen des Fingers auf das Feld. Ein solches Antippen kurzer Dauer, das wenig oder keine X- oder Y-Fingerbewegung während des Antippens erfordert, wird dem Host als kurzes Klicken der Maustaste präsentiert. Wenn eine Maus mit mehreren Tasten simuliert wird, kann die Antippgeste ein Klicken der „primären“ Maustaste simulieren, oder die zu simulierende Taste kann unter Verwendung einer Umschalttaste, eines Steuerfelds oder eines anderen bekannten Mittels vom Bediener wählbar sein. Zweifaches Antippen in schneller Folge wird dem Host als Doppelklicken der Taste vorgelegt. Im Allgemeinen wird mehrfaches Antippen in offensichtlicher und natürlicher Weise in mehrfaches Klicken übersetzt.

[0206] Da es nicht möglich ist zu sagen, ob ein Fingeranschlag ein gültiges Antippen ist (im Gegensatz zu einer Cursorbewegung), während der Finger noch unten ist, meldet das Gerät der derzeit bevorzugten Ausführung ein Tastenklicken solange nicht, bis der Finger gehoben wird. Diese Verzögerung ist im Allgemeinen für den Bediener nicht merklich, da Antippbewegungen laut Definition sehr kurze Anschläge sind.

[0207] Ein kleiner Betrag an Bewegung kann während der Antippbewegung aufgrund von Faktoren wie der natürlichen Verformung der Fingerspitze unter Druck eintreten. Dies kann bewirken, dass das durch die Antippgeste erzeugte virtuelle Klicken das falsche Element oder die falsche Position auf dem Bildschirm wählt. Um dies zu vermeiden, muss entweder die Bewegung unterdrückt werden, bis die Bewegung groß genug ist oder die Dauer lang genug ist, um ein Antippen zu disqualifizieren, oder die Bewegung muss zugelassen werden, aber dann rückwirkend aufgehoben werden, sobald die Antippgeste erkannt wird. Die letztere Lösung ist bevorzugt, da auch ein kleiner Betrag an unterdrückter Bewegung für den Bediener merklich ist.

[0208] Nach der derzeit bevorzugten Ausführung der Erfindung werden Bewegungsereignisse dem Host wie üblich übermittelt und auch in einem Register oder einer Warteschlange gespeichert. Wird die Antippgeste erkannt, wird ein entsprechender negativer Bewegungsbetrag schnell wiederholt, um die bereits gemeldete Bewegung „aufzuheben“ und die ursprüngliche Cursorposition zu dem Moment, da das Vorhandensein des Fingers erstmals detektiert wurde, wiederherzustellen. Die Bewegung während des Anschlags kann in Form einer Folge mehrerer Pakete an den Host gesendet worden sein. Für größere Präzision kann diese Sequenz gespeichert und umgekehrt wiederholt werden. Wenn die Bewegungsverarbeitung des Host aber linear ist, reicht es aus, den Gesamtbetrag der Bewegung während des Anschlags zu kumulieren und eine kompensierende Bewegung in einem einzelnen Paket zu senden. Da das „Beschleunigungs“-Merkmal eines typischen Maustreibers nur bei hohen Geschwindigkeiten aktiviert wird, ist diese Annahme der Linearität in diesem Zusammenhang meist unbedenklich.

[0209] Die von der Antippeinheit **280** berücksichtigten Eingaben sind CurPos, die aktuelle (X,Y) Fingerposition von der Recheneinheit; Z, der aktuelle Druckwert, und CurTime, die aktuelle Zeit in geeigneten Zeiteinheiten (wie Millisekunden oder der Anzahl an verarbeiteten Abtastungen).

[0210] Es gibt neun Zustandsvariable, die in der Antippeinheit **280** verwendet werden. TapState ist KEINE, wenn keine Geste abläuft. TIPPEN, wenn eine Tipp- oder Ziehgeste abläuft, und ARRETIERT, wenn ein arrelierendes Ziehen abläuft. TapOkay ist RICHTIG, wenn in dem aktuellen Anschlag ein ausreichend hoher Z-Wert gesehen wurde, damit sich der Anschlag als Antippen qualifizieren kann. DownPos ist die (X,Y) Position, bei der der Finger zuletzt das Feld berührt hat. DownTime ist die Zeit, bei der der Finger zuletzt berührt hat. UpPos und UpTime halten die Position und Zeit fest, bei der der Finger sich das letzte Mal von dem Feld hob. TapButton ist eines von LINKS, MITTE oder RECHTS, wobei identifiziert wird, ob die aktuelle Geste eine Aktion an der linken, mittleren bzw. rechten virtuellen Maustaste simuliert. Suppress ist RICHTIG, wenn die virtuellen Tasten für ein Doppelklicken unterdrückt werden. Out stellt schließlich die Ausgabe der Antippeinheit dar und ist eines von KEINE, LINKS, MITTE oder RECHTS.

[0211] Zum Steuern der Antippenheit werden mehrere Parameter verwendet. TapTime ist die Höchstdauer eines Anschlags, damit er sich als Tippgeste qualifizieren kann. DragTime ist der maximale Intervall zwischen dem ersten Tippen und dem Zurückkehren des Fingers zur Bildung einer Ziehgeste. HopTime ist die maximale Anhebezeit vor einem Tippen zur Qualifizierung als Hüpfen. TapRadius ist der maximale Bewegungsbetrag, der während eines Tippens eintreten kann. DragRadius ist die maximale Strecke zwischen dem ersten Tippen und dem Zurückführen des Fingers für ein Ziehen. HopDistance ist die zu bewegendende Mindeststrecke zur Qualifizierung für ein Hüpfen. Zthresh ist der Mindestdruck (Z) für das Detektieren eines Fingers. In den Ansprüchen dieser Anmeldung nehmen Schritte, die das „Detektieren des Vorhandenseins“ eines Fingers oder anderen Objekts (oder andere Worte dieses Sinngelalts) vortragen, an, dass ein Druck größer als Zthresh detektiert wurde. Schließlich ist Ztap der Mindestdruck (Z) für das Detektieren eines tippenden Fingers.

[0212] [Fig. 15a](#) zeigt den Zeitverlauf einer grundlegenden Tippgeste. Zuerst wird ein erfolgreiches Tippen gezeigt, gefolgt von einem Fingeranschlag, der zu lang ist, um sich als Tippen zu qualifizieren. Bei dem ersten Anschlag ist der Finger während der Zeit „t1“ unten, die kürzer als TapTime ist. Ferner ist (nicht in [Fig. 15a](#) dargestellt) die (X,Y) Bewegung während der Zeit „t1“ kürzer als TapRadius. Schließlich übersteigt das Z-Signal die Schwelle Ztap mindestens einen Teil des Anschlags lang. Dadurch qualifiziert sich der Anschlag als Tippen. Das Signal Out (die untere Leiterbahn von [Fig. 15a](#)) wird für einen bestimmten Zeitbetrag „t2“ richtig, dann falsch. Wie später besprochen wird, ist der Zeitbetrag „t2“ gleich der DragTime. Bei dem in den folgenden Flussdiagrammen beschriebenen Gerät ist die Variable TapState über den gesamten Intervall „t2“ gleich TIP-PEN. Wie derzeit bevorzugt wird, beträgt TapTime etwa 400 ms, TapRadius ist etwa 2% der Breite des Sensorfelds und Ztap ist etwas größer als Tthresh, dessen Wert durch den Bediener einstellbar ist.

[0213] In der rechten Hälfte von [Fig. 15a](#) wird der Finger länger als der in der Figur als „t3“ gezeigte Parameter TapTime niedergehalten. Daher qualifiziert er sich nicht als Tippgeste und aus diesem Anschlag wird kein Out Signal erzeugt.

[0214] Bei der grundlegenden Ziehgeste tippt der Bediener einmal, bringt den Finger schnell zurück in Kontakt mit dem Feld, bewegt den Finger dann in eine erwünschte Richtung in der XY-Ebene des Felds. Die simulierte Maustaste wird zu Beginn der Ziehgeste gedrückt und wird nur freigegeben, wenn der Finger wieder vom Feld gehoben wird. Die Gestenlogik veranlasst, dass das erste Tippen der Ziehgeste in das verlängerte Ziehen einfließt, statt dem Host ein weiteres einzelnes Tastenklicken vorzulegen.

[0215] Bei einer Abänderung der Ziehgeste beginnt die oben beschriebene Geste ein Ziehen, das sich fortsetzt, selbst wenn der Finger angehoben wird. Das Ziehen endet (d.h. die simulierte Maustaste wird freigegeben), wenn der Finger wieder auf das Sensorfeld tippt. Dieses Merkmal ist als „arretierendes Ziehen“ bekannt. Das arretierende Ziehen lässt ein Ziehen über längere Strecken zu, als durch eine einzelne Fingerbewegung auf einem kleinen Feld durchgeführt werden kann, kann aber höchst verwirrend sein, wenn es versehentlich aktiviert wird. Das arretierende Ziehen wird ein verborgener Modus, ein bekanntes unerwünschtes Element in der Untersuchung von Bedienerchnittstellen. Daher wird es in der bevorzugten Ausführung dem Bediener als Option präsentiert, die in der Voreinstellung deaktiviert ist. Das zuvor beschriebene „Randweiterbewegungs“-Merkmal dient als alternative Möglichkeit zur Verwirklichung von Ziehen über lange Strecken.

[0216] Die Ziehgeste wird wie folgt implementiert. Wird ein Tippen erkannt, wird die virtuelle Maustaste wie zuvor beschrieben niedergedrückt. Die virtuelle Maustaste wird aber nicht freigegeben, bis der Finger ausreichend lang weg vom Feld geblieben ist, um sich als Ziehgeste zu qualifizieren. Dieser Zeitbetrag DragTime sollte lang genug gewählt werden, um eine bequeme Ziehgeste zuzulassen, aber kurz genug, damit das sich aus einer Tippgeste ergebende Klicken immer noch angemessen kurz ist. Wie derzeit bevorzugt wird eine Zeit von etwa 200 ms verwendet.

[0217] Wie in [Fig. 15b](#) gezeigt wird, beginnt die Ziehgeste mit einem oben beschriebenen Tippen von einer Dauer „t4“, die kürzer als TapTime ist. Das Signal Out wird als Reaktion auf dieses Tippen hoch. Der Finger bleibt dem Feld über eine Zeitdauer „t5“ fern, die kürzer als DragTime ist, kehrt dann zum Feld zurück und bleibt dort eine Zeit „t6“ lang, die länger als TapTime ist. Dies qualifiziert die Geste als Ziehen. Das Signal Out bleibt hoch, bis der Finger schließlich bei Zeit „t7“ gelöst wird. Bei der Implementierung von [Fig. 15b](#) ist die Zeit „t7“ zwischen dem Entfernen des Fingers und der Freigabe der virtuellen Maustaste null; bei anderen ähnlichen Implementierungen könnte dies ungleich null, aber klein sein, z.B. gleich DragTime. Zu beachten ist, dass TapState den gesamten Intervall von „t5“ bis zu „t7“ gleich TIP-PEN sein wird.

[0218] Es gibt eine Reihe von Alternativen, die für die Zeitsteuerung von DragTime in Betracht kommen können. [Fig. 15a](#) zeigt den Intervall „t2“, der auch der obere Grenzwert am Intervall „t6“ ist, als exakt gleich dem

Parameter DragTime, In einer Alternative wird DragTime im Verhältnis zu DownTime anstelle zu UpTime gemessen, was mit der Aussage gleichwertig ist, dass die Intervalle "t1" und "t2" ("t5" bzw. "t6") addiert DragTime ergeben müssen. Eine Folge dieses Verfahrens ist, dass bei der grundlegenden Tippgeste ein längeres, langsames Antippen ein kürzeres virtuelles Tastenklicken bewirkt. Dieser Widerspruch macht dieses Vorgehen für den Bediener weniger zufrieden stellend als das in [Fig. 15a-b](#) gezeigte Vorgehen.

[0219] In einer anderen Alternative wird DragTime proportional zur Länge des Intervalls „t1“ (bzw. „t5“) gemacht, so dass ein kurzes Tippen ein kurzes virtuelles Tastenklicken erzeugt und ein längeres Tippen (bis zum Grenzwert TapTime) ein längeres Klicken erzeugt. Diese Alternative gibt dem Bediener mehr Kontrolle über das simulierte Tastenklicken, macht aber das Verhalten von einem vom Bediener als unwesentlich empfundenen Merkmal der Tippaktion abhängig.

[0220] [Fig. 15c](#) zeigt die arretierende Ziehgeste. Das arretierende Ziehen beginnt mit einer standardmäßigen Ziehgeste, die die Intervalle „t8,“ bis „t10“ erfordert. Wenn aber der Finger angehoben wird, bleibt das Signal Out hoch. (In den Flussdiagrammen ändert sich TapState zu diesem Zeitpunkt von TIPPEN zu ARRETIERT). Die Figur zeigt eine zweite Ziehbewegung einer Länge, die länger als TapTime ist (gezeigt als "t11"), was das arretierende Ziehen nicht beendet, gefolgt von einer weiteren Bewegung der Länge „t12“, die kürzer als TapTime ist. Da sich diese letzte Bewegung als Tippen qualifiziert, beendet sie das arretierende Ziehen zum Zeitpunkt "t13". In den Flussdiagrammen ändert sich TapState zu diesem Zeitpunkt zurück zu TIPPEN; ein reguläres Tippen wird dann verarbeitet, das über eine Zeit „t13“ gleich DragTime wie gewöhnlich Out weiter hoch hält. Eine sinnvolle alternative Implementierung könnte das Ziehen nach einem anderen Intervall „t13“ beenden, beispielsweise Null.

[0221] [Fig. 15d](#) zeigt eine Doppeltippgeste. Das Doppeltippen ist anfangs nicht von einer Ziehgeste zu unterscheiden. Die zweite Bewegung „t16“ ist aber kürzer als TapTime, wodurch sie sich als zweites Tippen statt als Ziehen qualifiziert. Ein reguläres Antippverhalten bewirkt, dass Out einen weiteren Intervall „t18“ der Länge DragTime hoch bleibt; die in den Flussdiagrammen gezeigte spezielle Handhabung eines Doppeltippens unterdrückt aber die virtuelle Maustaste einen kurzen Zeitraum „t17“ lang nach Erkennung des Tippens. Dadurch nimmt der Hostrechner zwei einzelne Klicks wahr statt des einen langen, ineinander übergehenden Klickens, das er ohne diese spezielle Handhabung sehen würde.

[0222] Andere Gesten können zum Simulieren einer Mehrstasten-Maus verwendet werden. Bei einem solchen Vorgehen werden die Grundgesten durch eine „Hüpf“-Geste ergänzt, bei der der Finger von seiner Ruheposition an einer Stelle auf dem Feld gehoben wird und eine erhebliche Strecke weg von der Ruheposition auftippt. Wenn die Strecke groß genug ist (HopDistance typischerweise ein Teil der Breite des Sensorfelds; derzeit bevorzugt etwa 25%) und die Dauer zwischen dem Anheben und dem anschließenden Tippen unter einem geeigneten Schwellwert liegt (HopTime typischerweise ein Teil einer Sekunde; derzeit bevorzugt etwa 0,5 s), dann wird die durch das Tippen begonnene Klick- oder Ziehgeste auf einer anderen Maustaste simuliert. Diese andere Taste kann eine feste „sekundäre“ Taste sein oder sie kann durch ein Steuerfeld oder andere Mittel vom Bediener wählbar sein oder sie kann eine Funktion der Richtung sein, in der der Finger hüpfte (z.B. nach links statt nach rechts). Die Hüpfgeste ist als Option verfügbar, die in der Voreinstellung deaktiviert ist.

[0223] Zu beachten ist, dass zwar manche Bediener bei der Hüpfgeste bevorzugt mit einem zweiten Finger tippen, diese Geste aber nie mehr als einen Finger gleichzeitig auf dem Feld erfordert. Eine ähnliche Geste, das „Zickzack“, wird ebenfalls hier beschrieben und erfordert die gleichzeitige Verwendung von zwei Fingern.

[0224] [Fig. 15e](#) zeigt eine „Hüpf“-Geste. Diese Geste beginnt mit dem Finger bereits auf dem Feld. Der Finger wird dann einen Intervall „t19“ lang gehoben, der unter der HopTime liegt; dann kommt der Finger für ein reguläres Tippen „t20“ herunter. Ferner muss sich der Finger, was nicht in der Figur gezeigt wird, während des Intervalls „t19“ mindestens eine bestimmte Strecke HopDistance weg von seiner vorherigen Position bewegt haben. Wenn dies eintritt, wird die Geste als „Hüpfen“ statt als reguläres Tippen verarbeitet und das virtuelle Tastendrücken „t21“ erfolgt auf der rechten Taste Out(R) anstatt auf der üblichen linken Taste Out(L). Es ist unschwer ersichtlich, wie dem Tippen „t20“ weitere Fingeraktionen folgen könnten, um ein Ziehen oder ein Doppeltippen auf der rechten Taste zu bilden.

[0225] Eine andere Mehrstasten-Geste verwendet „Tippzonen“, in denen die Oberfläche des Felds in zwei oder mehr Zonen unterteilt ist. Ein in einer vorgegebenen Zone eingeleitetes Tippen oder Ziehen simuliert ein Ereignis auf einer Taste, die dieser Zone entspricht. Selbst wenn sich der Finger während eines Ziehens zwischen Zonen bewegt, wird das gesamte Ziehen auf der der Zone des ursprünglichen Tippens, das die Ziehgeste einleitete, entsprechenden Taste simuliert.

[0226] [Fig. 16a](#) und [Fig. 16b](#) veranschaulichen zwei Tippzonenformen. In [Fig. 16a](#) ist das Feld in drei vertikale Streifen **288**, **290** und **292** unterteilt, die der linken, der mittleren bzw. der rechten Maustaste entsprechen. In [Fig. 16b](#) ist das Feld in einen Hauptbereich **294** unterteilt, der die linke Maustaste simuliert, und in einen kleinen Eckbereich **296**, der die rechte Maustaste simuliert. Die Implementierung von [Fig. 16b](#) ist geeigneter, wenn eine Taste bei typischen Applikationen vermehrt eingesetzt wird als die andere Taste/die anderen Tasten.

[0227] Es ist bevorzugt, dass die Zonen eindeutig gekennzeichneten Regionen auf der Feldfläche entsprechen. Für den Fachmann ist offensichtlich, dass andere Zonenformen, zum Beispiel mehrere Ecken oder horizontale Streifen, gleichermaßen unkompliziert sind.

[0228] Zwischen Tippzonen und dem Randweiterbewegungsmerkmal gibt es eine Interaktion, die berücksichtigt werden muss. Insbesondere bei der Eckenzone von [Fig. 16b](#) regen die Tippzonen den Bediener an, nahe des Rands des Felds zu tippen. Wenn die Randweiterbewegung während der Tipp- und Ziehbewegungen oder ständig aktiv ist, dann neigt die Randweiterbewegung dazu, das richtige Verhalten des Eckentippens zu beeinträchtigen. Um dies zu verhindern, kann die Aktivierungslogik der Randweiterbewegung von [Fig. 13](#) leicht abgewandelt werden. Bei einer vorgegebenen Bewegung funktioniert die Randweiterbewegung nur, wenn der Finger mindestens einmal während dieser Bewegung in der inneren Zone war. Wenn der Finger in der äußeren Zone aufsetzt, wird die Randweiterbewegung daher nicht aktiviert, bis der Finger den Rand des Felds verlässt und dann zurückkehrt.

[0229] Alle oben beschriebenen Gesten sind Varianten der grundlegenden Tipp- und Ziehgesten. Bei dem hierin beschriebenen System werden alle diese Gesten von der Antippeinheit **280** erkannt. Der Betrieb der Antippeinheit **280** wird am einfachsten als Algorithmus in Form eines Flussdiagramms beschrieben. Anhand dieser Offenbarung wird ein Durchschnittsfachmann erkennen, dass die hier beschriebene Antippeinheit sogar als bekannte und offensichtliche Entsprechungen implementiert werden könnte, beispielsweise als Softwareprogramm, Hardwarezustandsmaschine oder anderes. Alle diese Implementierungen sollen in den Schutzzumfang der vorliegenden Erfindung fallen.

[0230] [Fig. 17a](#) bis [Fig. 17f](#) umfassen ein Flussdiagramm für den Betrieb der Antippeinheit **280**. Die Antippeinheit **280** implementiert die hierin beschriebenen Tipp-, Zieh-, arretierende Zieh-, Eckentipp- und Hüpfgesten. Bei den hierin beschriebenen Erkennungsoperationen wird das Eckentippen zur Simulation der rechten virtuellen Maustaste verwendet. Hüpfen nach links und rechts werden zum Simulieren der mittleren und der rechten virtuellen Maustaste verwendet. Einfaches Tippen simuliert die linke (primäre) virtuelle Maustaste.

[0231] Die Verarbeitung beginnt bei Schritt **300**, wenn jede (X,Y,Z) Abtastung von der Recheneinheit **16** von [Fig. 1](#) eintrifft. In einer derzeit bevorzugten Ausführung der Erfindung treffen diese Daten **40** mal pro Sekunde ein. Der Algorithmus von [Fig. 17a](#) bis [Fig. 17f](#) läuft jedes Mal, wenn eine Abtastung eintrifft, von Anfang (Schritt **300**) bis Ende (Schritt **376**) durch.

[0232] Schritt **302** ermittelt, ob der Finger oben oder unten ist, indem Z (Druck) mit Zthresh verglichen wird, um zu ermitteln ob ein Finger vorhanden („unten“) ist oder nicht („oben“). Anstelle eines einfachen Schwellwertvergleichs können zwei Schwellwerte verwendet werden, um eine Hysterese zu bieten, wie auf dem Gebiet bekannt ist. In [Fig. 17a](#) wird keine Hysterese gezeigt, eine ähnliche Hysterese wird aber später in [Fig. 20](#) für die „Schiebe“-Geste veranschaulicht.

[0233] Bei Schritt **304** weiss man, dass der Finger unten ist. Das vorherige Z wird geprüft, um zu sehen, ob der Finger zuvor unten war oder jetzt erst das Feld berührt.

[0234] Bei Schritt **306** wird ein Finger-Unten-Übergang detektiert. Bei der in [Fig. 15b](#) gezeigten Ziehgeste ist es von Vorteil zu prüfen, dass sich der Finger während der Zeit „t5“, dem Intervall zwischen dem ersten Tippen und der Rückführung des Fingers auf das Feld, keine große Strecke bewegt hat. Wenn Schritt **306** ermittelt, dass der Finger an einer anderen Stelle auf das Feld zurückgeführt wurde, dann war eine Ziehgeste wahrscheinlich nicht gewollt.

[0235] Daher berechnet Schritt **306** die Strecke zwischen der aktuellen Position CurPos (die gefilterten und geglätteten X- und Y-Positionsdaten) und der gespeicherten Position des vorherigen Tippens, DownPos. Wenn die Strecke unter einem Schwellwert DragRadius liegt und TapState TIPPEN ist (eine Ziehgeste steht an), dann geht der Ablauf weiter zu Schritt **308**. Ansonsten springt er zu Schritt **310**. Der Schwellwert DragRadius sollte ein Teil der Breite des Felds sein, vorgeweise größer (größzügiger) als der TapRadius, der bei der De-

tektion eines Grundtippens verwendet wird.

[0236] Der Durchschnittsfachmann wird erkennen, dass mehrere mögliche Distanzmaße zur Verwendung in Schritt **306** geeignet sind. Ein echtes Euklid-Distanzmaß ist plausibel, aber teuer zu berechnen; ein einfacheres Maß ist die Summe oder der Höchstwert der absoluten Werte der Distanzen in X und Y. Die Summe bzw. der Höchstwert erzeugt eine „Ziehzone“ um das ursprüngliche Tippen, die rauten- bzw. quadratförmig ist, an Stelle der von einem Euklid-Distanzmaß erzeugten kreisförmigen Zone. Experimente legen nahe, dass Bediener nicht den Unterschied zwischen diesen Zonenformen wahrnehmen können, daher ist das Maß bevorzugt, das am einfachsten zu berechnen ist. Ferner kann die Geometrie von Finger und Touchpad bewirken, dass die erhebliche Bewegung immer in einer Richtung liegt, z.B. X, in welchem Fall eine einfache absolute Differenz der X-Koordinaten bevorzugt sein kann.

[0237] In der bevorzugten Ausführung kann der Bediener den Grad der Gestenerkennung mit Hilfe eines Steuerfelds oder anderer Mittel ändern. Wenn der Bediener beschließt, nur Tippen, aber kein Ziehen zuzulassen, dann kann Schritt **306** programmiert werden, direkt zu Schritt **308** zu gehen, um auszuschließen, dass eine der Tippgesten eine Ziehgeste wird.

[0238] Bei Schritt **308** wurde eine Ziehgeste ausgeschlossen. TapState wird von TIPPEN zu KEINE geändert; die Wirkung ist eine einfache Tippgeste gefolgt von einer Cursorbewegung, ohne dass eine virtuelle Taste niedergedrückt wird.

[0239] Schritt **310** hält die Position und die Zeit fest, zu der der Finger auftraf.

[0240] Schritt **312** initialisiert die Flagge TapOkay auf FALSCH.

[0241] Schritt **314**, der bei allen Abtastungen, bei denen der Finger unten ist, abläuft, vergleicht Z mit dem Ztap-Schwellwert; Schritt **316** setzt TapOkay auf RICHTIG, wenn Z größer als der Ztap-Schwellwert ist. Wenn der Finger angehoben wird, ist daher TapOkay RICHTIG, wenn Z während der kurzen Bewegung, die ein Kandidat für eine Tippgeste ist, jemals den Tippschwellwert überstieg.

[0242] Unter Bezug nun auf [Fig. 17b](#) ist bei Schritt **318** bekannt, dass der Finger nicht auf dem Feld ist. Das vorherige Z wird geprüft, um zu sehen, ob der Finger zuvor oben war oder jetzt erst vom Feld genommen wird.

[0243] Bei Schritt **320** wird ein Finger-Oben-Übergang detektiert. Es werden verschiedene Tests bezüglich der letzten Bewegung (Zeitraum Finger unten) durchgeführt, um zu sehen, ob sie sich als Tippen qualifiziert. Um sich zu qualifizieren muss die Bewegung eine kurze Dauer (CurTime minus DownTime muss unter TapTime liegen), wenig oder keine Bewegung (die Strecke von CurPos zu DownPos muss kleiner als TapRadius sein) und genügend Spitzen-Fingerdruck (TapOkay muss RICHTIG sein) aufweisen, um sich als Tippen zu qualifizieren.

[0244] Bei Schritt **322** wird eine eingetretene Fingerbewegung rückwirkend durch schnelles Wiederholen eines entsprechenden negativen Bewegungsbetrags vom Register oder der Warteschlange zum Host aufgehoben, um die bereits gemeldete Bewegung „aufzuheben“ und die ursprüngliche Cursorposition zu dem Moment, da das Vorhandensein des Fingers erstmals detektiert wurde, wiederherzustellen. Wenn die Bewegung während des Anschlags dem Host in Form einer Folge mehrerer Pakete gesendet wurde, kann diese Sequenz gespeichert und rückwärts wiederholt werden. Wenn die Bewegungsverarbeitung des Host linear ist, reicht es aus, den Gesamtbetrag der Bewegung während des Anschlags zu kumulieren und eine kompensierende Bewegung in einem einzelnen Paket zu senden. Da das „Beschleunigungs“-Merkmal eines typischen Maustreibers nur bei hohen Geschwindigkeiten aktiviert wird, ist diese Annahme der Linearität in diesem Zusammenhang meist unbedenklich.

[0245] Schritt **324** führt eine von mehreren Aktionen beruhend auf dem aktuellen TapState aus. Wenn TapState KEINE ist (keine Gesten laufen ab), geht der Ablauf einfach zu Schritt **330** von [Fig. 17c](#). Wenn zweites TapState TIPPEN ist (ein kürzliches Tippen läuft noch ab), dann wird ein Doppeltippen detektiert. Schritt **326** setzt die Flagge Suppress auf RICHTIG, um das virtuelle Tastensignal für eine Abtastung auf niedrig zu setzen. Dies entspricht der Zeit „t17“ von [Fig. 15f](#). Bei einem alternativen Vorgehen können ein oder mehr zusätzliche Pakete, die eine Freigabe der virtuellen Tasten anzeigen, in den regulären Paketstrom eingefügt werden, statt eine Flagge Suppress zu verwenden, wie hierin gezeigt wird.

[0246] Wenn schließlich TapState ARRETIERT ist, ist das das Tippen, das ein arretierendes Ziehen beendet.

Schritt **328** setzt TapState zurück auf TIPPEN, springt dann direkt zu Schritt **358** von [Fig. 17e](#), wobei die Schritte übergangen werden, die entscheiden, welche der drei Maustasten simuliert werden sollen. Dadurch ändert sich das arretierende Ziehen wieder zu einem Tippen auf die gleiche virtuelle Maustaste. Nach der üblichen kurzen Dauer („t13“ von [Fig. 15c](#)) wird die virtuelle Taste freigegeben.

[**0247**] Es ist wesentlich, dass die Tastenwahl ([Fig. 17c](#)) in dem Fall ARRETIERT übergangen wird. Wenn ein arretierendes Ziehen mit der rechten Taste zum Beispiel durch ein Tippen auf die Ecke des Felds eingeleitet wird, dann sollte es möglich sein, das Ziehen durch Tippen auf eine beliebige Stelle des Felds, nicht nur auf die Ecke, zu beenden. Es ist auch wesentlich, dass die Tastenwahl in dem Fall des Doppeltippens enthalten ist. Andernfalls wäre es unmöglich, zum Beispiel linkes und rechtes Tastenklicken in schnellem Wechsel auszuführen, indem abwechselnd auf zwei verschiedene Stellen auf dem Feld getippt wird.

[**0248**] In einer anderen Ausführung wird, wenn TapState ARRETIERT ist, TapState auf KEINE gesetzt und Schritt **364** als Nächstes ausgeführt. Dadurch wird "t13" von [Fig. 15c](#) gleich Null.

[**0249**] Unter Bezug nun auf [Fig. 17c](#) prüft Schritt **330**, ob das sich aktuelle Tippen als „Hüpf“-Geste qualifiziert. Diese Prüfung umfasst mehrere Tests. Zuerst muss die Hüpfgeste vom Bediener aktiviert werden. Zweitens darf der Finger nur kurze Zeit zwischen dem aktuellen Tippen und dem letzten Zeitpunkt, da er sich auf dem Feld befand, angehoben gewesen sein (DownTime minus UpTime muss kleiner als HopTime sein). Schließlich muss die Position dieses Tippens beträchtlich weit weg von der vorherigen Position sein (die Strecke von DownPos zu UpPos muss größer als HopDistance sein). Wiederum sind eine Vielzahl von Distanzmaßen möglich. Die in [Fig. 17a](#) bis [Fig. 17f](#) gezeigten Operationen unterstützen linke und rechte Hüpfen; dadurch ist ein vernünftiges Distanzmaß eine absolute Differenz der X-Koordinate zwischen DownPos und UpPos.

[**0250**] In einer Variante, die leicht als praktisch gleichwertig zu erkennen ist, werden CurTime und CurPos an Stelle von DownTime und DownPos bei Schritt **330** verwendet.

[**0251**] Wenn sich das Tippen als Hüpfen qualifiziert, geht der Ablauf weiter zu Schritt **332**. Da das System dieses Beispiels zwei verschiedene Hüpfgesten unterstützt, wird die Hüpfrichtung geprüft, um die Art der Geste zu ermitteln. Wenn die X-Koordinate von DownPos (oder CurPos) kleiner als die X-Koordinate von UpPos ist, ist ein Hüpfen nach links eingetreten (unter der Annahme, dass X nach rechts zunimmt). Wenn die X-Koordinate von DownPos (oder CurPos) größer als die X-Koordinate von UpPos ist, ist ein Hüpfen nach rechts eingetreten. Zu beachten ist, dass aufgrund der Prüfungen des Schritts **330** DownPos an diesem Punkt entweder erheblich links oder erheblich rechts von UpPos ist.

[**0252**] Bei Schritt **334** bewirkt ein Hüpfen nach links, dass TapButton auf das Symbol MITTE gesetzt wird, so dass die Tippgeste ein virtuelles Klicken der mittleren Maustaste bewirkt.

[**0253**] Bei Schritt **336** bewirkt ein Hüpfen nach rechts, dass TapButton auf RECHTS gesetzt wird, was ein virtuelles Klicken der rechten Taste auslöst.

[**0254**] Schritt **338** läuft ab, wenn kein Hüpfen detektiert wurde. Er rückt vor, um auf die andere unterstützte alternative Geste, das Eckentippen, hin zu prüfen. Ein Eckentippen ist ein Tippen, das auf einer kleinen Eckenzone erfolgt, wie in [Fig. 16b](#) gezeigt wird. Ein Eckentippen tritt ein, wenn Eckentippen durch den Bediener aktiviert wurde; die X-Koordinate von DownPos (oder CurPos) ist größer als eine Koordinate CornerX; und die Y-Koordinate ist größer als eine Koordinate CornerY. CornerX und CornerY werden in [Fig. 16b](#) gezeigt.

[**0255**] Es sollte für den Durchschnittsfachmann offensichtlich sein, dass andere Tippzonen, wie die der [Fig. 16a](#), oder mehrere Eckenzonen auf völlig analoge Weise durch Prüfen der X- und Y-Koordinaten der angetippten Stelle entschlüsselt werden können.

[**0256**] In der derzeit bevorzugten Ausführung hat der Bediener die Wahl zwischen Hüpfgesten, Eckentippen oder keinem als Mechanismus für das Simulieren abwechselnden Tastenklickens. Nichts steht dem im Wege, dass eine Implementierung sowohl Hüpfen als auch Eckentippen auf einmal anbietet, außer dass dies für den Bediener wahrscheinlich eher verwirrend als vorteilhaft wäre.

[**0257**] Bei Schritt **340** wurde kein Eckentippen detektiert, daher wird TapButton auf LINKS gesetzt, um ein Klicken auf die linke virtuelle Maustaste zu simulieren.

[**0258**] Bei Schritt **342** wurde ein Eckentippen detektiert, daher wird TapButton auf RECHTS gesetzt, um ein

Klicken auf die rechte virtuelle Maustaste zu simulieren.

[0259] Schritt **344** hält die aktuelle Position als die neue UpPos fest, die Anhebe-Position, die für späteres Entschlüsseln von Hüpfen verwendet wird. Im Allgemeinen wird UpPos jedes Mal, wenn ein Anheben des Fingers vom Feld erkannt wird, aktualisiert. Es gibt aber zwei Ausnahmen zu dieser Regel. Erstens: wenn das Anheben des Fingers selbst Teil einer hüpfenden Tippgeste ist, wird UpPos nicht aktualisiert. Dies ist in dem linken Zweig des Flussdiagramms von [Fig. 17c](#) ersichtlich. Diese Ausnahme ist nötig, um zum Beispiel ein Doppelklicken der rechten virtuellen Taste zu unterstützen. Der Finger wird angehoben, wesentlich nach rechts bewegt, dann tippt er zweimal. Die beiden Tipper erfolgt in etwa am gleichen Ort. Wenn UpPos durch das erste Tippen aktualisiert würde, würde das zweite Tippen als linker Tastenklick entschlüsselt.

[0260] Zweitens: in den Flussdiagrammen von [Fig. 17a](#) bis [Fig. 17f](#) wird UpPos bei einem Tippen, das ein arretierendes Ziehen beendet, nicht aktualisiert. Bedientests zeigen, dass die von Bediener wahrgenommene letzte Stelle des Anhebens für gewöhnlich das letzte Anheben während des arretierenden Ziehens ist, wobei das beendende Tippen eine unterbewusste Aktion ist, deren Ort nicht als relevant wahrgenommen wird. Daher macht es mehr Sinn, die Aktualisierung von UpPos für das beendende Tippen eines arretierenden Ziehens zu übergehen.

[0261] Schritt **346** setzt TapState nach jeder Tipp-, Eckentipp- oder Hüpfgeste auf TIPPEN, wodurch festgehalten wird, dass eine Geste abläuft.

[0262] Unter Bezug nun auf [Fig. 17d](#) läuft Schritt **348** ab, wenn sich der Finger von dem Feld so abhebt, dass dies nicht als Tippen gilt. Dieser Schritt prüft, ob TapState TIPPEN ist; wenn ja, muss sich der Finger nach der langen Bewegung einer Ziehgeste, z.B. Zeit „t7“ von [Fig. 15b](#), nach oben bewegt haben. Abhängig von der Bedienerpräferenz wird die Ziehgeste entweder durch das Anheben des Fingers beendet oder wird arretiert, so dass sie ein arretierendes Ziehen wird.

[0263] Schritt **350** prüft, ob arretierendes Ziehen vom Bediener aktiviert wurde. Diese Entscheidung kann bei einem System, bei dem Ziehen immer arretierend oder immer nicht arretierend ist, zum Zeitpunkt des Designs erfolgen oder sie kann auf einer Laufzeitoption beruhen, beispielsweise einem Steuerfeld.

[0264] Bei Schritt **352** wird ein Ziehen in ein arretierendes Ziehen umgewandelt.

[0265] Bei Schritt **354** wird ein Ziehen durch Anheben des Fingers beendet.

[0266] Bei Schritt **356**, der ausgeführt wird, sobald der Finger angehoben wird und sich nicht als Tippen qualifiziert, wird UpPos auf die aktuelle Position aktualisiert, wie vorstehend beschrieben wurde.

[0267] Unter Bezug nun auf [Fig. 17e](#) läuft Schritt **358** ab, sobald der Finger vom Feld angehoben wird. Die Variable UpTime wird aktualisiert, um die Zeit festzuhalten, bei der der Finger vom Feld hochgehoben wird.

[0268] Schritt **360** wird für jede Abtastung ausgeführt, bei der der Finger dem Feld fern bleibt. Die einzige Verarbeitung ist in diesem Fall das Vergleichen von CurTime minus UpTime mit DragTime, um zu sehen, ob der Finger nach einem Tippen dem Feld zu lange fern blieb, um das Tippen zu einem Ziehen auszuweiten. Wird die Zeitgrenze überschritten und TapState ist gleich TIPPEN, dann geht der Ablauf zu Schritt **362** weiter. (Ansonsten springt der Ablauf zu Schritt **364** weiter, insbesondere wenn TapState gleich ARRETIERT ist. Für die Länge des arretierenden Ziehens gibt es keine Zeitgrenze.)

[0269] Schritt **362** ändert TapState von TIPPEN zu KEINE, was das Tippen beendet und dadurch verhindert, dass das Tippen zu einem Ziehen ausgeweitet wird.

[0270] Unter Bezug nun auf [Fig. 17f](#) laufen alle Strecken bei Schritt **364** zusammen, der unabhängig vom Fingerzustand bei jeder Abtastung ausgeführt wird. Dieser Schritt leitet eine Reihe von Prüfungen zur Ermittlung des Ausgangs der Antippeinheit **280** für diese Abtastung ein. Wenn die Flagge Suppress bei Schritt **364** RICHTIG ist, werden zuerst die virtuellen Tasten unterdrückt, so dass der Ausgang bei Schritt **366** auf KEINE gesetzt wird.

[0271] Wenn die Flagge Suppress FALSCH ist und keine Tastenunterdrücken vorliegt, wird bei Schritt **368** TapState geprüft. Wenn TapState TIPPEN oder ARRETIERT ist, dann wird die durch TapButton angezeigte Taste bei Schritt **370** ausgegeben.

[0272] Wenn TapState KEINE ist, keine Tipp-, Zieh- oder Hüpfgeste abläuft, setzt Schritt **372** in diesem Fall den Ausgang auf KEINE.

[0273] Schritt **374** setzt Suppress auf FALSCH, was anzeigt, dass das Unterdrücken der virtuellen Maustasten für eine Abtastung verwirklicht wurde. Diese Anordnung nimmt an, dass dem Host für jede (X,Y,Z) Abtastung, die von der Recheneinheit eintrifft, Informationen übertragen werden. Ist dies nicht der Fall, sollte Schritt **374** nur ausgeführt werden, sobald ein die unterdrückten virtuellen Tasten enthaltendes Paket tatsächlich dem Host geschickt wird. Es ist auch vernünftig, vor dem Setzen der Flagge Suppress auf FALSCH dem Host mehrere Pakete zu liefern.

[0274] Die Verarbeitung endet bei Schritt **376** (ENDE). Die Antippeinheit beginnt erneut bei Schritt **300** (START), wenn die nächste (X,Y,Z) Abtastung von der Recheneinheit eintrifft.

[0275] Das Randweiterbewegungsmerkmal von [Fig. 13](#) ist während einer Ziehgeste äußerst hilfreich. Es ist daher bevorzugt, die Eingabe MotionEnable der Bewegungseinheit **18** aus dem Zustand der Gesteneinheit **20** ableiten zu lassen. Insbesondere wird das Signal „MotionEnable“ in das UND-Gatter **268** von [Fig. 13](#) durch MotionEnable = (TapState = TIPPEN) ODER (TapState = ARRETIERT) erhalten.

[0276] Die „Zickzack“-Einheit **282** von [Fig. 14](#) entschlüsselt eine Zweifingergeste, bei der ein Finger auf dem Feld ruhen bleibt, während ein anderer Finger seitlich des primären Fingers auftippt. Bezüglich der von dem Grundgerät erzeugten (X,Y,Z) Informationen erhöht diese Geste effektiv den Z-Wert, während der X- und/oder Y-Wert schnell um eine erhebliche Strecke verschoben wird. (Wenn zwei Finger auf dem Feld sind, liegt die gemeldete scheinbare Position zwischen den beiden Fingern.) Wird eine solche Änderung detektiert und folgt ihr eine schnelle Rückkehr zu den ursprünglichen X-, Y- und Z-Werten, dann wird ein Tippen eines zweiten Fingers erkannt.

[0277] Da ein zweites Fingertippen nicht zuverlässig erkannt werden kann, bis der zweite Finger angehoben wird, werden dem Host unvermeidbar plötzliche Cursorbewegungen zuerst zu einer Seite und dann wieder zurück gemeldet. Die Bezeichnung „Zickzack“ bezeichnet diese charakteristischen Cursorbewegungen. Ein Bewegungsumkehrmechanismus ähnlich dem in der Antippeinheit **280** verwendeten kann eingesetzt werden, um sicherzustellen, dass das virtuelle Tastenklicken an der ursprünglichen, nicht zickzackförmigen Position eintritt. Die einzige Schwierigkeit in diesem Fall ist, dass die betroffenen Bewegungen so groß sein können, dass das Beschleunigungsmerkmal des Host ausgelöst wird, was bedeutet, dass die Umkehrbewegung entweder gespeichert und Paket für Paket wiederholt werden muss oder die Zickzackeinheit und die Hostsoftware zusammenwirken müssen, um sicherzustellen, dass der Cursor wirklich zu der erwünschten Stelle zurückkehrt.

[0278] Es ist möglich, das Tippen zweiter Finger mit Hilfe nur der (X,Y,Z) Informationen aus der standardmäßigen Recheneinheit **16** zu erkennen, wie hierin beschrieben wird. Es ist aber klar, dass die Recheneinheit so abgewandelt werden könnte, dass weitere Informationen erzeugt werden, beispielsweise die Breite oder Form des Sensorleiterbahnenprofils, was bei der präzisen Erkennung dieser Geste helfen würde.

[0279] [Fig. 18a](#) bis [Fig. 18c](#) sind ein Flussdiagramm, das den Algorithmus für die Zickzack-Gesteneinheit **282** beschreibt. Wie bei der Antippeinheit **280** wird die Zickzackeinheit am besten als Flussdiagramm beschrieben. Eine Hardware-Zustandsmaschine ist aber ein bekanntes Pendant und wäre auch eine vernünftige Implementierung dieser Einheit. Im Gegensatz zum Flussdiagramm der Antippeinheit **280** der [Fig. 17a](#) bis [Fig. 17f](#) wird das Flussdiagramm der Zickzackeinheit einmal pro Bewegung ausgeführt. Wenn das Vorhandensein des Fingers detektiert wird ($Z > Z_{\text{thresh}}$), beginnt der Ablauf bei Schritt **380**. Wenn der Finger das Feld vor dem Ende des Ablaufs verlässt, bricht die Zickzackeinheit ihre Berechnung ab und beginnt bei der nächsten Bewegung erneut bei Schritt **380**.

[0280] [Fig. 18a](#) bis [Fig. 18c](#) veranschaulichen das weitere Merkmal, dass linksgerichtete Zickzackgesten ein linkes Tastenklicken simulieren, während rechtsgerichtete Zickzackgesten ein rechtes Tastenklicken simulieren.

[0281] Die Zickzackeinheit **282** erfordert die gleiche Position Z und die gleichen Zeiteingaben wie die Antippeinheit **280**. Sie fordert auch ein Geschwindigkeitsmaß S, das als die Strecke von der vorherigen zur aktuellen Fingerposition zu jedem vorgegebenen Zeitpunkt berechnet wird. Wenn bei den normalen (X, Y) Ausgaben der Recheneinheit **16** ein Filtern oder Glätten ausgeführt wird, wie vorstehend offenbart wurde, ist es am Besten, die Geschwindigkeit S aus den ungefilterten (X,Y) Werten zu berechnen.

[0282] Zustandsvariable der Zickzackeinheit umfassen ZigZ und ZigZ', die die beiden letzten Werte für Z festhalten; ZigPos und ZigPos', die die beiden letzten Positionen festhalten; ZigTime, das die Zeit festhält, bei der das Vorhandensein des zweiten Fingers detektiert wurde; ZigLeft und ZigRight, die RICHTIG sind, wenn eine linksgerichtete bzw. eine rechtsgerichtete Zickzackgeste detektiert wurde; und Out, das die Ausgabe der Zickzackeinheit darstellt und eines von LINKS, RECHTS oder KEINE ist.

[0283] Die Zickzackeinheit **282** verwendet mehrere Parameter. ZigDistance, die Mindeststrecke, um die sich die Fingerposition bewegen kann, um sich für diese Geste zu qualifizieren. ZigMaxTime ist der maximale Zeitbetrag, den der zweite Finger anwesend sein kann, um sich zu qualifizieren. Szig ist die augenblickliche Fingergeschwindigkeit, die erforderlich ist, um die Detektion der Geste zu beginnen, und abhängig von der Abtastungsrate, den Sensormaßen und dem Betrag analogen Filterns in den Ladungsintegratoren experimentell ermittelt wird. ZigRadius und ZigLimit geben an, wie nahe die Position bzw. die Z-Werte zu ihren ursprünglichen Vor-Zickzackwerten zurückkehren müssen, nachdem der zweite Finger angehoben wurde. ZigRadius ist mit TapRadius vergleichbar, und ZigLimit beträgt etwa 30% von Zthresh in der derzeit bevorzugten Ausführung.

[0284] Unter Bezug nun auf [Fig. 18a](#) beginnt der Ablauf, wenn bei Schritt **380** das Vorhandensein des Fingers detektiert wird.

[0285] Bei Schritt **382** wartet die Zickzackeinheit, bis etwa drei (X,Y,Z) Abtastungen von der Recheneinheit eintreffen. In der bevorzugten Ausführung treffen diese Abtastungen bei einer Geschwindigkeit von 40 pro Sekunde ein. Diese Anfangsverzögerung dient dazu, ein Verwechseln von Z-Fluktuationen zu Beginn der Bewegung mit einem zweiten Finger zu verhindern.

[0286] Bei Schritt **384** werden ZigZ und ZigZ' auf einen reservierten Wert, der mit KEINE bezeichnet ist, initialisiert.

[0287] Bei Schritt **386** wartet die Einheit auf das Eintreffen der nächsten (X,Y,Z) Abtastung.

[0288] Schritt **388** prüft auf das Beginnen eines „Zick“, der ersten Hälfte der Zickzackgeste, bei der der scheinbare Finger wächst und zu einer Seite springt. Die Geschwindigkeit S der aktuellen Abtastung wird mit dem Schwellwert Szig verglichen. Ist S größer und enthält ZigZ gültige Daten (nicht den reservierten Wert KEINE), dann geht der Ablauf zur weiteren Validierung der Geste in [Fig. 18b](#) weiter.

[0289] Bei Schritt **390** wurde kein anfängliches „Zick“ gesehen, daher wird ZigPos' aktualisiert, um die letzte Fingerposition wiederzugeben, und ZigPos wird aktualisiert, um die vorletzte Fingerposition wiederzugeben. Wird am Ausgang der Recheneinheit **16** von [Fig. 1](#) und [Fig. 8](#) Glätten oder Filtern angewandt, dann sollte im Gegensatz zu der früher beschriebenen S-Berechnung ZigPos aus den gefilterten oder geglätteten Positionsdaten aktualisiert werden. Es sollte mit anderen Worten aus den verarbeiteten Positionsdaten aktualisiert werden, die zum Aktualisieren der Cursorposition am Host verwendet werden.

[0290] Bei Schritt **392** werden ZigZ' und ZigZ ähnlich aktualisiert, um die zwei letzten Z-Werte wiederzugeben. Bei typischen Gebrauchsmustern tritt ein Tippen des zweiten Fingers für gewöhnlich links oder rechts ein, d.h. bei einem anderen X, aber nicht unbedingt bei einem anderen Y. Daher neigt der X-Nenner (Ausgabe des Subtrahierers **168** von [Fig. 8](#)) dazu, um einen eindeutigen Faktor von 2 zu steigen, wenn ein zweiter Finger vorhanden ist, während der Y-Nenner (Ausgabe von Subtrahierer **172**) abhängig von der Linearität der Ladungsintegratoren **144** entsprechend steigen kann oder auch nicht. Daher ist es bevorzugt, die Ausgabe des X-Nenners direkt als Z für die Zwecke der Zickzackeinheit zu verwenden, statt des kombinierten und verarbeiteten Werts, der normalerweise von dem Block **178** von [Fig. 8](#) erhalten wird.

[0291] Nach Schritt **392** kehrt der Ablauf zu Schritt **386** zurück, wo die nächste Abtastung abgewartet wird.

[0292] Unter Bezug nun auf [Fig. 18b](#) hält der Schritt **394** den Zeitpunkt fest, bei dem das anfängliche „Zick“ detektiert wurde.

[0293] Schritt **396** initialisiert dann die Flaggen ZigLeft und ZigRight. Diese Flaggen werden RICHTIG, wenn ein Bewegen des Fingers erheblich weit nach links bzw. rechts von seiner Ausgangsposition gesehen wird.

[0294] Wenn ein Finger auf das Feld auftritt, nehmen die (X,Y,Z) Werte typischerweise zwei oder drei Abtastungen, um ihre neuen Werte anzugleichen, welche das Vorhandensein der beiden Finger wiedergeben. Schritt **398** wartet, bis ein oder zwei weitere Abtastungen eintreffen, woraufhin sich die (X,Y,Z) Werte eingepegelt ha-

ben sollten. Die Wahl von ein, zwei oder mehr Abtastungen hängt von Faktoren wie der grundlegenden Abtastgeschwindigkeit und dem Filterbetrag ab, der im analogen Eingangsabschnitt des Geräts erfolgt.

[0295] Nach Schritt **398** gibt CurPos die scheinbare Zick-Fingerposition wieder, und ZigPos gibt die Position von den beiden Abtastungen wieder, bevor die Geschwindigkeit den Schwellwert Szig überschritt. Der Verlauf der zwei Abtastungen ist wichtig, da aufgrund des sich nähernden zweiten Fingers ein kleiner Bewegungsbeitrag eingetreten sein kann, bevor der Finger berührt und die große Bewegung erzeugte, die Szig überstieg. Nach Schritt **398** enthält ZigPos die aktuelle Position, die zu einem Zeitpunkt gespeichert wurde, da der zweite Finger wahrscheinlich eine Wirkung gehabt hatte. Analog enthält ZigZ den Z-Wert vor dem Auftreffen des zweiten Fingers.

[0296] Schritt **400** prüft, um zu sehen, ob Z den ruhenden Z-Wert ZigZ wesentlich überschritten hat. In der derzeit bevorzugten Ausführung wird Z mit einem Schwellwert verglichen, der 30% größer als ZigZ ist. Ist Z zu klein, wird „Zick“ disqualifiziert und der Ablauf kehrt zu Schritt **384** zurück.

[0297] Schritt **402** prüft, um zu sehen, ob die aktuelle Position weit links der Ruheposition ZigPos ist. Da die Zickzackigkeit **282** nach einer abrupten „unnatürlichen“ Änderung der Position sucht, ist bevorzugt, dass der Schritt **402** vor dem Filtern oder Glätten, das normalerweise auf Positionsdaten angewendet wird, Positionsdaten direkt von den Dividierern **174** und/oder **176** der Recheneinheit verwendet. Diese Daten werden hierin als RawPos bezeichnet, um sie von dem gefilterten und geglätteten Wert CurPos zu unterscheiden. Der Wert CurPos kann aber bei Bedarf mit nicht ganz optimalen Ergebnissen verwendet werden.

[0298] Bei dieser Implementierung vergleicht Schritt **402** die X-Koordinate von RawPos mit der X-Koordinate von ZigPos minus ZigDistance. Der Parameter ZigDistance kann beruhend auf dem beobachteten Abstand zwischen zwei Fingern auf dem Feld, wenn ein Finger nach unten gehalten wird und der andere in natürlicher Weise auftippt, experimentell gewählt werden.

[0299] Wenn ein geeignetes linksgerichtetes „Zick“ detektiert wird, setzt Schritt **404** ZigLeft auf RICHTIG.

[0300] Schritt **406** prüft analog, ob die aktuelle Position weit rechts der Ruheposition ist; wenn ja setzt Schritt **408** ZigRight auf RICHTIG.

[0301] Schritt **410** wartet dann auf das Eintreffen der nächsten (X,Y,Z) Abtastung.

[0302] Schritt **412** prüft, ob der zweite Finger vom Feld hochgehoben wurde, indem er Z mit einem zweiten „Zack“-Schwellwert vergleicht, der etwas kleiner als der „Zick“-Schwellwert von Schritt **400** ist. (In dem aktuellen System ist dieser Schwellwert etwa 20% größer als ZigZ.). Der Schwellwert „Zack“ wird unter dem Schwellwert „Zick“ angesetzt, um eine einfache Hysterese zu ermöglichen.

[0303] Wurde der zweite Finger noch nicht angehoben, kehrt der Ablauf zu Schritt **402** zurück und wartet weiter. Wurde der zweite Finger angehoben, geht der Ablauf zu Schritt **414** in [Fig. 18c](#).

[0304] Unter Bezug nun auf [Fig. 18c](#) wartet Schritt **414** ein oder zwei Abtastungen lang, dass sich die (X,Y,Z) Daten stabilisieren, wenn sich der zweite Finger hebt; dieser Schritt ist analog zu Schritt **398**.

[0305] Schritt **436** nimmt eine letzte Prüfung auf eine vollständige Zickzackgeste hin vor. Bis jetzt wurde eine plötzliche Bewegung begleitet von einer Zunahme von Z gesehen, gefolgt von einer plötzlichen Abnahme von Z. Schritt **416** prüft zusätzlich, dass die Position zu ihrem Vorzickzackwert zurückgekehrt ist (die Strecke von RawPos zu ZigPos ist kleiner als ZigRadius); der Z-Wert ist analog auf den Normalwert zurückgekehrt (der absolute Wert von Z minus ZigZ ist kleiner als ZigLimit); und entweder ZigLeft oder ZigRight, aber nicht beide, ist RICHTIG. Ferner muss das Tippen des zweiten Fingers von kurzer Dauer sein, d.h. CurTime minus ZigTime ist kleiner als ZigMaxTime.

[0306] Wenn sich eine Bewegung nicht als Zickzack qualifiziert, kehrt der Ablauf zu Schritt **384** zurück, um auf die Detektion einer Zickzackgeste zu warten. Hat sich die Bewegung als Zickzack qualifiziert, sieht Schritt **418** eine Rückwärtsbewegung vor, um den Cursor bei Bedarf auf die exakte Position zurückzuführen, die ZigPos entspricht. Dieser Schritt ist analog zu Schritt **322** von [Fig. 17b](#).

[0307] Bei Schritt **420** wird vollständiger Zickzack detektiert. Wenn ZigLeft RICHTIG ist, ist die Bewegung ein linksgerichteter Zickzack. Ansonsten muss ZigRight RICHTIG sein und die Bewegung ist ein rechtsgerichteter

Zickzack. Dementsprechend simuliert Schritt **442** entweder ein Drücken der linken Taste für einen linksgerichteten Zickzack oder Schritt **424** simuliert ein Drücken der rechten Taste für einen rechtsgerichteten Zickzack.

[0308] Schritt **426** pausiert eine bestimmte Zeitdauer. Zum Beispiel könnte dieser Schritt das Eintreffen von ein oder mehreren weiteren Abtastungen abwarten oder könnte das Senden von ein oder mehreren Paketen an den Host abwarten. (Normalerweise besteht eine 1:1-Entsprechung zwischen Abtastungen und Datenpaketen.)

[0309] Schließlich beendet Schritt **428** das simulierte Tastendrücken, indem er Out auf KEINE setzt. In diesem Beispiel bewirkt die Zickzackgeste nur das Simulieren von Klicken, nicht von Ziehen. Das Zickzack geht nicht sauber in ein Ziehen über, wie das normale Tippen mit einem Finger, da dies implizieren würde, dass die gesamte Ziehbewegung mit zwei unbequem auf dem Feld gehaltenen Fingern eintritt. Eine Alternative ist das Simulieren einer arretierenden Taste, wie es häufig bei Trackball-Tasten auf dem Gebiet getan wird, bei denen konsekutive Zickzackbewegungen abwechselnd die virtuelle Taste drücken und freigeben. Eine andere Alternative besteht darin, die Zickzackgeste die virtuelle Taste drücken zu lassen und die virtuelle Taste nur freizugeben, wenn auch der primäre Finger von dem Feld entfernt wird.

[0310] Nach Schritt **428** kehrt der Ablauf zu Schritt **384** zurück, um die Detektion weiterer Zickzackgesten abzuwarten.

[0311] Eine weitere Geste, die bei spezialisierten Anwendungen nützlich ist, ist eine „Schiebe“-Geste, die einfach die Z-Informationen (Druck) mit einem zweiten Z-Schwellwert ZpushDown vergleicht, der erheblich höher als der grundlegende Fingerdetektionsschwellwert ist, und eine Maustastenaktion simuliert, sobald Z diesen Schwellwert übersteigt. Diese „Schiebe“-Geste ähnelt der Art und Weise, wie Zeigegeräte auf der Basis von Stiften normalerweise arbeiten; sie ist aber zu ungenau und für den Finger zu ermüdend, um als primäre Klick- oder Ziehgeste verwendet zu werden. Die „Schiebe“-Geste ist unter speziellen Umständen wie Programme für freihändiges Zeichnen äußerst brauchbar.

[0312] [Fig. 19](#) ist ein Zeitverlaufdiagramm, das eine „Schiebe“-Geste veranschaulicht. Um diese Geste auszuführen, wird der Finger zuerst nahe genug gebracht, um eine Cursorbewegung zu verursachen, ohne ein Drücken einer virtuellen Taste auszulösen. Als Nächstes nimmt der Fingerdruck über den Schwellwert ZpushDown hinaus zu, was das Drücken der virtuellen Taste verursacht. Später fällt der Druck unter einen Schwellwert ZpushUp, was die Freigabe der virtuellen Taste verursacht. Ist ZpushUp etwas niedriger als ZpushDown, verhindert die sich ergebende Hysterese eine unerwünschte Oszillation an der virtuellen Taste, wenn der Fingerdruck um den „Schiebe“-Schwellwert herum leicht schwankt.

[0313] In einer Variante, die bevorzugt sein kann, wird ZpushUp gleich Zthresh gesetzt, so dass nach Beginn eines Schiebens der Finger vollständig vom Feld gehoben werden muss, um die simulierte Taste freizugeben. Andere Bediener bevorzugen unter Umständen, dass ZpushUp viel näher bei ZpushDown liegt als bei Zthresh, was zu mehr Feinfühligkeit führt.

[0314] Die Schiebeeinheit **284** von [Fig. 14](#) erkennt die Schiebegeste. [Fig. 20](#) ist ein Flussdiagramm, das die Implementierung dieser Geste veranschaulicht. Das entsprechende Diagramm für eine gleichwertige Hardware-Schaltung zum Erkennen dieser Geste wäre recht unkompliziert.

[0315] Der Ablauf beginnt bei Schritt **440**, immer wenn eine neue (X,Y,Z) Abtastung von der Recheneinheit **16** eintrifft. Zu beachten ist, dass die Schiebeeinheit nur den Z-Wert jeder Abtastung prüft.

[0316] Schritt **442** prüft, ob bereits eine „Schiebe“-Geste abläuft oder nicht.

[0317] Schritt **444** wird ausgeführt, wenn keine „Schiebe“-Geste abläuft. Dieser Schritt prüft, ob eine „Schiebe“-Geste beginnen sollte. Zuerst müssen „Schiebe“-Gesten vom Bediener aktiviert werden. Zum anderen muss der aktuelle Z-Wert größer als der Schwellwert ZpushDown sein.

[0318] Wenn Z groß genug ist, um eine Schiebegeste zu beginnen, setzt Schritt **446** Out auf LINKS, um anzuzeigen, dass jetzt die linke Taste gedrückt wird.

[0319] Schritt **448** prüft, ob die aktuelle Schiebegeste enden sollte. Diese Prüfung beinhaltet einfach das Vergleichen von Z mit ZpushUp. Wenn Z kleiner als ZpushUp ist, wird die Schiebegeste bei Schritt **450** beendet.

[0320] Der Ablauf endet bei Schritt **452**. Wenn weder Schritt **446** noch Schritt **450** ausgeführt wurde, dann bleibt Out gleich, wodurch die oben erwähnte Hysterese vorgesehen wird. Die Zustandsvariable Out sollte zum Zeitpunkt des Einschaltens auf KEINE initialisiert werden.

[0321] Der Durchschnittsfachmann wird feststellen, dass die Antippeinheit **280** für die Verwendung mit einem beliebigen Touchpad geeignet ist, das (X,Y) Informationen und Fingeranwesenheitsinformationen liefert, und dass die Schiebeeinheit **284** für die Verwendung mit einem beliebigen Touchpad geeignet ist, das Z-Informationen (Druck) erzeugt. Nur die Zickzackeinheit hängt von den speziellen Eigenschaften der hierin offenbarten bestimmten Touchpad-Technologie ab, nämlich der Tatsache, dass zwei Finger zuverlässig eine gemittelte Fingerposition melden.

[0322] Es können zwei weitere Algorithmen, die nicht direkt Teil der Gestenverarbeitung sind, eingesetzt werden, um kleinere Probleme zu lösen, die sich bei Tippen des Bedieners auf das Feld ergeben. Im Einzelnen schert die Fingerposition manchmal scharf in eine Richtung, gerade wenn der Finger angehoben wird. Das ist auf natürliches Rutschen des Fingers während dieser Aktion zurückzuführen und wird verschlimmert, wenn der Finger bei einem flachen Winkel gehalten wird. Ein „Umkehrbewegungs“-Algorithmus kann dieses Problem zum Teil lösen, wenn aber die scheinbare Fingerposition so weit springt, dass der TapRadius Test scheitert, kann die Umkehrbewegung nicht helfen.

[0323] Wenn ein schnelles Ändern von Z zwischen den aktuellen und den vorherigen Abtastungen gesehen wird (d.h. wenn die absolute Differenz zwischen den aktuellen und den vorherigen Z-Werten unter einem empirisch ermittelten Schwellwert liegt), dann kann die Zeitkonstante der (X,Y) Filterung der Ausgabe der Recheninheit angehoben werden. Normalerweise werden der alte Filterwert und der neue Quotient mit in etwa gleicher Gewichtung gemittelt, um den neuen Filterwert zu erzeugen. Ändert sich Z schnell, wird der alte Filterwert stattdessen erheblich stärker (z.B. um eine Größenordnung) gewichtet als der neue Quotient. Das Ergebnis ist, dass eine während dieses Moments einer hohen Z-Änderung eintretende Bewegung stark gedämpft wird.

[0324] Häufig erfolgt die unerwünschte Bewegung, die sich aus einem Fingerheben ergibt, in der allerletzten Abtastung, bevor Z unter den Finger-Unten-Schwellwert Z_{thres} sinkt. Eine weitere Lösung des Problems der unerwünschten Fingerhebebewegung ist der „Anhebesprungunterdrückungs“-Mechanismus, der versucht, dieses letzte Ereignis der unerwünschten Bewegung zu unterdrücken. [Fig. 21](#) zeigt eine veranschaulichende Schaltung für das Ausführen der Funktion Anhebesprungunterdrückung.

[0325] Die in [Fig. 21](#) gezeigte Schaltung führt die Anhebesprungunterdrückung aus. Sie prüft die Folge von (X,Y) Positionsabtastungen, die von den Dividierern **174** und **176** von [Fig. 8](#) eintreffen, um eine Geschwindigkeit S zu erzeugen, die weiter bearbeitet wird, um ein Bewegungsunterdrückungssignal zu erhalten. Wie zuvor beschrieben, ist es am Besten, bei der Berechnung der Geschwindigkeit S die Quotientenwerte direkt vor einer Glättungs- oder Filterungsphase zu verwenden.

[0326] Unter Bezug auf [Fig. 21](#) werden die X-Koordinaten in der Verzögerung **460** gespeichert. Der Subtrahierer **462** berechnet den absoluten Wert der Differenz zwischen dem aktuellen X-Wert und dem vorherigen Wert, der in der Verzögerung **460** gespeichert ist. Analog berechnen die Verzögerung **464** und der Subtrahierer **466** die absolute Änderung von Y. Der Addierer **468** bildet die Summe dieser absoluten Differenzen, um die Geschwindigkeit S, die Strecke zwischen den aktuellen und vorherigen Abtastungen, zu erzeugen. Wie vorstehend beschrieben ist klar, dass andere Distanzmaßnahmen für diese Berechnung herangezogen werden können. Zu beachten ist, dass neben der Schaltung von [Fig. 21](#) die Zickzackeinheit **282** auch den Geschwindigkeitswert S heranzieht, wie vorstehend beschrieben wurde.

[0327] Verzögerungseinheiten **470** und **472** halten die letzten und vorletzten Werte von S, die jeweils als S' bzw. S'' bekannt sind, fest. Der Dividierer **474** berechnet die Größe halb S, die als S/2 bezeichnet wird. Die Anhebesprungunterdrückungseinheit sucht in dem Versuch, unerwünschte Anhebesprünge zu erkennen, nach einer charakteristischen Beziehung zwischen den Werten S, S', S'' und S/2. Ein Fachmann wird erkennen, dass S'' nicht vor der vierten Abtastung einer vorgegebenen Fingerbewegung gültig ist; dadurch ist die Anhebesprungunterdrückungseinheit die ersten drei Abtastungen jeder Bewegung deaktiviert. Die Anhebesprungunterdrückungseinheit verwendet auch einen Parameter LiftJump, einen Geschwindigkeitsschwellwert, der experimentell ermittelt wird und durch die Abtastungsgeschwindigkeit und die Empfindlichkeit des Sensorfelds beeinträchtigt wird.

[0328] Der Komparator **476** prüft, ob die Geschwindigkeit S größer als der Schwellwert LiftJump ist. Der Komparator **478** prüft, um zu sehen, ob die vorherige Geschwindigkeit S' kleiner als LiftJump ist, und der Kompa-

rator **480** prüft, ob S' kleiner als $S/2$ ist. Analog prüft der Komparator **482**, um zu sehen, ob die vorletzte Geschwindigkeit S'' kleiner als LiftJump ist, und der Komparator **484** prüft, ob S'' kleiner als $S/2$ ist. Wenn alle fünf Bedingungen erfüllt sind, gibt das UND-Gatter **486** ein „Bewegungsunterdrückungs“-Signal aus, das das Arbeiten der Bewegungseinheit **18** für diese Abtastung unterdrückt. Wenn die Bewegungseinheit **18** unterdrückt wird, wird ihre Ausgabe ($\Delta X, \Delta Y$) für die aktuelle Abtastung nicht erzeugt und ihre Verzögerungseinheit **260** wird nicht getaktet.

[0329] Das von der Anhebesprungunterdrückungseinheit detektierte Profil tritt meist während einer letzten unerwünschten Bewegungsabtastung vor dem Anheben des Fingers ein. Da Z bei der allernächsten Abtastung unter Z_{thresh} fällt, wird die aktuelle Abtastung nie zu Bewegungsereignissen beitragen, die dem Host übermittelt werden. Der Algorithmus unterdrückt ausgestaltungsbedingt garantiert nicht mehr als eine Abtastung in einer Reihe. Wenn daher der Algorithmus „falsch rät“ und Z nicht unter Z_{thresh} fällt, wird die übersprungene Fingerbewegung mit einem nur winzigen Zögern bei der wahrgenommenen Cursorbewegung in das ($\Delta X, \Delta Y$) Paket aufgenommen, das durch die nächste Abtastung erzeugt wird.

[0330] Die erhöhte Empfindlichkeit des Berührungssensorsystems lässt auch eine leichtere Eingabefingerberührung zu, was es einfach für den menschlichen Gebrauch macht. Eine größere Empfindlichkeit macht es auch einfacher, andere Eingabeobjekte zu verwenden, wie Eingabestifte, etc. Ferner lässt diese Empfindlichkeit im Gegenzug eine dickere Schutzschicht oder andere Materialien zu, welche beide niedrigere Herstellungskosten ermöglichen können.

[0331] Eine stärkere Störunterdrückung gestattet eine größere Flexibilität im Gebrauch und eine verminderte Empfindlichkeit gegenüber Problemen mit Störerauschen. Es werden zwei Techniken eingesetzt, die eine Herleitung des größtmöglichen Rauschunterdrückungsvorteils ermöglichen.

[0332] Aufgrund der verwendeten Ansteuer- und Erfassungstechniken ist die Datenerfassungsgeschwindigkeit um etwa einen Faktor 30 gegenüber dem Stand der Technik gestiegen. Dies bietet mehrere offensichtliche begleitende Wirkungen. Zum einen kann bei gleichem Umfang an Signalverarbeitung die Schaltung die meiste Zeit abgeschaltet werden und der Stromverbrauch in dem analogen Teil der Konstruktion um etwa einen Faktor 30 gesenkt werden. Da mehr Daten verfügbar sind, kann zum anderen mehr Signalverarbeitung wie Filtern und Gestenerkennung durchgeführt werden.

[0333] Die verwendete Sensorelektronik ist sehr robust und kalibriert Prozess- und systematische Fehler heraus. Sie verarbeitet die kapazitiven Informationen von dem Sensor und liefert einem externen Gerät, beispielsweise einem Mikroprozessor, digitale Informationen.

[0334] Aufgrund der einzigartigen physikalischen Eigenschaften gibt es einige ergonomisch interessante Anwendungen, die zuvor nicht möglich waren. Derzeit ist die Verwendung einer Maus oder eines Trackballs auf tragbaren Computern physikalisch nicht praktisch. Es wird eine sehr praktische und einfach zu bedienende Cursorpositionslösung an die Hand gegeben, die diese Geräte ersetzt.

[0335] Bei mausartigen Anwendungen kann der erfindungsgemäße Sensor an einer praktischen Position angeordnet werden, z.B. unter der „Leertaste“ eines tragbaren Computers. Bei Anordnung an dieser Position kann der Daumen des Bedieners als Positionszeiger auf dem Sensor verwendet werden, um die Cursorposition auf dem Computerbildschirm zu steuern. Der Cursor kann dann bewegt werden, ohne dass die Finger des Bedieners die Tastatur verlassen müssen. Ergonomisch ähnelt dies dem Konzept des Macintosh Power Book mit seinem Trackball, es wird aber ein erheblicher Größenvorteil gegenüber dem Trackball geboten. Erweiterungen dieser Grundidee sind dahingehend möglich, dass für eine noch bessere Steuerung von Merkmalen zwei Sensoren unterhalb der „Leertaste“ angeordnet werden könnten.

[0336] Die Computeranzeige mit ihrer Cursorsrückmeldung ist ein kleines Beispiel für einen sehr allgemeinen Anwendungsbereich, bei dem eine Anzeige ein Feld von Leuchtdioden oder LEDs, eine LCD-Anzeige oder eine Kathodenstrahlröhre sein könnte. Beispiele umfassen Berührungssteuerungen an Laborgeräten, bei denen bisherige Geräte eine Kombination aus Knopf/Taste/berührungsempfindlichem Bildschirm verwenden. Aufgrund der Angliederungsfähigkeit dieser Schnittstelle könnten eine oder mehrere der Eingänge zu einem der beschriebenen Eingänge kombiniert werden.

[0337] Unterhaltungselektronikgerät-Anwendungen (Stereos, Terzfilter, Mischgeräte) nutzen häufig die Oberfläche der Vorderseite für Gleitpotentiometer, da eine variable Steuerung erforderlich ist. Eine solche Steuerung kann an einer kleinen Touchpad-Position vorgesehen werden. Da elektronische Heimsysteme verbreite-

ter werden, ist eine dichtere und leistungsstärkere Schnittstelle zum Menschen erforderlich. Die beschriebene Sensortechnologie erlaubt ein sehr dichtes Steuerfeld. Handgehaltene TV/Video/Stereo-Bedienungen könnten ergonomisch geformt werden und leistungsstärkere Merkmale ermöglichen, wenn diese Sensortechnologie verwendet wird.

[0338] Der Sensor kann jeder Oberfläche angepasst werden und kann so ausgelegt werden, dass er mehrere Berührungspunkte detektiert, was einen leistungsstärkeren Joystick möglich macht. Die einzigartige Druckdetektionsfähigkeit der Sensortechnologie ist auch für diese Anwendung ausschlaggebend. Computerspiele, „Fern“-Bedienungen (Hobby-Elektronik, Flugzeuge) und Werkzeugmaschinensteuerungen sind ein paar Beispiele für Anwendungen, die von der Sensortechnologie profitieren würden.

[0339] Tasteninstrumente (Synthesizer, elektrische Klaviere) erfordern auf Geschwindigkeit ansprechende Tasten, die durch die Druckerfassungsfähigkeit dieses Sensors vorgesehen werden können. Es gibt auch Tonhöhe beugende Steuerungen und andere Gleitschalter, die durch diese Technologie ersetzt werden könnten. Eine noch einzigartigere Anwendung umfasst ein Musikinstrument, das Noten als Funktion der Position und des Drucks der Hände und Finger in einer sehr ausdrucksfähigen 3D-Schnittstelle erzeugt.

[0340] Die Sensortechnologie kann am besten jedes gegen sie pressende leitende Material detektieren. Durch Hinzufügen einer komprimierbaren Isolierschicht, die von einer Schicht leitenden Materials bedeckt ist, oben auf dem Sensor kann der erfindungsgemäße Sensor auch indirekt Druck von einem bedienten Objekt unabhängig von dessen elektrischer Leitfähigkeit detektieren.

[0341] Aufgrund der von diesem Sensor verfügbaren Informationsmenge eignet er sich sehr gut als Eingabegerät für virtuelle Realität erzeugende Maschinen. Es ist leicht, sich eine Konstruktion vorzustellen, die eine Positionsüberwachung in drei Dimensionen und ein gewisses Maß an Reaktion (Druck) auf die Aktionen zulässt.

Patentansprüche

1. Verfahren für das Kompensieren einer unbeabsichtigten seitlichen Bewegung bei Antippen eines Berührungssensorfelds in einem Berührung erfassenden System, welches einem Host X- und Y-Positionsangaben liefert, wobei es die folgenden Schritte umfasst:

Detektieren **(320)** des Eintretens eines von einem leitenden antippenden Objekt vorgenommenen Antippens des Berührungssensorfelds; und

bei Detektieren des Eintretens eines Antippens Senden eines Signals an den Host, welches das Eintreten des Antippens anzeigt; wobei das Verfahren durch folgendes gekennzeichnet ist:

Senden von X- und Y-Positionsangaben an den Host, welche eine seitliche Bewegung des antippenden Objekts an dem Berührungssensorfeld während des Antippens darstellen; und

bei Detektieren des Eintretens eines Antippens Senden **(322)** von weiteren X- und Y-Positionsangaben an den Host, welche eine Bewegung entgegengesetzt zur seitlichen Bewegung des antippenden Objekts an dem Berührungssensorfeld während des Antippens darstellen.

2. Verfahren nach Anspruch 1, welches weiterhin den Schritt des Detektierens **(338)**, in welchem von mehreren vorbestimmten Bereichen das Antippen eingetreten ist, umfasst.

3. Verfahren nach Anspruch 1, welches weiterhin folgende Schritte umfasst:

Vergleichen der Zeitdauer, die das leitende Objekt auf dem Berührungsfeld verweilt, mit einer Bezugszeitdauer; und

Vergleichen des Betrags der von dem leitenden Objekt ausgeführten Bewegung, während es auf dem Berührungsfeld verweilt, mit einem Bezugsbewegungsbetrag; und wobei

ermittelt wird, dass ein Antippen eingetreten ist, wenn die Zeitdauer des Verweilens des leitenden Objekts auf dem Berührungsfeld unter der Bezugszeitdauer liegt und wenn der Betrag der von dem leitenden Objekt ausgeführten Bewegung während dessen Verweilens auf dem Berührungsfeld unter dem Bezugsbewegungsbetrag liegt, wobei das Signal eine vorbestimmte Zeitdauer aufrecht erhalten wird.

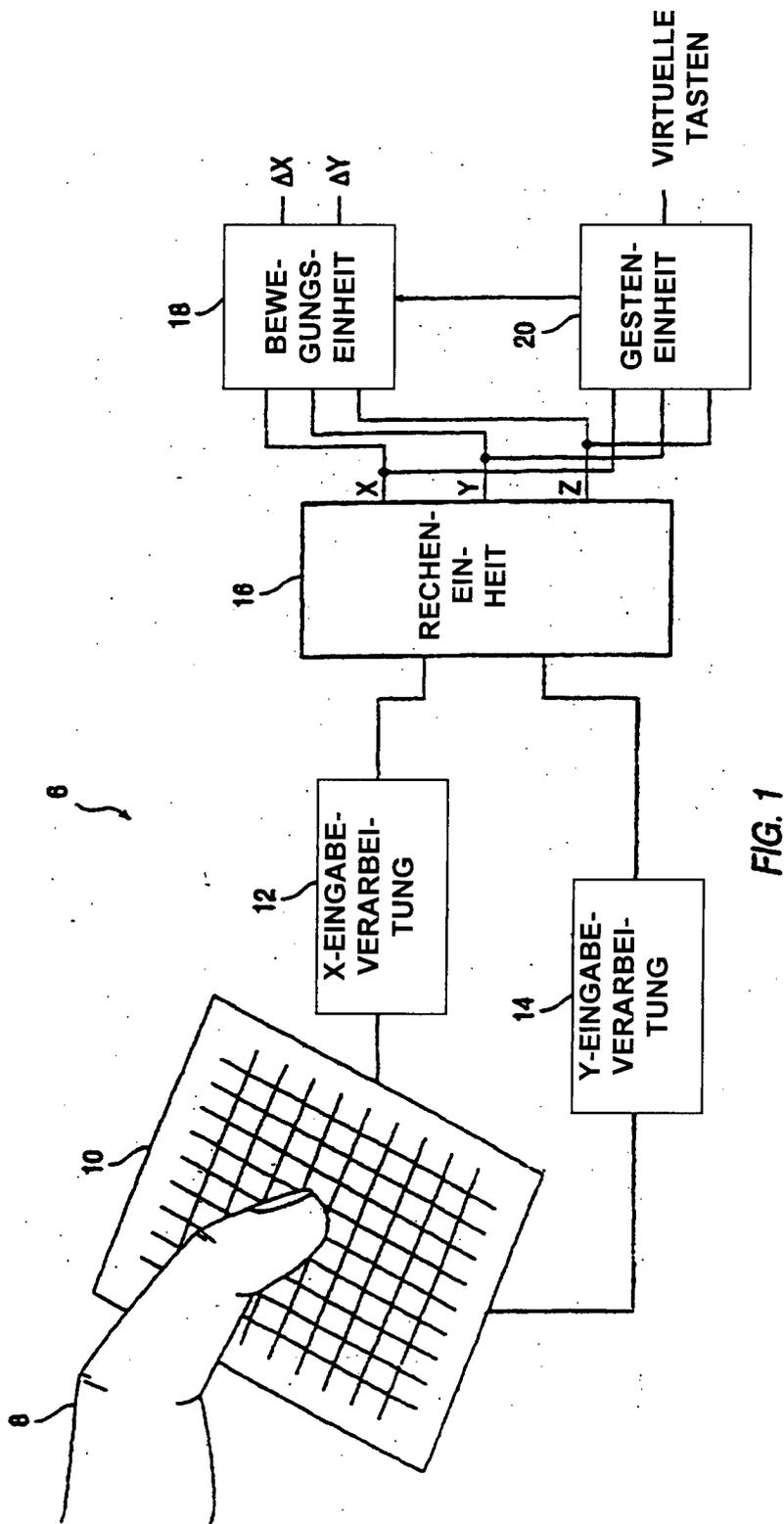
4. Verfahren nach Anspruch 3, welches weiterhin die folgenden Schritte umfasst

Detektieren **(338)**, in welchem von mehreren vorbestimmten Bereichen das Antippen eingetreten ist; und

Senden eines zweiten Signals an den Host, welches anzeigt, in welchem der vorbestimmten Bereiche auf dem Sensorfeld das Antippen eingetreten ist.

Es folgen 26 Blatt Zeichnungen

Anhängende Zeichnungen



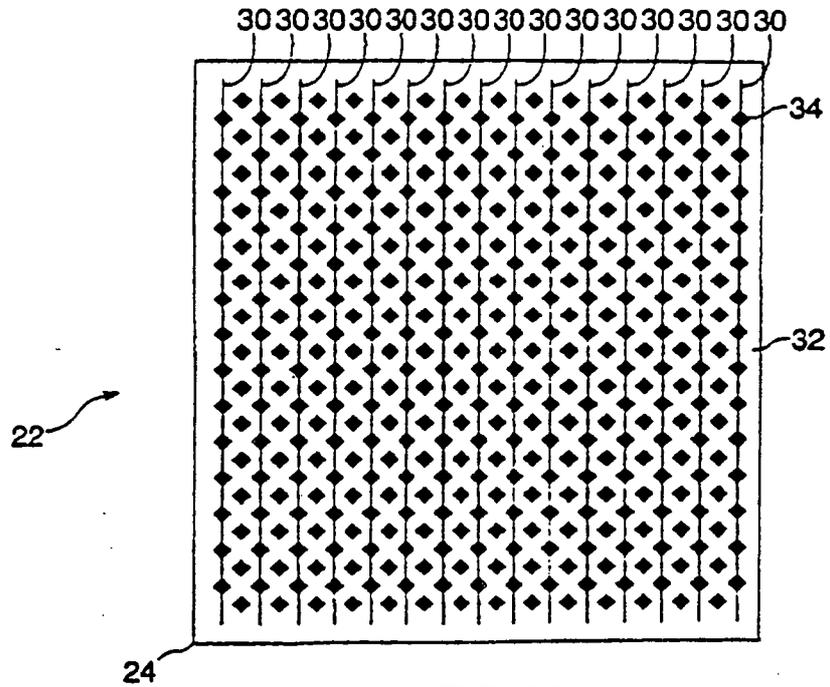


FIG. 2A

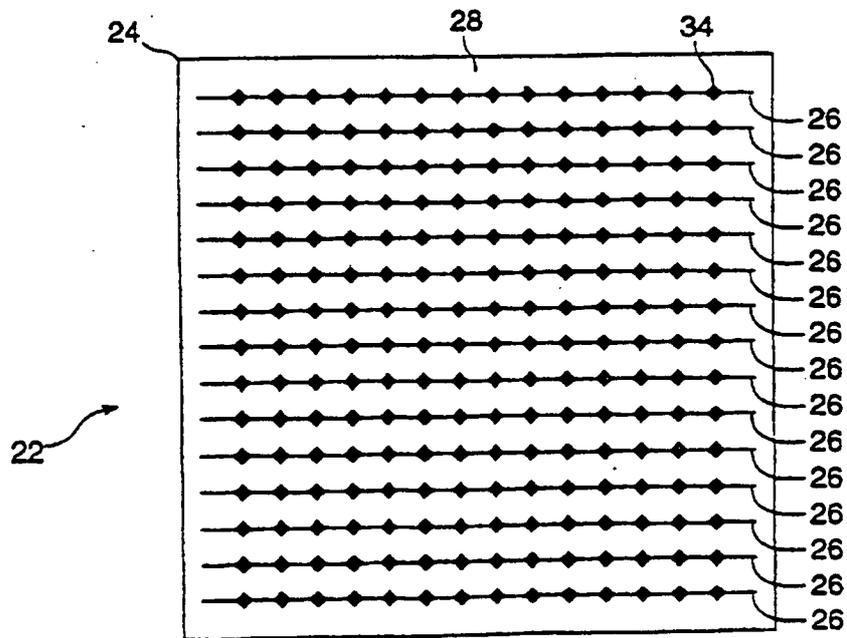
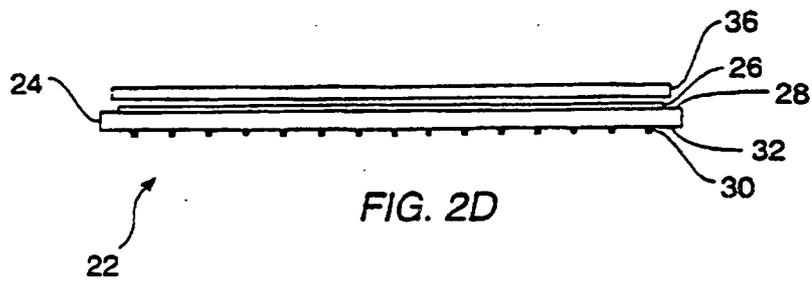
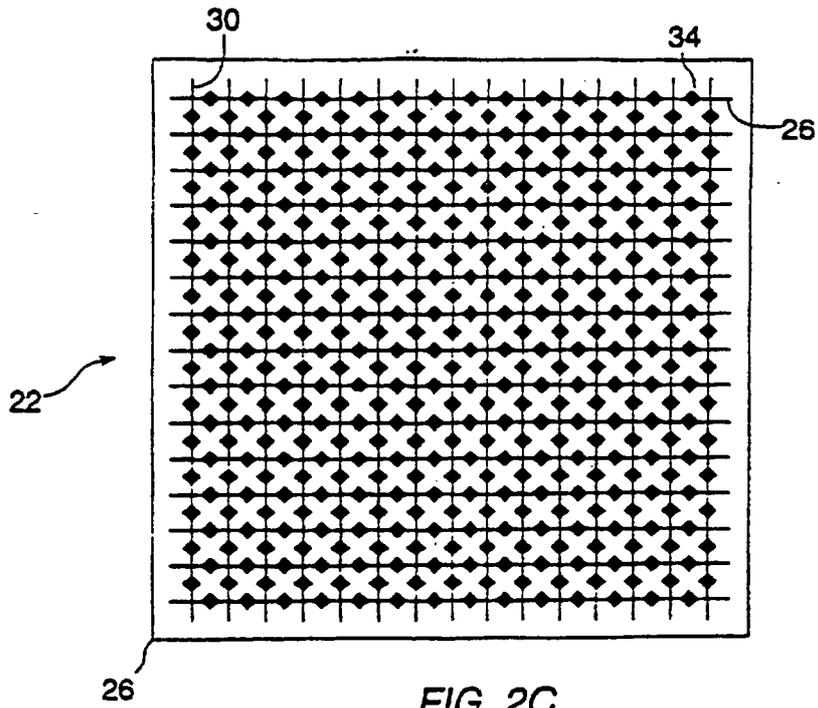


FIG. 2B



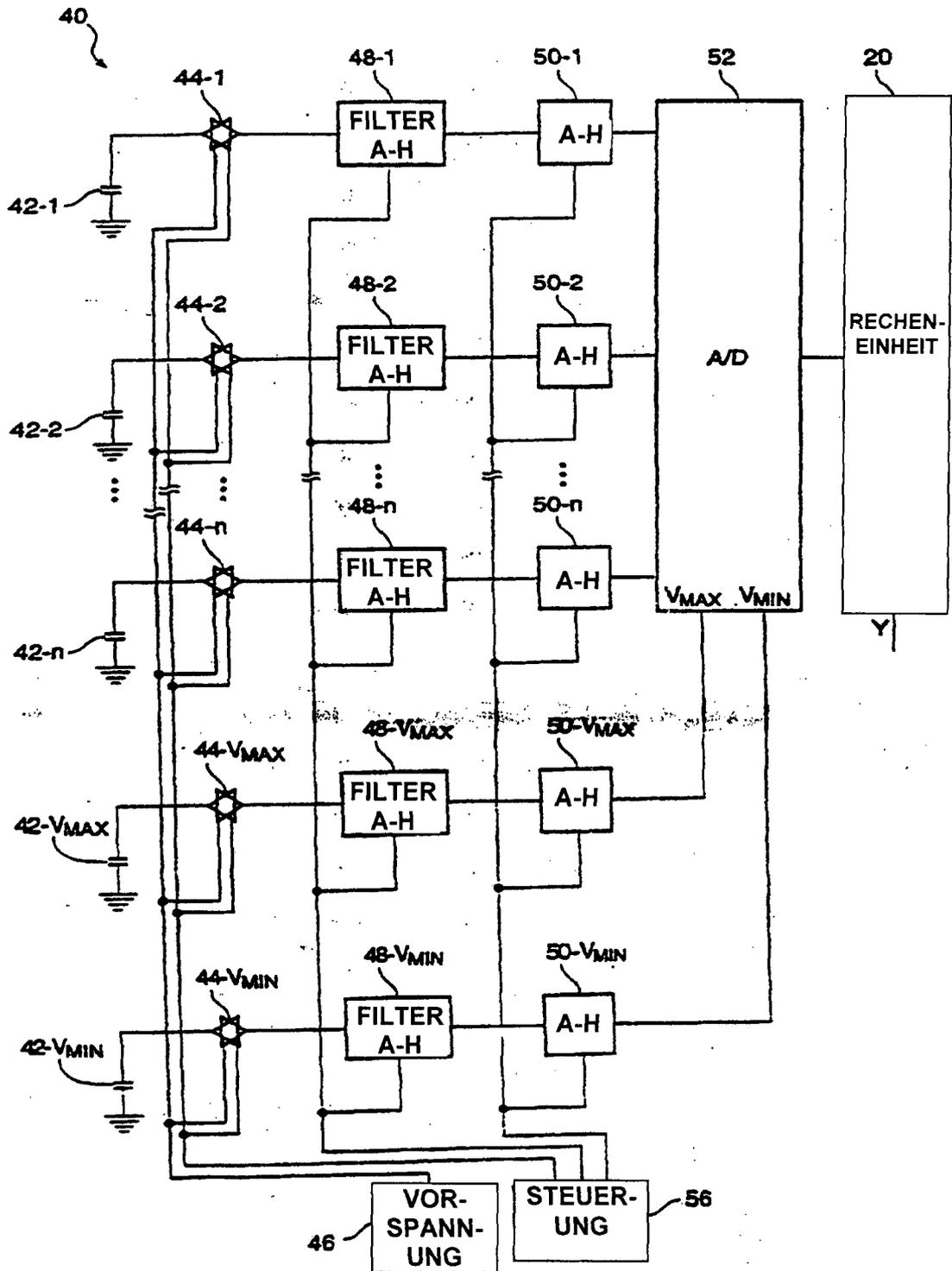


FIG. 3

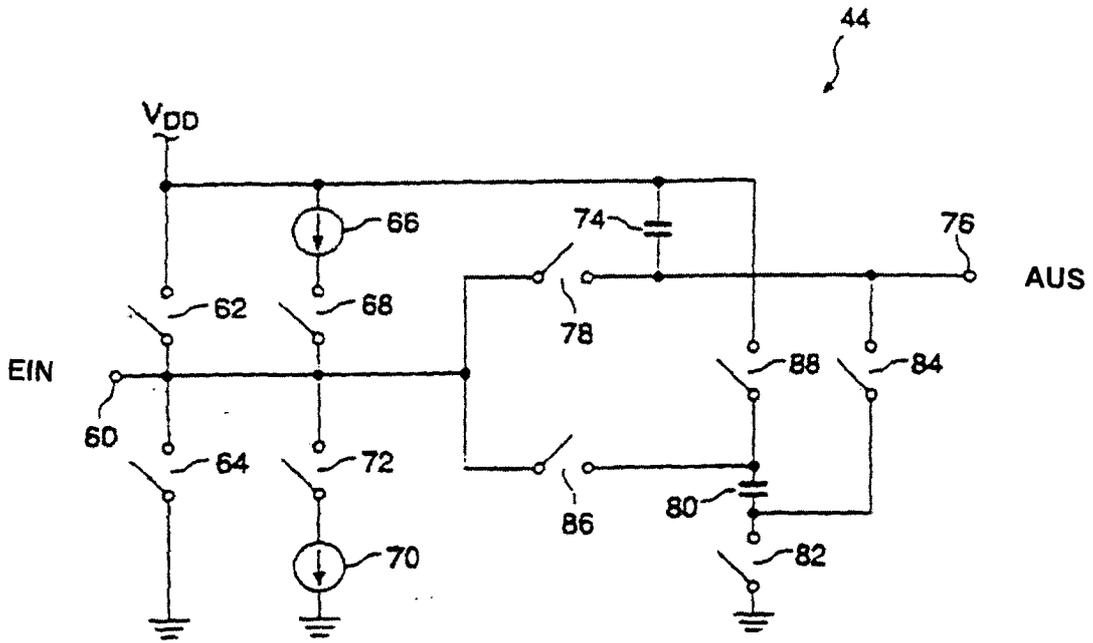


FIG. 4A

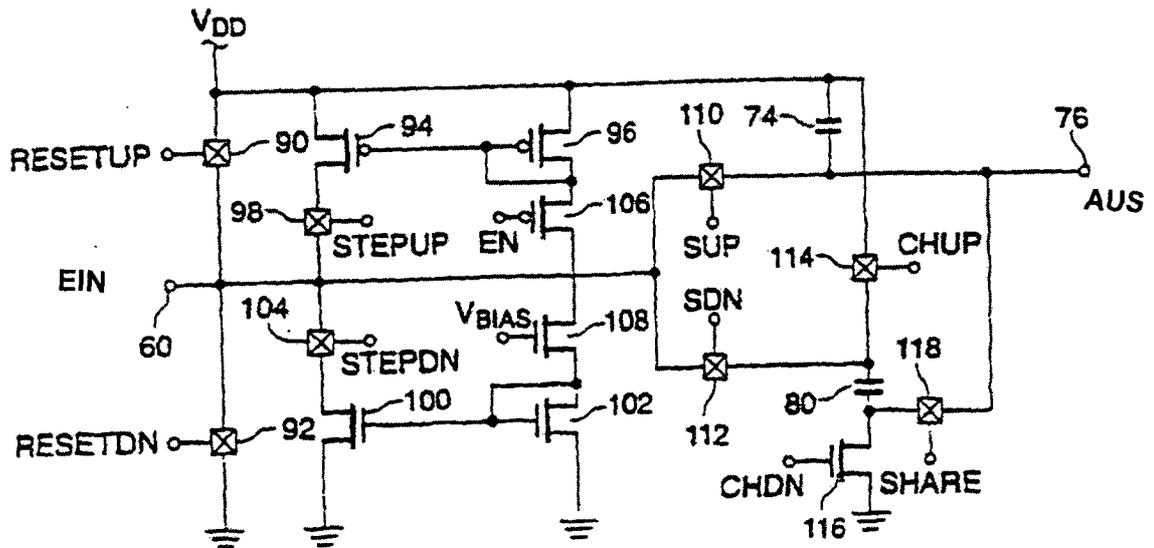


FIG. 4B

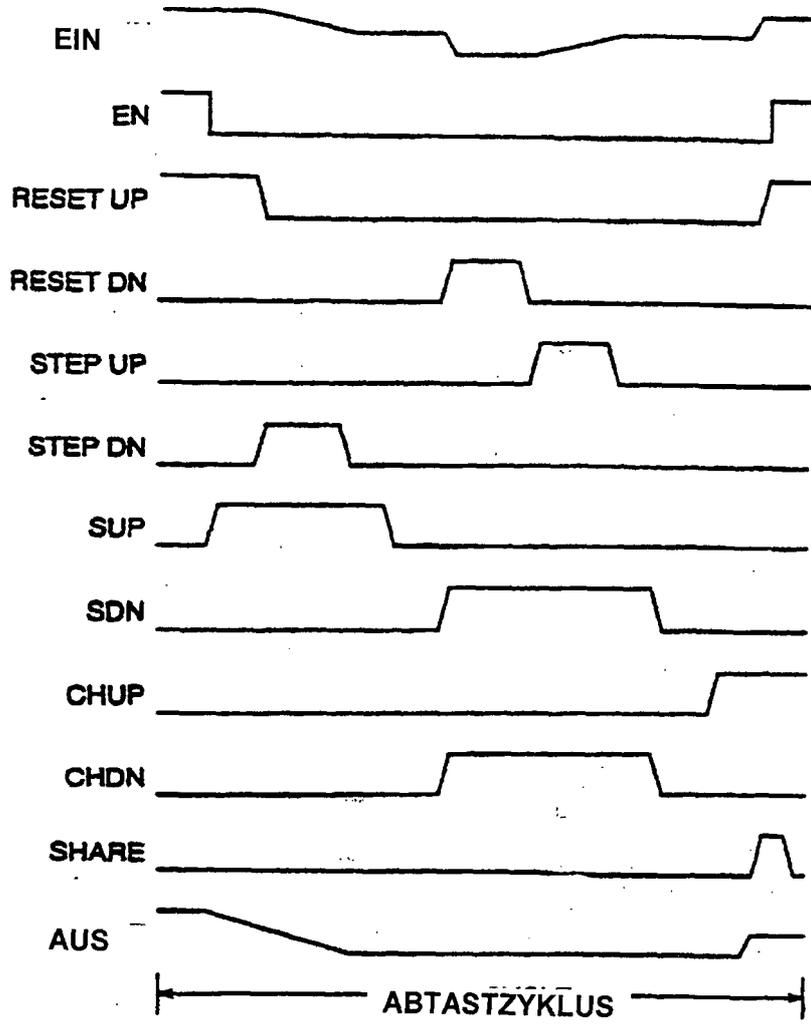


FIG. 5

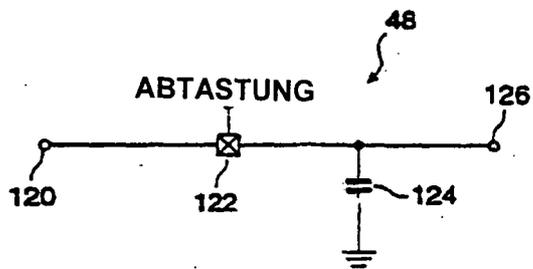


FIG. 6

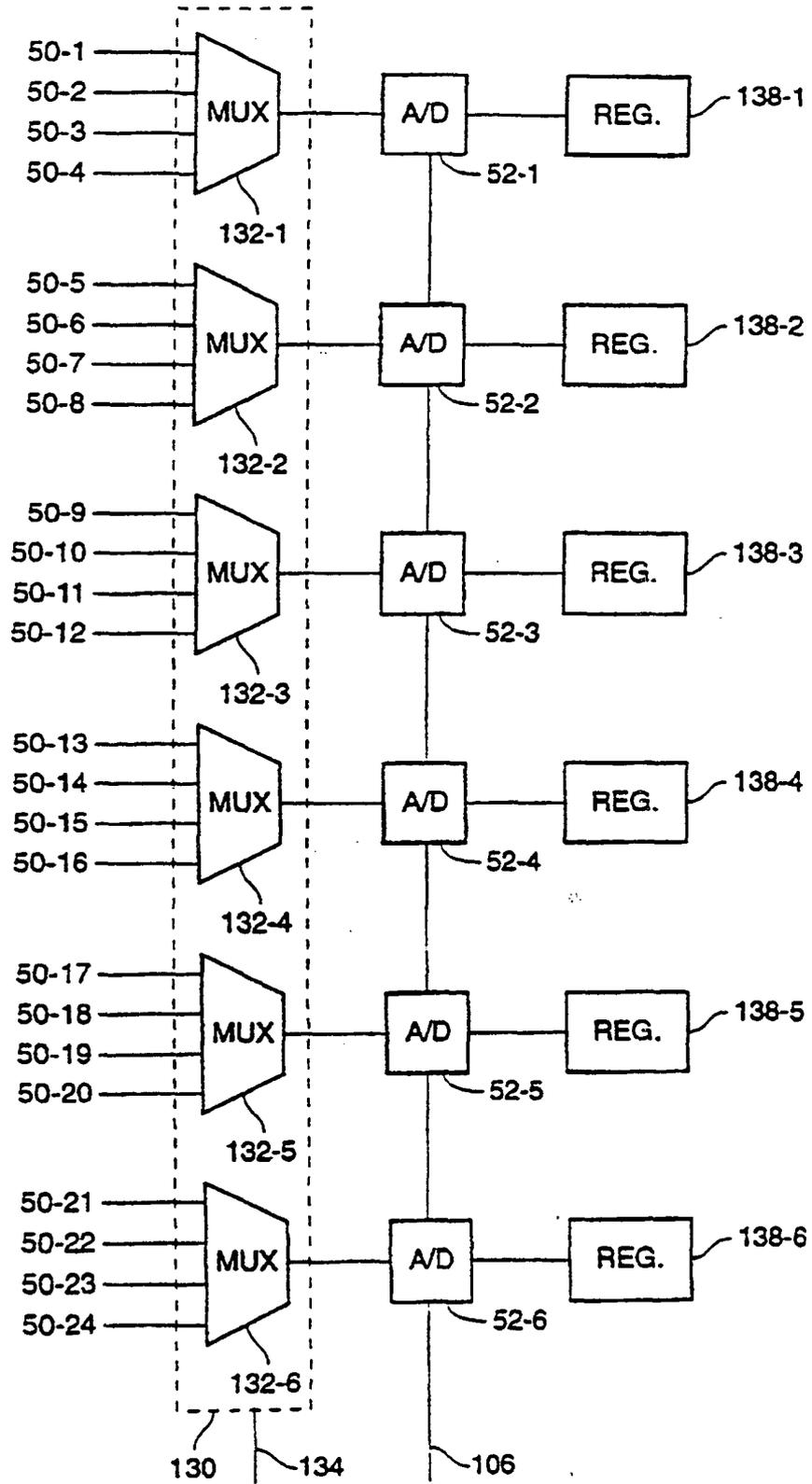
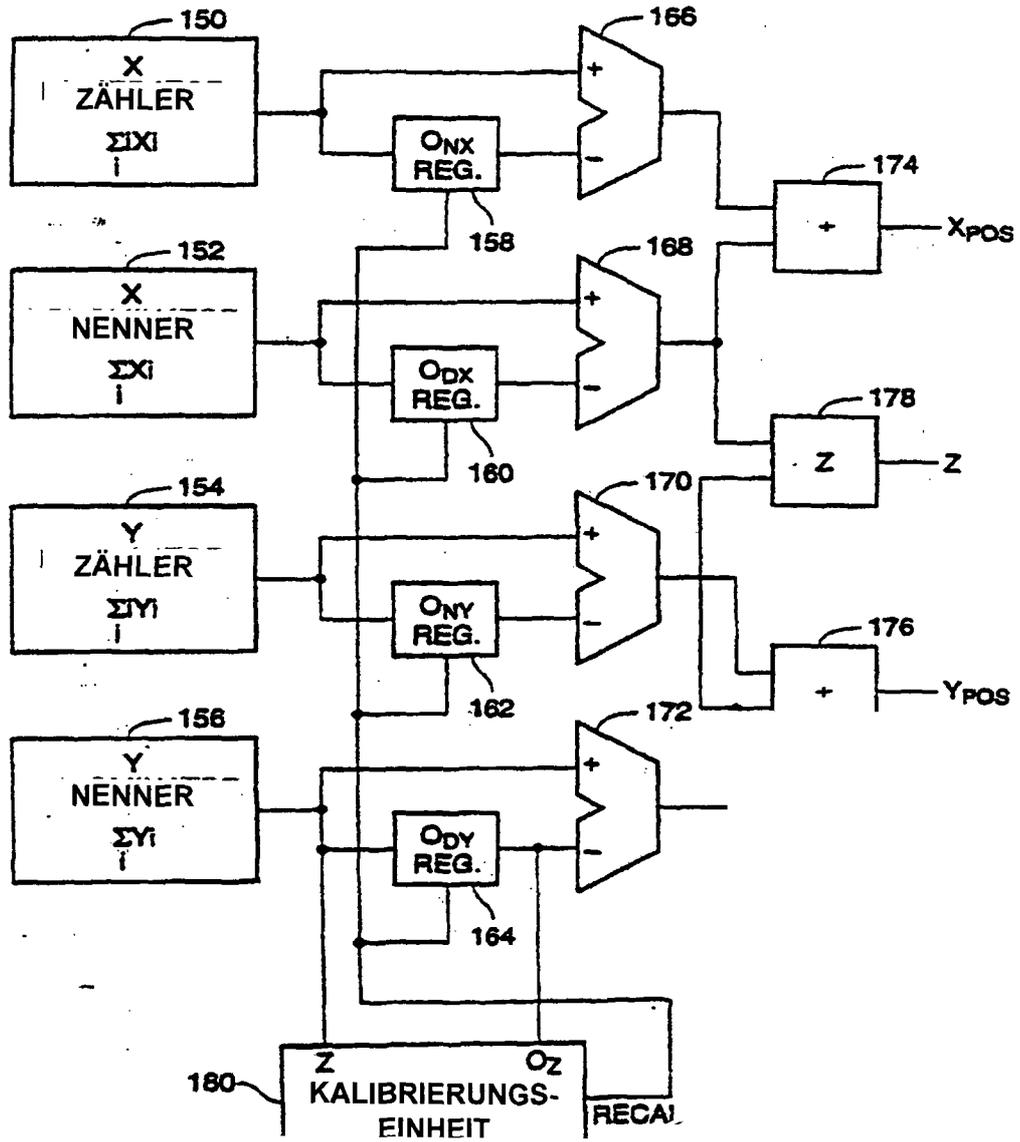


FIG. 7



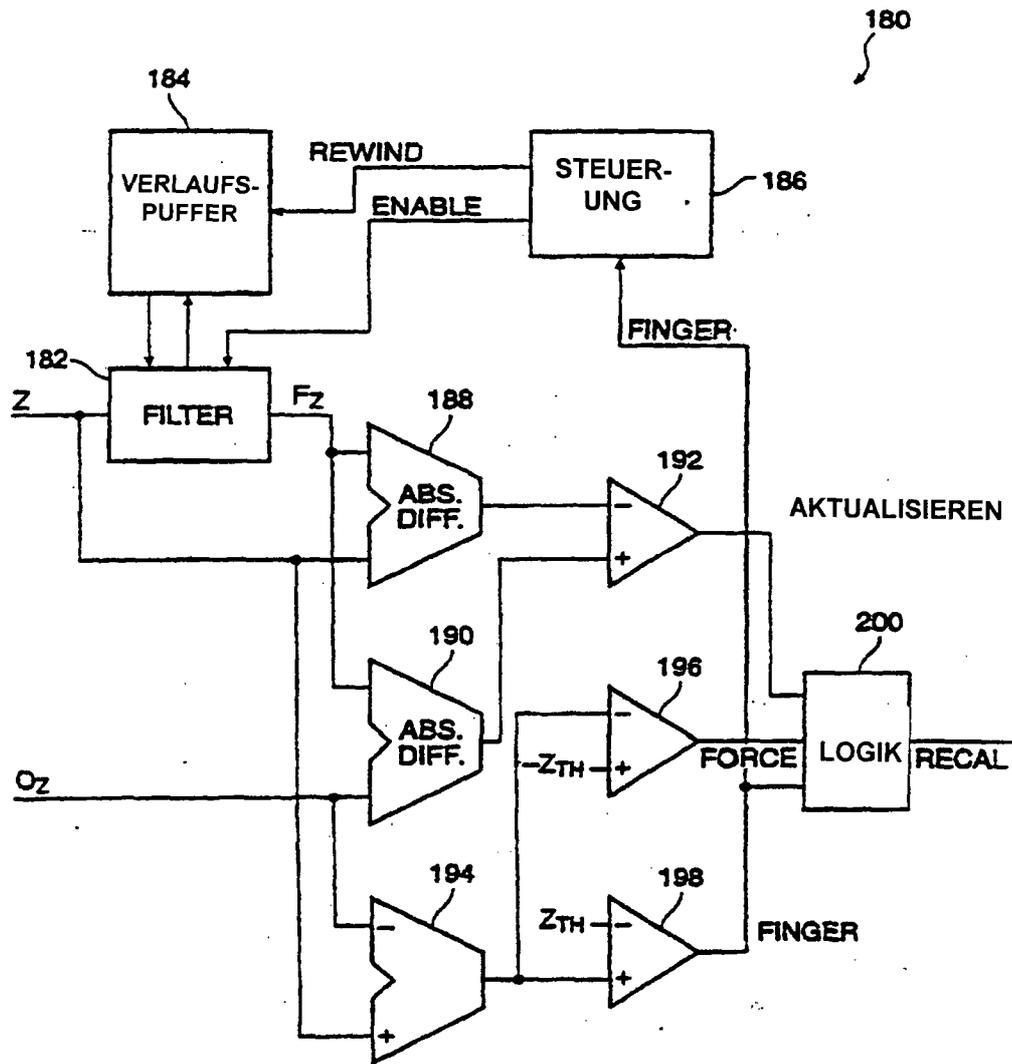


FIG. 9

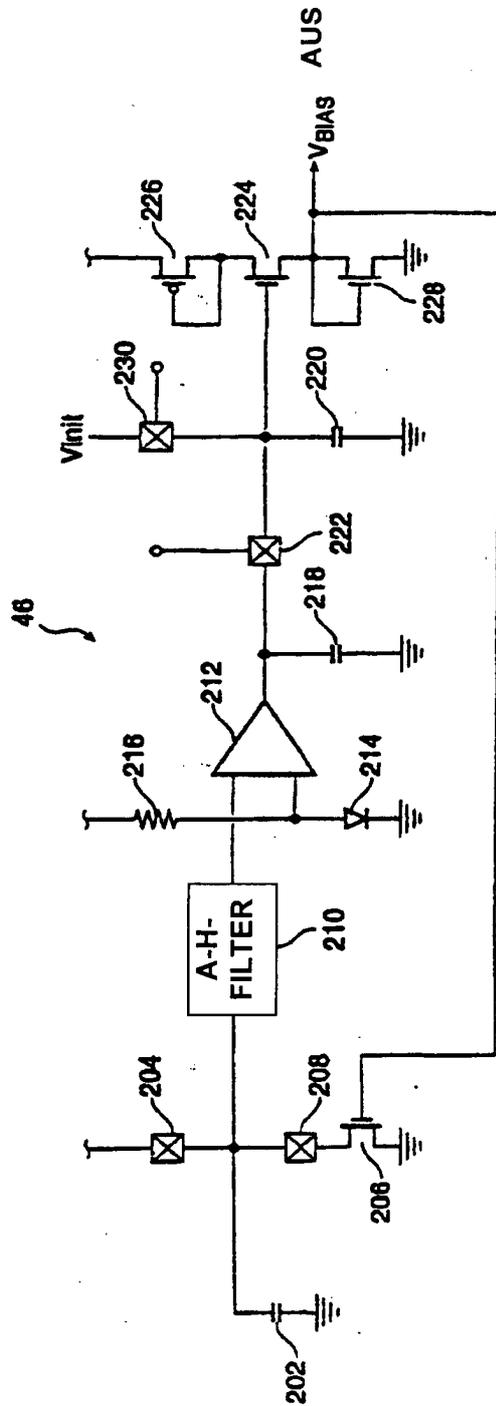
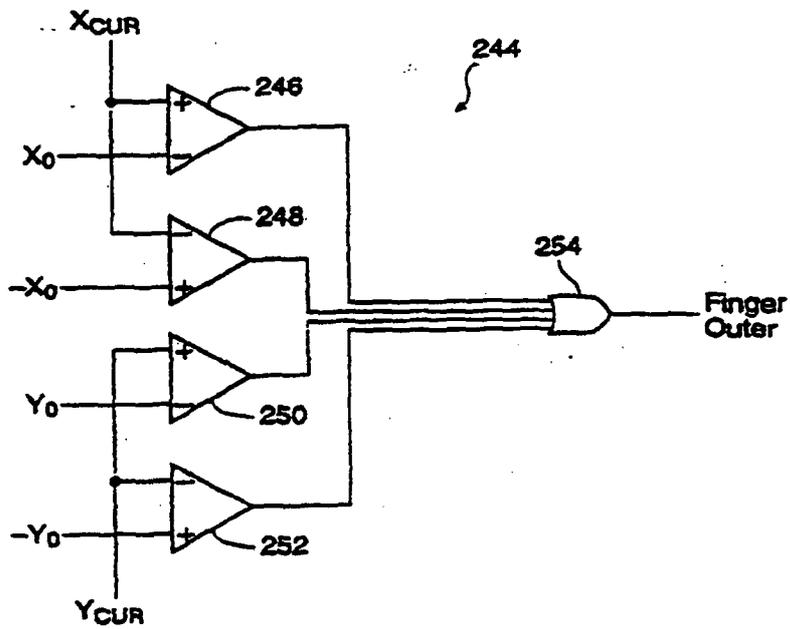
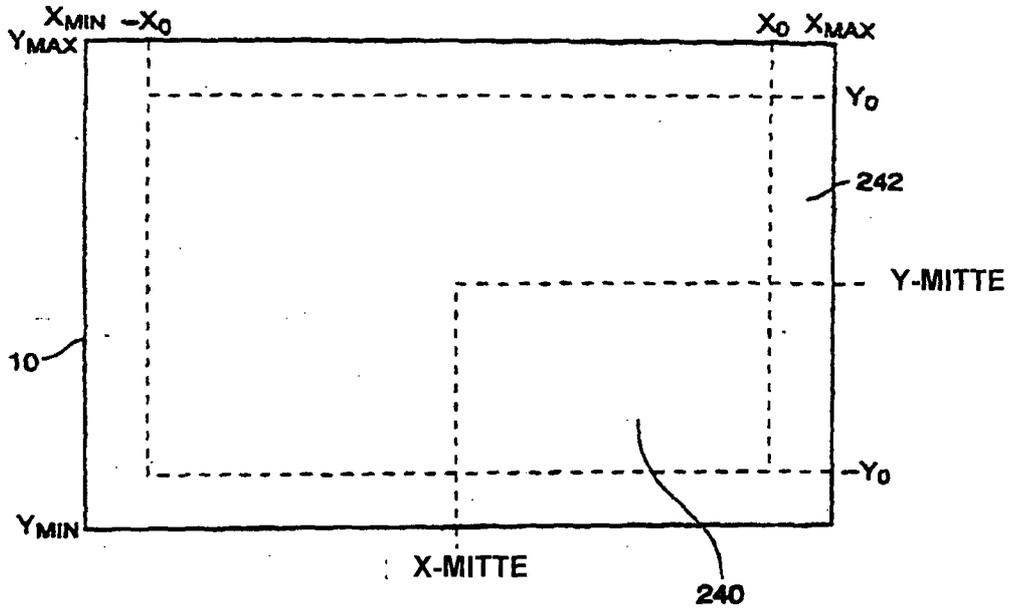


FIG. 10



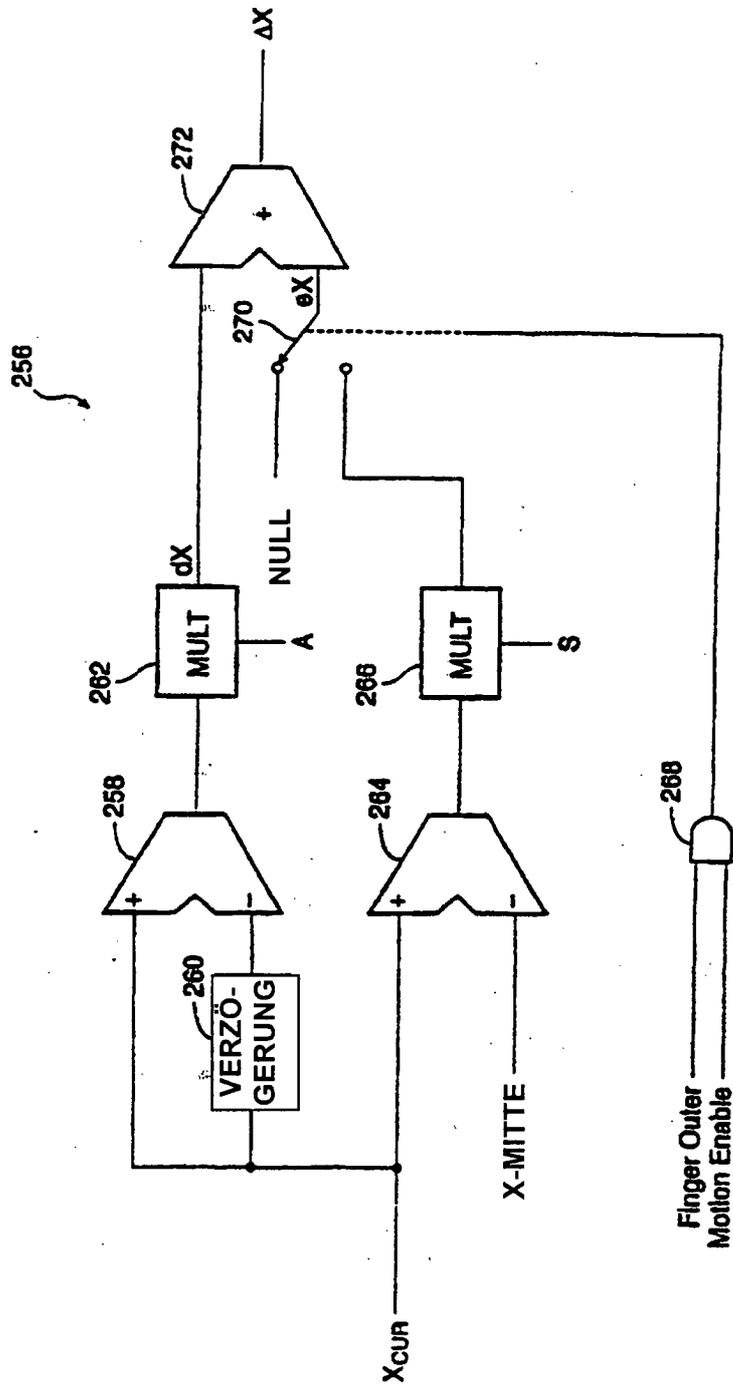


FIG. 13

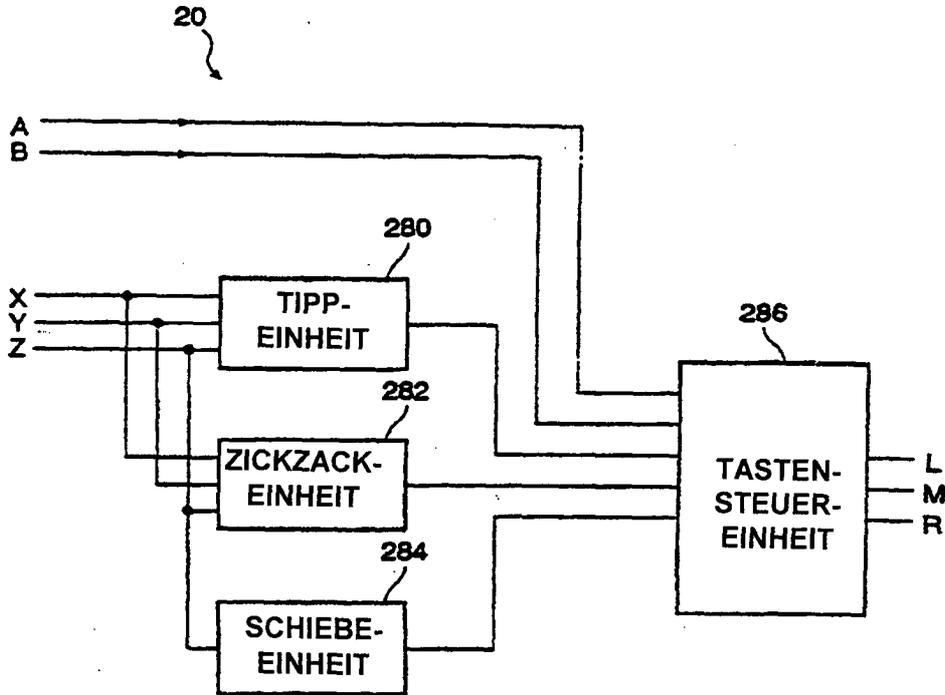


FIG. 14

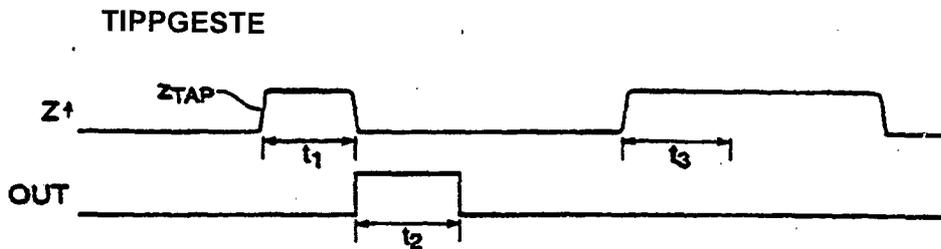


FIG. 15A

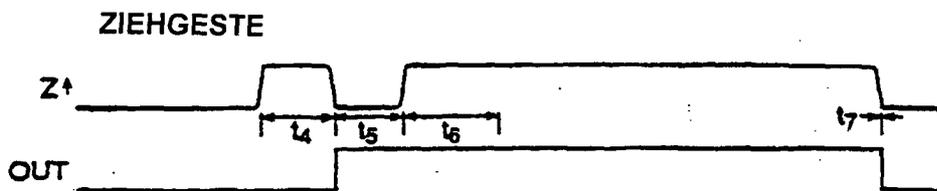
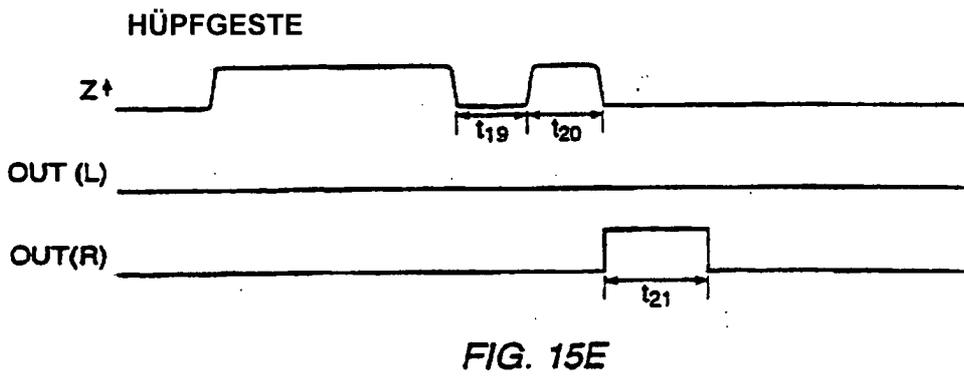
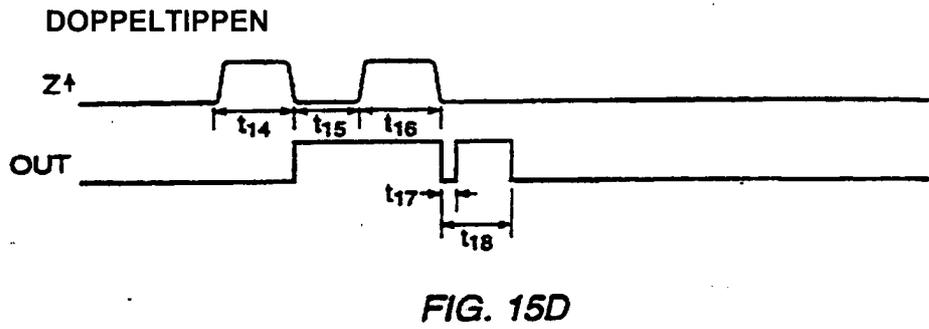
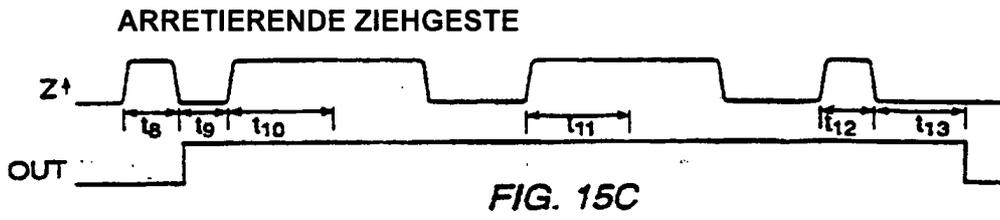


FIG. 15B



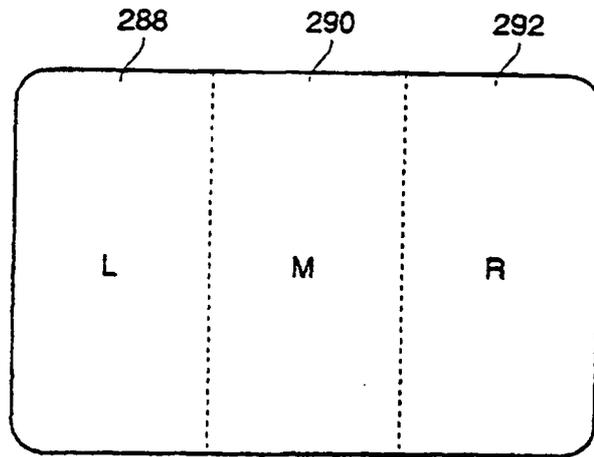


FIG. 16A

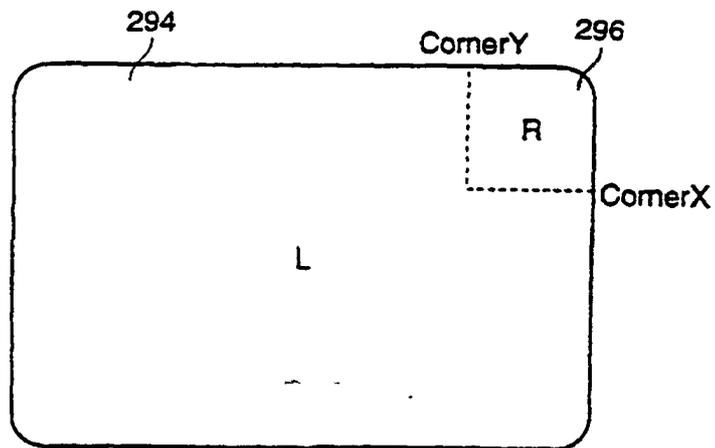


FIG. 16B

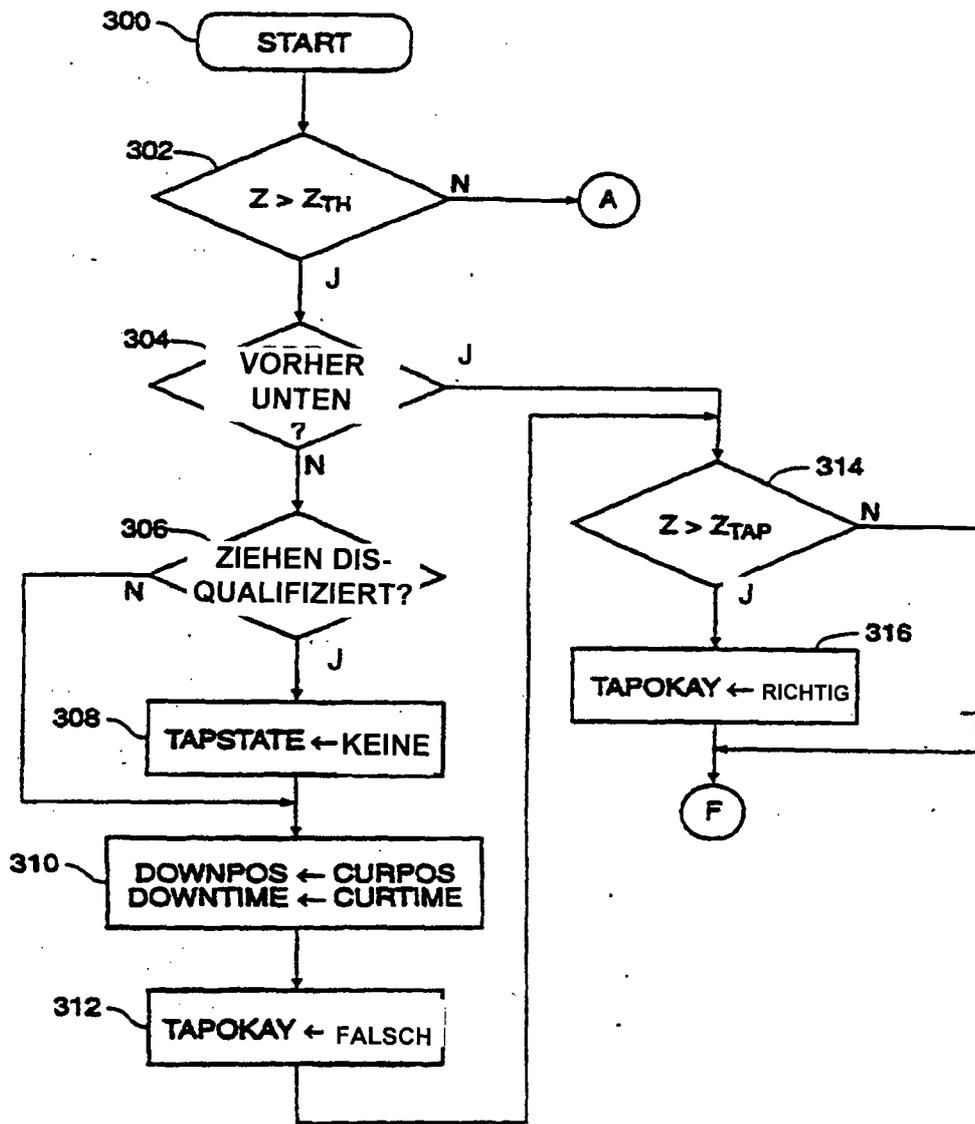


FIG. 17A

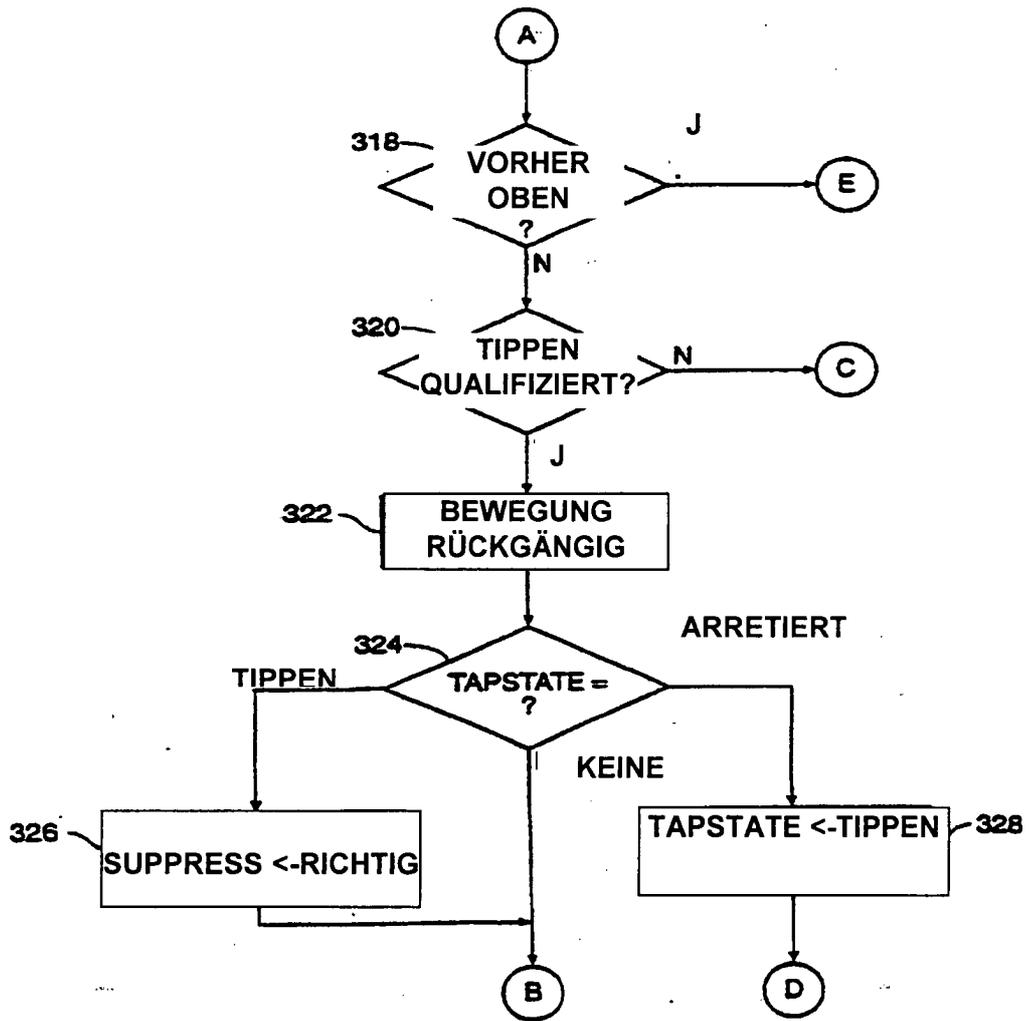


FIG. 17B

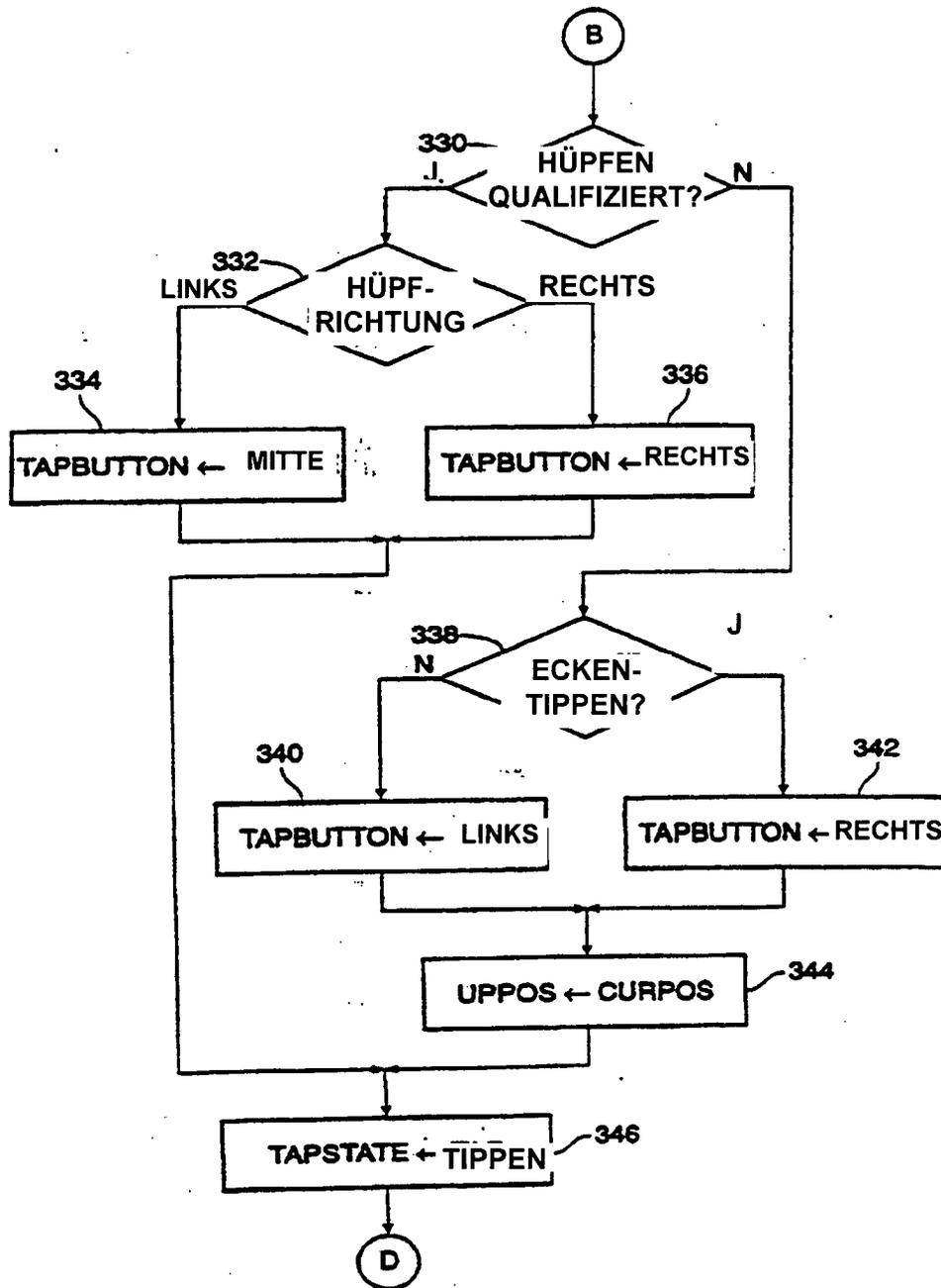


FIG. 17C

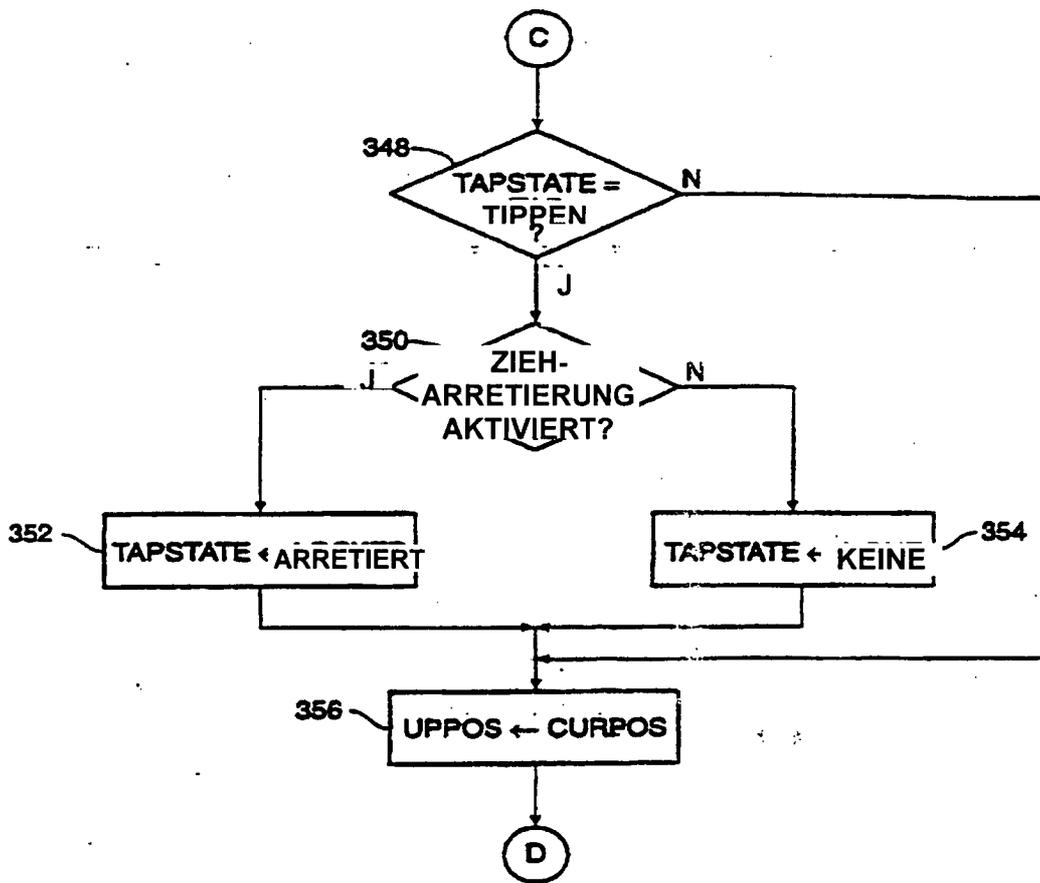


FIG. 17D

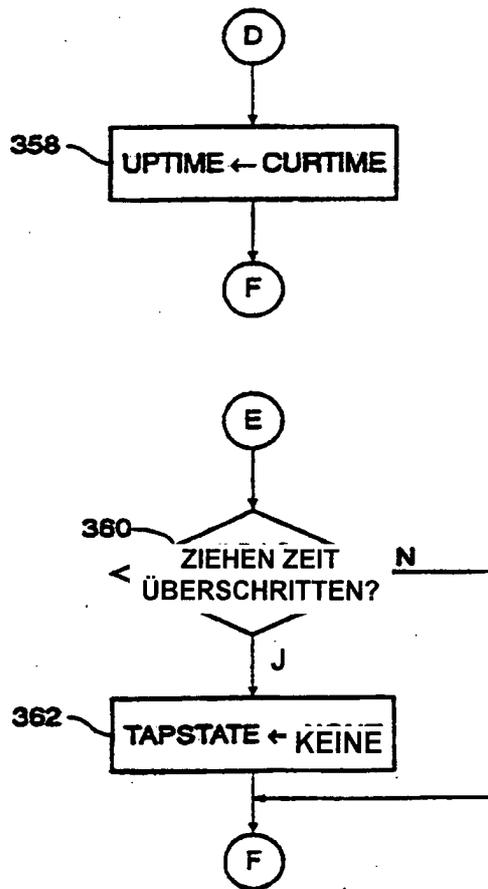


FIG. 17E

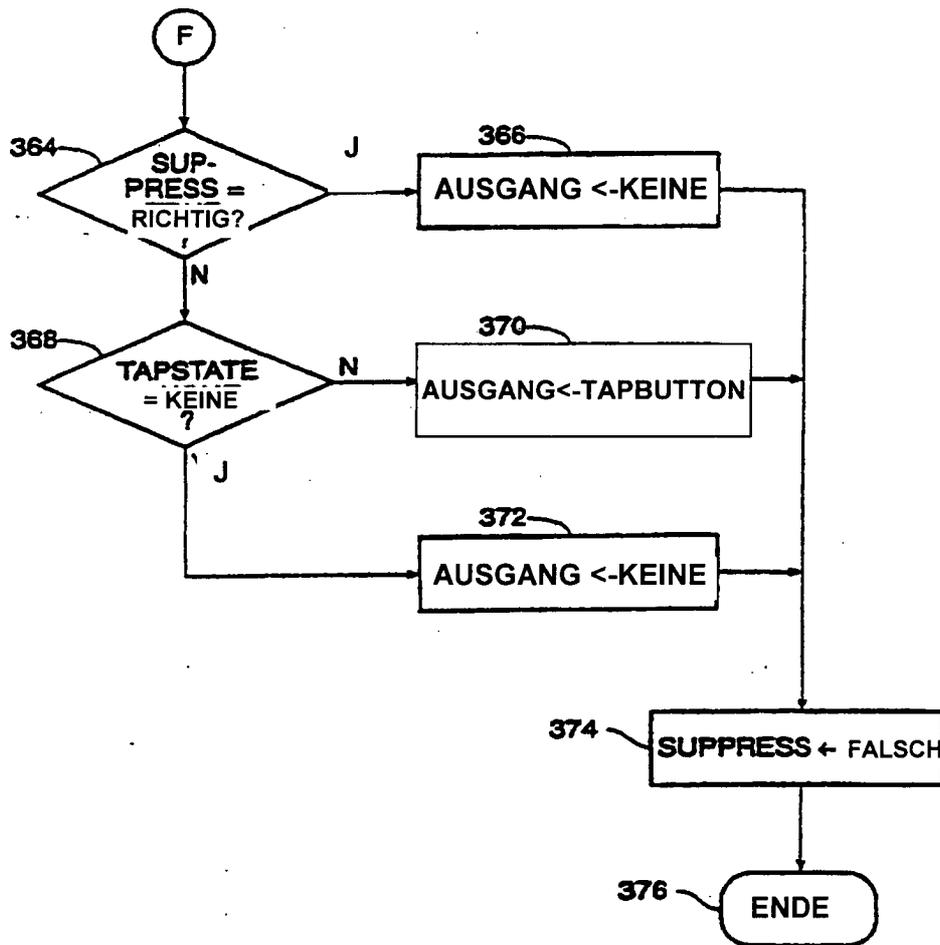


FIG. 17F

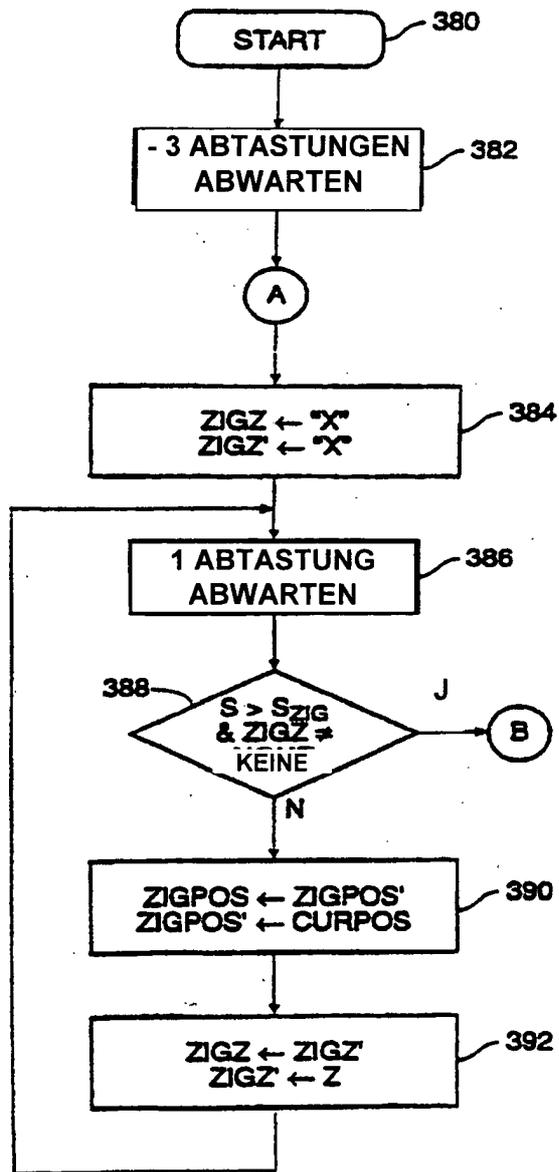


FIG. 18A

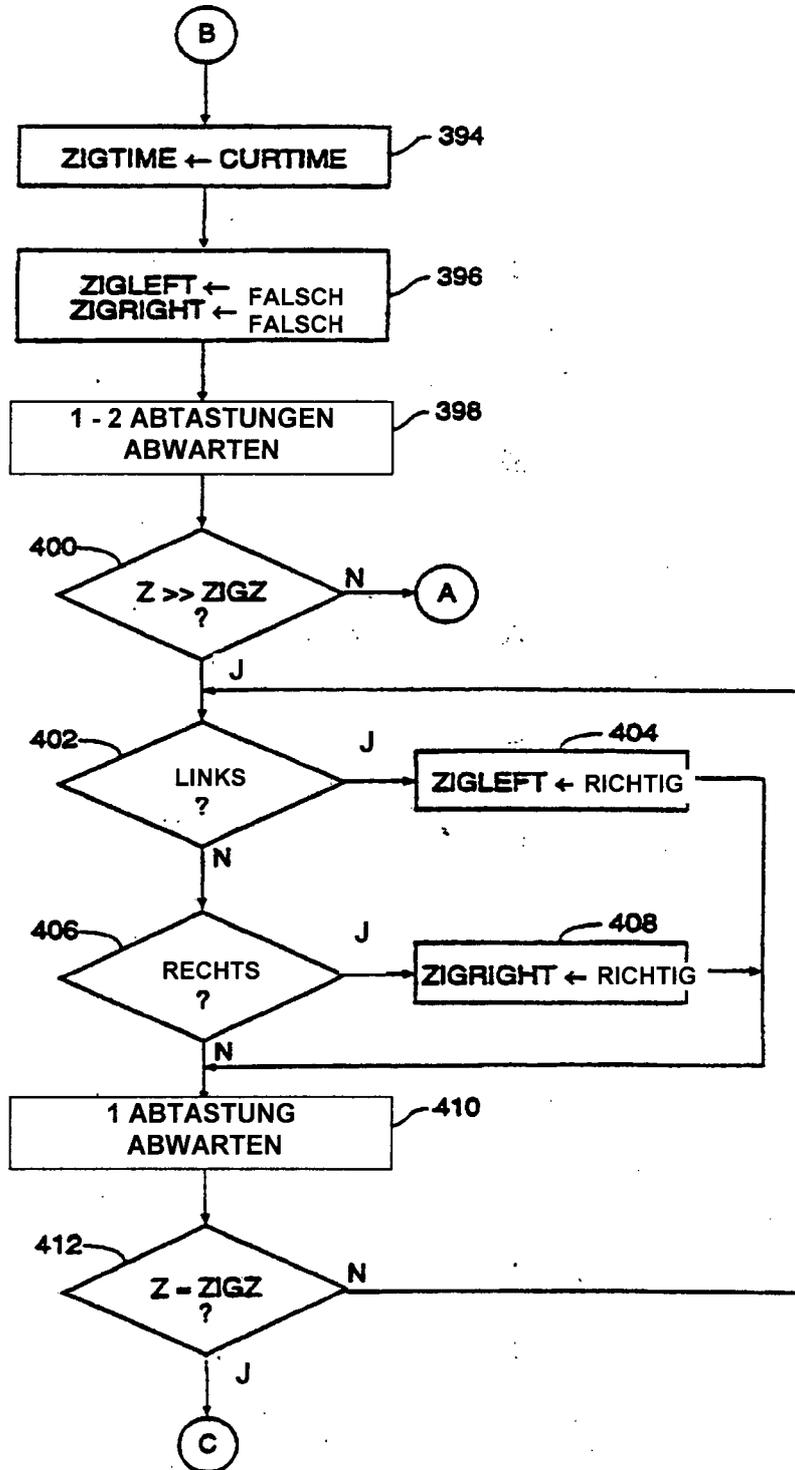


FIG. 18B

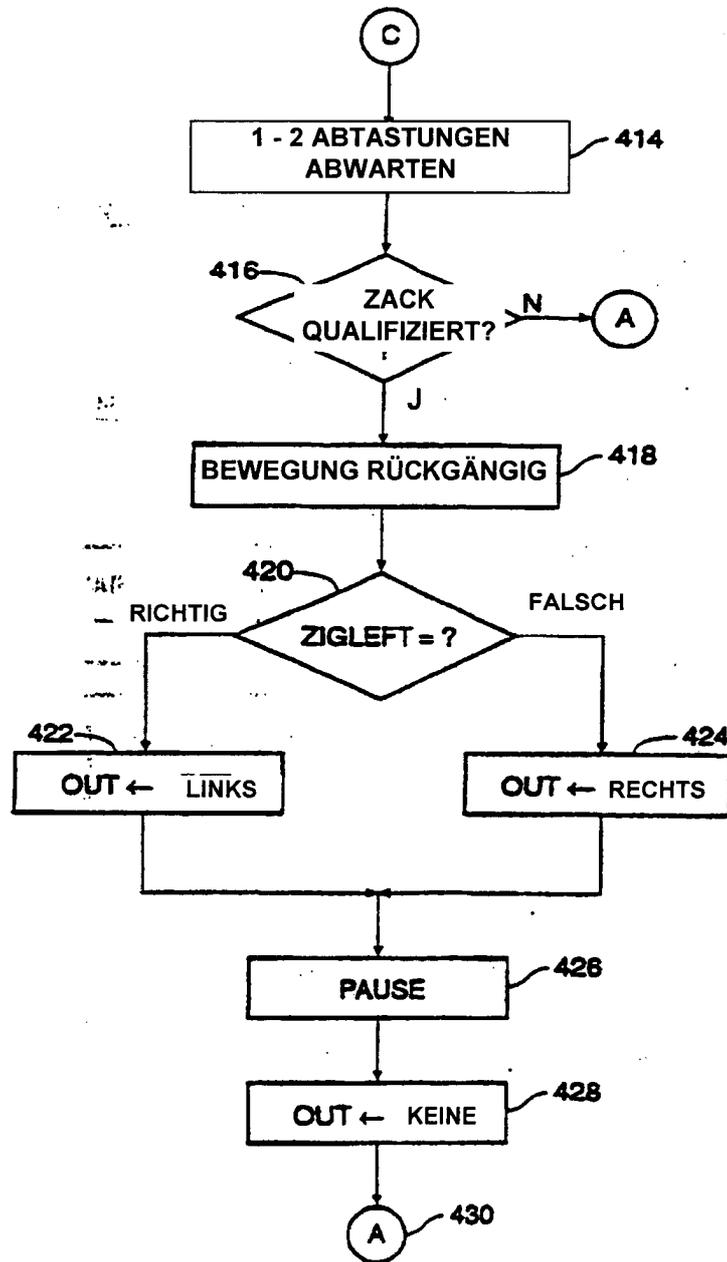


FIG. 18C

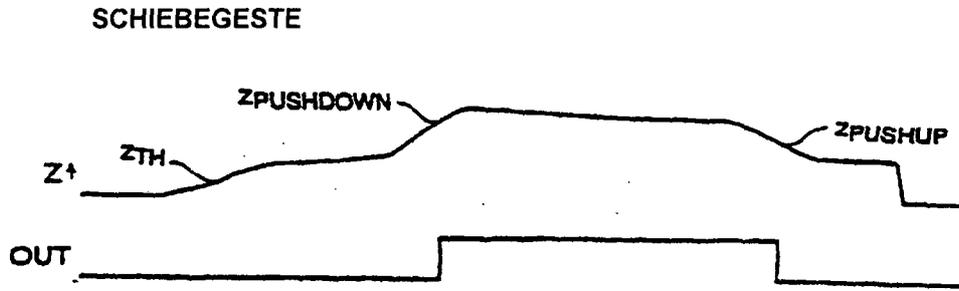


FIG. 19

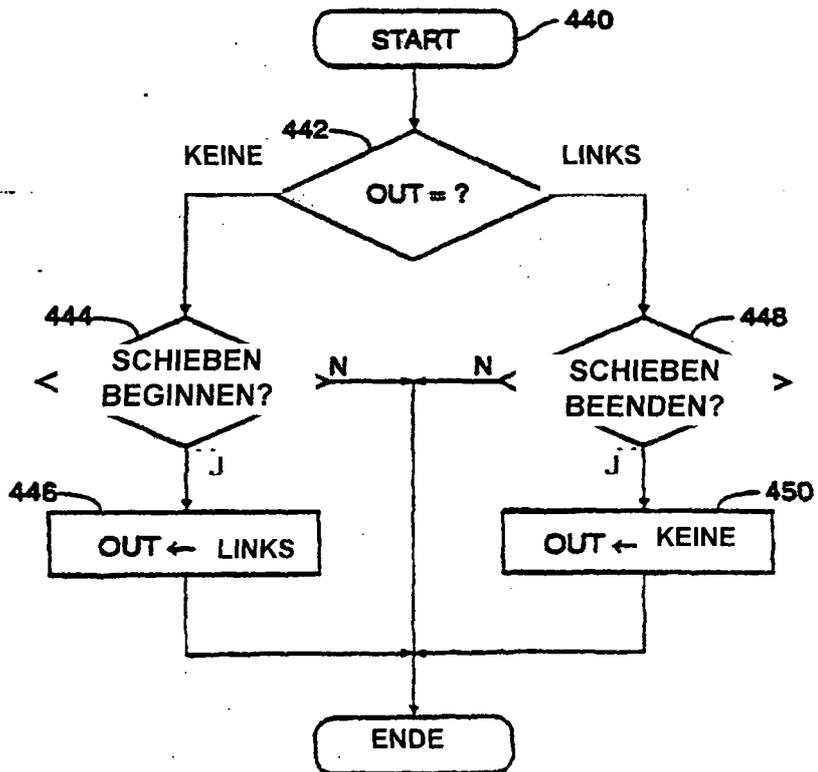


FIG. 20

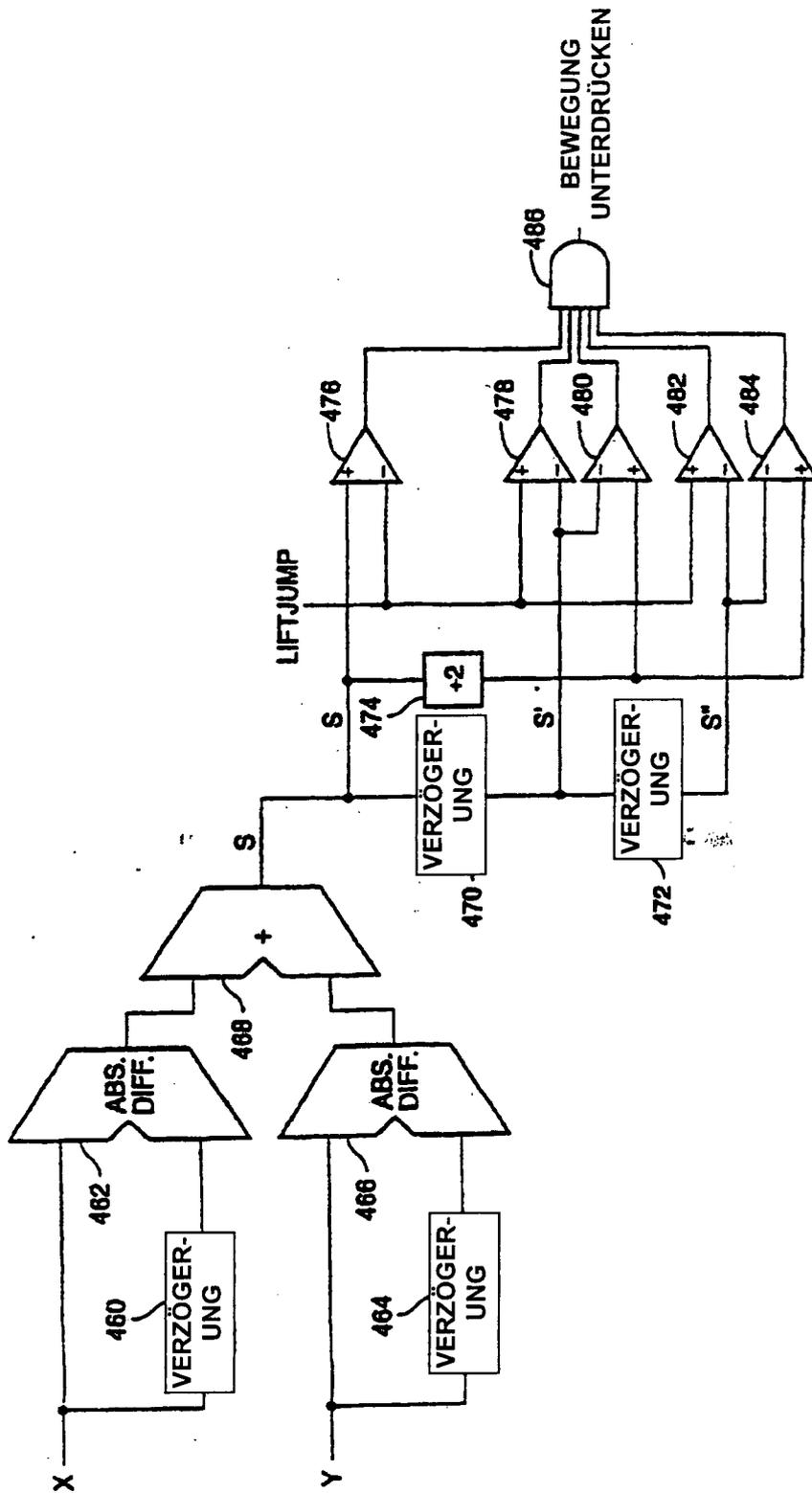


FIG. 21