

(19)



Europäisches Patentamt

European Patent Office

Office européen des brevets



(11)

EP 1 286 331 B1

(12)

EUROPÄISCHE PATENTSCHRIFT

(45) Veröffentlichungstag und Bekanntmachung des
Hinweises auf die Patenterteilung:
24.11.2004 Patentblatt 2004/48

(51) Int Cl.7: **G10L 19/12**

(21) Anmeldenummer: **02102146.4**

(22) Anmeldetag: **16.08.2002**

(54) **Verfahren für die algebraische Codebook-Suche eines Sprachsignalkodierers**

Method for algebraic codebook search for a speech signal coder

Procédé de recherche algébrique d'une table de codes pour un codeur de paroles

(84) Benannte Vertragsstaaten:
**AT BE BG CH CY CZ DE DK EE ES FI FR GB GR
IE IT LI LU MC NL PT SE SK TR**

(30) Priorität: **17.08.2001 DE 10140507**

(43) Veröffentlichungstag der Anmeldung:
26.02.2003 Patentblatt 2003/09

(73) Patentinhaber:
• **Philips Intellectual Property & Standards GmbH
20099 Hamburg (DE)**
Benannte Vertragsstaaten:
DE
• **Koninklijke Philips Electronics N.V.
5621 BA Eindhoven (NL)**
Benannte Vertragsstaaten:
**AT BE BG CH CY CZ DK EE ES FI FR GB GR IE
IT LU MC NL PT SE SK TR LI**

(72) Erfinder: **Gradl, Dietmar,
c/o Philips Corporate Intellectual
52088 Aachen (DE)**

(74) Vertreter: **Volmer, Georg, Dipl.-Ing. et al
Philips Intellectual Property & Standards GmbH,
Postfach 50 04 42
52088 Aachen (DE)**

(56) Entgegenhaltungen:
US-A- 5 717 825 US-A- 5 924 062

- **SALAMI R ET AL: "ITU-T G.729 ANNEX A:
REDUCED COMPLEXITY 8 KB/S CS-ACELP
CODES FOR DIGITAL SIMULTANEOUS VOICE
AND DATA" IEEE COMMUNICATIONS
MAGAZINE, IEEE SERVICE CENTER.
PISCATAWAY, N.J, US, Bd. 35, Nr. 9, 1.
September 1997 (1997-09-01), Seiten 56-63,
XP000704424 ISSN: 0163-6804**
- **GUYADER LE A ET AL: "ROBUST AND FAST
CODE-EXCITED LINEAR PREDICTIVE CODING
OF SPEECH SIGNALS" SPEECH PROCESSING
1. GLASGOW, MAY 23 - 26, 1989,
INTERNATIONAL CONFERENCE ON
ACOUSTICS, SPEECH & SIGNAL PROCESSING.
ICASSP, NEW YORK, IEEE, US, Bd. 1 CONF. 14,
23. Mai 1989 (1989-05-23), Seiten 120-123,
XP000089686**

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).

EP 1 286 331 B1

Beschreibung

[0001] Die Erfindung betrifft ein Verfahren für die algebraische Codebook-Suche eines Sprachsignalkodierers, vorzugsweise nach dem Code Excited Linear Prediction-Verfahren, bei dem zur Berechnung von Koeffizienten der Dreiecksmatrix der Autokorrelationsmatrix vom Toeplitz-Typ ein n Sprachsignalabtastungen umfassender Zeitabschnitt in eine ganzzahlige Anzahl Tracks t mit je p möglichen Pulspositionen zerlegt wird. Die Erfindung bezieht sich ebenfalls auf eine Kommunikationseinrichtung, insbesondere ein Mobiltelefon, mit einem Sprachsignalkodierer.

[0002] Derartige Verfahren kommen bei digitalen Sprachübertragungsverfahren zum Einsatz. Wird ein analoges Sprachsignal mit einer bestimmten Abtastrate in ein digitales Signal umgewandelt, so entsteht eine sehr große Datenmenge, die über einen im Durchsatz begrenzten Funkkanal nicht vollständig übertragen werden kann. Aus diesem Grund muss nach der Digitalisierung des Sprachsignals eine Kompression der Signale vorgenommen werden. Ein Signal wird komprimiert, indem man nichtrelevante Elemente weglässt, sich wiederholende Elemente mit einer Kurzbezeichnung versieht und nur noch diese Kurzbezeichnungen als Kodierungen überträgt. Auf dem Gebiet der Kodierverfahren für Mobilfunkanwendungen hat das CELP-Verfahren (Code Excited Linear Prediction) besondere Bedeutung erlangt. Bei diesem effizienten Kodierverfahren werden als Koeffizienten in einer Autokorrelationsmatrix hinterlegte Klangelemente erkannt und übertragen. Die Autokorrelationsmatrix kann dabei mit einem Notizbuch oder Codebook verglichen werden, von dem nur die Notizbuchadresse übertragen wird. Der Empfänger benötigt zwangsläufig das gleiche Notizbuch, um das empfangene Digitalsignal in ein dem ursprünglichen Signal möglichst ähnliches, analoges Sprachsignal umzuwandeln.

[0003] Eine Reihe von Kodierern/Dekodierern sind von der ITU international standardisiert, zu diesen gehören die Verfahren CS-ACELP und ACELP, die mit Bitraten von bis zu 8 kbps arbeiten.

[0004] Bei dem CELP-Verfahren wird zunächst eine LPC-Analyse (linear prediction coefficient) vorgenommen. Das verbleibende Signal wird anschließend durch einen Suchvorgang in einem adaptiven Codebook quantifiziert. Auf diese Weise werden periodische Anteile des Sprachsignals in einer LTP-Analyse (long term prediction) herausgefiltert. Das verbleibende Signal wird in einem zweiten Codebook quantifiziert, für dieses Verfahren gibt es bereits eine Reihe von Lösungen. Bei dem AMR-Verfahren (adaptive multirate speech codec) wird ein algebraisches Codebook benutzt. Das Prinzip der algebraischen Codebook-Suche beruht darauf, einen Codevektor zu suchen, der einen bestimmten Zeitabschnitt repräsentiert und bei dem eine begrenzte Anzahl von Pulsen eine Amplitude von +1 oder -1 besitzt. Dieser Codevektor wird durch einen Synthesefilter gefil-

tert, das heißt auf der Seite des Senders wird der Dekodiervorgang durchgeführt, der nach der Übermittlung des Signals auf der Empfängerseite vorgenommen wird. Eine sehr große Anzahl von möglichen Codevektoren wird durch geschachtelte Suchschleifen systematisch geprüft um denjenigen Codevektor zu bestimmen, der die geringste Fehlerenergie aufweist, d.h. der dem Ursprungssignal möglichst ähnlich ist. Diese iterative Bestimmung des Codevektors beansprucht den größten Teil der Berechnungskapazität eines Mobiltelefons, sodass eine Optimierung dieses Suchalgorithmus besonders effizient ist. Einerseits ist es wünschenswert, die Anzahl der benötigten Speicherstellen zu reduzieren, da die dafür benötigten RAM-Bausteine relativ teuer sind, andererseits wird auch angestrebt, die benötigte Anzahl der Rechenoperationen des Suchalgorithmus zu verringern. Ein Verfahren zur Reduzierung der benötigten Speicherplatzgröße wird z.B. in US-A-5717825 offenbart.

[0005] Die Autokorrelationsmatrix ist eine Toeplitz-Matrix, das heißt sie ist symmetrisch bezüglich ihrer Hauptdiagonale und ihre obere oder die damit identische untere Dreiecksmatrix enthält sämtliche Koeffizienten. Es ist daher bereits vorgeschlagen worden, anstelle der vollständigen Autokorrelationsmatrix lediglich eine der Dreiecksmatrizen zu speichern, um Speicherplatz einzusparen. Dieses Verfahren führt jedoch zu einer komplizierteren Adressierung der einzelnen Koeffizienten, sodass der Einsparung von Speicherplatz eine Erhöhung des Rechenaufwands gegenübersteht.

[0006] Der Erfindung liegt daher das Problem zugrunde, ein Verfahren anzugeben, bei dem der Speicherplatzbedarf und der Berechnungsaufwand verringert sind.

[0007] Zur Lösung dieses Problems ist bei einem Verfahren der eingangs genannten Art erfindungsgemäß vorgesehen, dass die Koeffizienten gruppiert nach Kombinationen benachbarter Tracks; Kombinationen nicht benachbarter Tracks; Kombinationen gleicher Tracks; und Koeffizienten der Hauptdiagonale der Autokorrelationsmatrix in einem Speicher abgelegt werden.

[0008] Bei dem erfindungsgemäßen Verfahren werden die benötigten Koeffizienten der Autokorrelationsmatrix in einer Weise gespeichert, die einen schnellen, sequentiellen Zugriff erlaubt. Die ansonsten zusätzlich erforderliche, relativ aufwändige Berechnung der Speicheradressen der Koeffizienten der Dreiecksmatrix kann wesentlich vereinfacht werden. Einige Koeffizienten werden sehr häufig benötigt, andere dagegen sehr selten. Dieser Umstand wird bei der optimierten Gruppierung der ausgenutzt, sodass die häufig benötigten Koeffizienten der Autokorrelationsmatrix einfacher adressiert werden können, was in einem sehr schnellen Zugriff resultiert.

[0009] Die Erfindung sieht vor, dass für die Gruppen der Kombinationen benachbarter und nicht benachbarter Tracks jeweils t Datensätze mit je $p \times p$ Koeffizienten

gespeichert werden. Eine in der Praxis sehr wichtige Betriebsart des CELP- bzw. ACELP-Verfahrens sieht vor, dass die Positionen zweier benachbarter Pulse gleichzeitig festgelegt werden, so dass sich bei p möglichen Pulspositionen pro Codevektor $p \times p$ Durchläufe der Suchschleife ergeben.

[0010] Ein äußerst schneller und einfacher Zugriff auf die in der Suchschleife benötigten Koeffizienten lässt sich erzielen, wenn die Koeffizienten sequentiell in einem Speicher abgelegt werden.

[0011] In weiterer Ausgestaltung der Erfindung ist vorgesehen, dass eine einen horizontalen oder vertikalen Vektor der Autokorrelationsmatrix darstellende Untergruppe eines Datensatzes mit p Koeffizienten durch eine Programmschleife ausgelesen wird, wobei ein die Speicherstelle des ersten Koeffizienten bezeichnender Wert und eine konstante Schrittweite bis zur nächsten Speicherstelle vorgegeben werden. Demnach genügt es, einen Anfangs- oder Startwert für die erste Speicheradresse und die Schrittweite, d.h. die Anzahl der Speicherstellen bis zur jeweils nächsten Speicherstelle festzulegen. Es kann vorgesehen sein, dass die Startwerte einer in einem Festspeicher abgelegten Nachschlagetabelle entnommen werden, alternativ werden sie jeweils berechnet.

[0012] Zweckmäßig wird für die Datensätze der Gruppe der Kombinationen benachbarter Tracks die Schrittweite eins gewählt. Die Koeffizienten werden sequentiell gespeichert und lassen sich besonders einfach auslesen.

[0013] Für die Datensätze der Gruppe der Kombinationen nicht benachbarter Tracks empfiehlt es sich, die Schrittweite p zu wählen.

[0014] Zur Verringerung des Speicherplatzbedarfs können für die Gruppe der Kombinationen gleicher Tracks t Dreiecksmatrizen sequentiell gespeichert werden. Jeder Kombination gleicher Tracks entspricht dabei eine Dreiecksmatrix und alle t Dreiecksmatrizen werden in einem Block gespeichert. Da diese Koeffizienten verhältnismäßig selten benötigt werden ist es kein Nachteil, wenn der Zugriff etwas aufwändiger ist. Um den Rechenaufwand weiter zu reduzieren, kann der Zugriff auch über eine Nachschlagetabelle erfolgen.

[0015] Die Koeffizienten der Hauptdiagonale werden in einer Gruppe zusammengefasst und sequentiell gespeichert.

[0016] Es hat sich als günstig herausgestellt, wenn innerhalb eines Zeitabschnitts 40 Sprachsignalabtastungen erfasst werden. Wenn dieser Wert gewählt wird ist das Verfahren mit den international festgelegten Regelungen kompatibel. Bei einer typischen Abtastrate von 8 kHz für das Sprachsignal dauert ein Zeitabschnitt 20 msec, innerhalb dieses kurzen Zeitabschnittes kann das Sprachsignal als quasistationär angesehen und durch einen Codevektor wiedergegeben werden.

[0017] Die Autokorrelationsmatrix ist vorzugsweise eine 40×40 -Matrix, entsprechend den 40 Sprachsignalabtastungen in einem Zeitfenster.

[0018] Zur Verringerung der Anzahl der Iterationen ist bei dem erfindungsgemäßen Verfahren vorgesehen, dass ein Zeitabschnitt in eine ganzzahlige Anzahl gleichlanger Tracks zerlegt wird. Bevorzugt wird dabei die Zerlegung eines Zeitabschnitts in 5 Tracks mit je 8 Pulspositionen oder die Zerlegung in 4 Tracks mit je 10 Pulspositionen.

[0019] Ein besonders schneller Zugriff auf die Koeffizienten wird ermöglicht, wenn die Koeffizientengruppen der Kombinationen benachbarter und nicht benachbarter Tracks aus einer Mehrzahl von je 64 Koeffizienten umfassenden Blöcken gebildet werden. Auf diese Koeffizientengruppen muss während der Iteration besonders oft zurückgegriffen werden. Diese Gruppen werden daher in der Reihenfolge gespeichert, in der sie für die Berechnung benötigt werden, sodass auf sie schnell zugegriffen werden kann, was zu einer Verringerung des Rechenaufwands führt.

[0020] Besonders gute Ergebnisse lassen sich erzielen, wenn für die Koeffizientengruppe der Kombinationen benachbarter Tracks 320 Werte ermittelt werden. Für die Koeffizientengruppe der Kombinationen nicht benachbarter Tracks werden zweckmäßig ebenfalls 320 Werte ermittelt. Die Koeffizientengruppe der Kombinationen gleicher Tracks enthält 140 Werte, zusammen mit den Koeffizienten der Hauptdiagonale werden insgesamt 820 Koeffizienten ermittelt.

[0021] Eine weitere Steigerung der Berechnungsgeschwindigkeit lässt sich erzielen, wenn der Speicher mehrere RAM-Speicherbänke aufweist und Koeffizientengruppen in unterschiedlichen RAM-Speicherbänken gespeichert werden. Sind Koeffizientengruppen in unterschiedlichen RAM-Speicherbänken gespeichert, so kann parallel darauf zugegriffen werden, d.h. zwei Koeffizienten können gleichzeitig gelesen werden. Die Speicherzugriffszeit lässt sich dadurch annähernd halbieren.

[0022] Das erfindungsgemäße Verfahren kann mit besonderem Vorteil in das Betriebssystem eines Mobiltelefons integriert werden.

[0023] Weitere Vorteile und Einzelheiten der Erfindung ergeben sich aus der Beschreibung eines besonders geeigneten Ausführungsbeispiels und den Figuren. Die Figuren sind schematische Darstellungen und zeigen:

Fig. 1 die Zerlegung eines Zeitabschnitts in 4 Tracks mit je 10 möglichen Pulspositionen;

Fig. 2 eine Tabelle der zu testenden Track-/Puls-kombinationen;

Fig. 3 eine Tabelle der benachbarten und nicht benachbarten Tracks;

Fig. 4 eine Dreiecksmatrix mit Koeffizienten einer Kombination gleicher Tracks;

- Fig. 5 die Koeffizienten der Hauptdiagonale;
- Fig. 6 eine Gesamtübersicht aller zu berechnenden Koeffizienten;
- Fig. 7 die Berechnung der Gruppe der Kombinationen benachbarter Tracks (Block 1);
- Fig. 8 die Speicherreihenfolge von Block 1 nach dem ersten Schritt;
- Fig. 9 die Speicherreihenfolge von Block 1 nach dem zweiten Schritt;
- Fig. 10 die Berechnung der Gruppe der Kombinationen nicht benachbarter Tracks (Block 2);
- Fig. 11 die Speicherreihenfolge von Block 2 nach dem ersten Schritt;
- Fig. 12 die Speicherreihenfolge von Block 2 nach dem zweiten Schritt;
- Fig. 13 die Berechnung des Blocks mit den Werten gleicher Tracks (Block 3); und
- Fig. 14 die Speicherplatzreihenfolge von Block 3.

[0024] Durch einen iterativen Suchvorgang soll derjenige Codevektor ermittelt werden, der mit dem echten Signal am besten übereinstimmt, d.h. dessen Fehlerenergie minimal ist. Innerhalb des Suchvorgangs werden die Pulse nacheinander festgelegt, sodass sich die Anzahl der Variablen im Laufe der Suche verringert.

[0025] Die Tabelle von Fig. 1 zeigt die Zerlegung eines 40 Sprachsignalabtastungen umfassenden Zeitabschnitts in vier Tracks mit je zehn Pulspositionen. Eine andere Zerlegung, die in der Praxis von Bedeutung ist, sieht eine Zerlegung in fünf Tracks mit je acht möglichen Pulspositionen vor. Für jeden Puls ist festgelegt, in welchen Track er gesetzt werden kann. Der erste Puls kann daher nur an 10 (bzw. 8) Positionen gesetzt werden, anstatt an allen 40 Positionen. Iterativ wird diejenige Pulsposition ausgewählt, die den geringsten Energiefehler aufweist. Anschließend wird die nächste Pulsposition unter Berücksichtigung der bereits festgelegten ersten Pulsposition iterativ bestimmt. Dieses Verfahren wird für alle Pulse durchgeführt.

[0026] Bei bestimmten häufig vorkommenden Betriebsarten werden zwei benachbarte Pulse gleichzeitig bestimmt. Dazu werden alle Kombinationen zweier Pulse berechnet und das beste Pulspaar unter Berücksichtigung der bereits gesetzten Pulspaare bestimmt. Bei einer Betriebsart, bei der ein Track acht Pulspositionen umfasst, sind $8 \times 8 = 64$ Berechnungen erforderlich, bei einem Track mit 10 Pulspositionen müssen $10 \times 10 = 100$ Berechnungen für jedes Pulspaar durchgeführt werden. Das folgenden Beispiel bezieht sich auf das

Verfahren, bei dem ein Pulspaar gleichzeitig bestimmt wird.

[0027] Fig. 2 zeigt eine Tabelle der zu testenden Track-/Pulskombinationen für die Betriebsart, bei der acht Pulse gesetzt werden. Der erste Puls lp_0 wird in den Track gesetzt, in dem sich das Maximum des zurückgefilterten Zielsignals befindet. Diese Festlegung erfolgt vor der eigentlichen Suchschleife, sie gilt für die gesamte Suchschleife. In dem dargestellten Ausführungsbeispiel befand sich das Maximum des zurückgefilterten Zielsignals in Track 2. Daher wird dieser Wert für den Puls lp_0 bei allen Iterationen festgehalten. Der zweite Puls lp_1 wird bestimmt, indem alle 8 möglichen Pulspositionen eines Tracks bestimmt werden. Wie Fig. 2 entnommen werden kann, werden in Iteration 1 die 8 Positionen von Track 3 getestet. Diejenige Pulsposition von Track 3 mit dem geringsten Energiefehler wird ausgewählt. Nach der Festlegung von lp_0 und lp_1 werden die 64 möglichen Kombination der Pulse lp_2 und lp_3 getestet. Wie Fig. 2 entnommen werden kann, muss lp_2 für die erste Iteration in Track 4 und lp_3 in Track 0 gefunden werden. Anschließend werden die Pulspaare lp_4 - lp_5 und lp_6 - lp_7 nach dem gleichen Verfahren festgelegt. Nachdem alle Kombinationen geprüft sind, wird der Codevektor mit minimalem Energiefehler gespeichert und Iteration 2 wird analog durchgeführt. Der Puls mit dem geringsten Energiefehler wird ausgewählt. Der Codevektor dieser Iteration ist dem Zielvektor am ähnlichsten. Für jede Iteration sind vier Pulspaare zu überprüfen, insgesamt also $4 \times 64 = 256$ Berechnungen. Bei vier Iterationen sind somit 1024 Berechnungen durchzuführen.

[0028] Fig. 3 zeigt eine Tabelle der benachbarten und nicht benachbarten Tracks, die gemeinsam geprüft werden. Aus Fig. 2 ist ersichtlich, dass bestimmte Kombinationen von Tracks häufig auftreten, z.B. Tr_0 - Tr_1 , Tr_1 - Tr_3 , während andere überhaupt nicht vorkommen. Von allen denkbaren Codevektoren wird nur eine kleine Auswahl geprüft. Die linke Spalte von Fig. 3 enthält die für den Suchvorgang erforderlich benachbarten Tracks. Der Suchvorgang gliedert sich in die eigentliche Suchschleife, in der auf einen Block von 64 Werten der Autokorrelationsmatrix zugegriffen wird, wobei für vier Iterationen mit je vier Pulspaaren mit je 64 Werten insgesamt 1024 Matrixzugriffe erfolgen.

[0029] Außerhalb der Suchschleife wird auf je acht Werte zugegriffen, insgesamt erfolgen 1280 Zugriffe auf die Autokorrelationsmatrix. Bei herkömmlichen Verfahren wird die gesamte Autokorrelationsmatrix mit $40 \times 40 = 1600$ Werten gespeichert. Da jedoch jeweils Blocks von 64 Werten benötigt werden, werden sie gemeinsam gespeichert. Die Reihenfolge innerhalb des Blocks wird so gewählt, dass auf die Werte durch eine Programmschleife konstanter Schrittweite zugegriffen werden kann, ohne dass eine umständliche Berechnung von Speicheradressen erforderlich ist.

[0030] Wie der linken Spalte von Fig. 3 zu entnehmen ist, existieren fünf Gruppen mit je 64 Werten benachbar-

ter Tracks mit insgesamt 320 Werten. Ebenso gibt es fünf Kombinationen von nicht benachbarten Tracks gibt, die jeweils 64 Werte umfassen, sodass auch hier insgesamt 320 Werte berechnet werden müssen.

[0031] Fig 4 zeigt eine Diagonalmatrix mit den Koeffizienten einer Kombination zweier gleicher Tracks, beispielsweise Tr0-Tr0. Diese Dreiecksmatrix enthält 28 Koeffizienten. Aus den fünf Kombinationen gleicher Tracks wird ein Block von insgesamt 140 Werten gebildet. Der Zugriff auf diesen Block erfolgt nur relativ selten, lediglich 10% aller Zugriffe fallen in diese Kategorie. Aus diesem Grund ist es kein Nachteil, dass der Zugriff, das heißt die Adressierung der Koeffizienten etwas umständlicher ist. Es ist auch denkbar, für den Zugriff eine Zuordnungstabelle zu verwenden.

[0032] Fig. 5 zeigt die Koeffizienten der Hauptdiagonalen. Da insgesamt 40 Signalabtastungen in einem Zeitabschnitt vorgenommen werden, enthält die Hauptdiagonale 40 Elemente, die in einem Block sequentiell gespeichert werden.

[0033] Insgesamt sind 320 Koeffizienten der Kombinationen benachbarter Tracks, 320 Koeffizienten der Kombinationen nicht benachbarter Tracks, 140 Koeffizienten der Kombinationen gleicher Tracks und 40 Koeffizienten der Hauptdiagonale zu berechnen, zusammen 820 Koeffizienten.

[0034] In Fig. 6 sind sämtliche zu berechnenden Koeffizienten gruppenweise dargestellt. Jedes der ellipsenförmigen Symbole bezeichnet eine Untergruppe mit einer bestimmten Anzahl von Koeffizienten. Bei den Blocks 1 und 2 umfasst jede Untergruppe acht Koeffizienten, im Block 4 jeweils fünf Koeffizienten. Die Anzahl der Koeffizienten im Block 3 ist wegen der Diagonalmatrix unterschiedlich.

[0035] Die Berechnung der einzelnen Blocks wird nun näher erläutert. Jeder der Blocks 1 bis 4 kann getrennt berechnet werden. Die Erzeugung der Blocks 1 und 2 ist praktisch identisch, sie erfolgt in zwei Schritten. In Fig. 7 sind diese Schritte für Block 1 dargestellt. Der erste Schritt beginnt bei dem Wert (38 /39) der Autokorrelationsmatrix. Die Matrix wird dabei diagonal durchlaufen, bis die in Fig 7 eingezeichnete Diagonale den Wert (0/1) erreicht. Dieser Endwert ist mit 'A' bezeichnet und setzt sich bei dem mit 'A' gekennzeichneten Wert (33/39) auf der rechten Seite fort. Entsprechendes gilt für das Symbol 'B'.

[0036] Die Speicherreihenfolge von Block 1 nach dem ersten Schritt ist in Fig. 8 dargestellt, die Pfeile geben dabei an, in welcher Reihenfolge die Koeffizienten aus der Autokorrelationsmatrix in die 8 x 8 Werte umfassenden Blocks gespeichert werden. Der zweite Teilschritt beginnt bei dem Wert (35/39), wie in Fig. 7 gezeigt ist. Diese Diagonale läuft zum Wert (0/4), der zweite Teil beginnt bei dem Wert (30/39) usw.

[0037] In Fig 9 ist die Speicherreihenfolge von Block 1 nach dem zweiten Teilschritt dargestellt. Alle Werte, die bereits in dem ersten Schritt gespeichert wurden, sind in Fig. 9 mit schwarzen Punkten markiert. Durch

diese zwei Schritte wird der gesamte Block gefüllt. Die erste Zeile enthält die Korrelationswerte von Track0-Track1, die zweite Zeile die Korrelationswerte von Track1-Track2 usw., entsprechend Fig 7.

[0038] Fig. 10 zeigt die Berechnung des Blocks 2 mit den Werten nicht benachbarter Tracks, der auf die gleiche Weise erzeugt werden kann. Analog zu Block 1 sind in Fig. 10 die benötigten Diagonalen eingezeichnet. Der erste Teil beginnt bei dem Wert (37/39). Diese Diagonale läuft bis zum Wert (0/2), der erste Teil wird bei dem Wert (32/39) fortgesetzt.

[0039] Fig 11 stellt die Speicherplatzreihenfolge von Block 2 nach diesem ersten Schritt dar. Der zweite Teil beginnt bei dem Wert (36/39). Die Diagonale läuft bis zum Wert (0/3), der zweite Teil wird bei dem Wert (31/39) fortgesetzt.

[0040] In Fig. 12 ist die Speicherplatzreihenfolge von Block 2 nach dem zweiten Schritt dargestellt. Alle Werte, die bereits in dem ersten Schritt gespeichert wurden, sind mit Punkten markiert.

[0041] Fig. 13 stellt die Berechnung des Blocks der Kombinationen gleicher Tracks dar. Analog zu den vorherigen Beispielen sind die benötigten Diagonalen eingezeichnet. Block 3 kann in einem einzigen Durchlauf berechnet werden. Die Speicherplatzreihenfolge von Block 3 ist in Fig. 14 dargestellt.

[0042] Bei den Koeffizienten für Block 4 handelt es sich um die Werte der Hauptdiagonalen der Autokorrelationsmatrix.

[0043] Im Vergleich zu der herkömmlichen Lösung, bei der 1600 Koeffizienten berechnet und gespeichert werden, sind bei diesem Verfahren lediglich 820 Koeffizienten zu berechnen. Dadurch ergibt sich eine Verringerung des Rechenaufwands von annähernd 30%. Der RAM-Speicherbedarf wird um etwa 40% verkürzt.

[0044] Zur weiteren Verkürzung der Rechenzeit werden die Blocks 1 und 2 in separaten RAM-Speicherbänken eines Speichers abgelegt, so dass zwei Werte gleichzeitig ausgelesen werden können.

Patentansprüche

1. Verfahren für die algebraische Codebook-Suche eines Sprachsignalkodierers, vorzugsweise nach dem Code Excited Linear Prediction-Verfahren, bei dem zur Berechnung von Koeffizienten der Dreiecksmatrix der Autokorrelationsmatrix vom Toeplitz-Typ ein n Sprachsignalabtastungen umfassender Zeitabschnitt in eine ganzzahlige Anzahl Tracks t mit je p möglichen Pulspositionen zerlegt wird,
dadurch gekennzeichnet,
dass die Koeffizienten gruppiert nach
 - Kombinationen benachbarter Tracks;
 - Kombinationen nicht benachbarter Tracks;
 - Kombinationen gleicher Tracks; und
 - Koeffizienten der Hauptdiagonale der Autokor-

relationsmatrix in einem Speicher abgelegt werden.

2. Verfahren nach Anspruch 1,
dadurch gekennzeichnet,
dass für die Gruppen der Kombinationen benachbarter und nicht benachbarter Tracks jeweils t Datensätze mit je $p \times p$ Koeffizienten gespeichert werden.
3. Verfahren nach Anspruch 1 oder 2,
dadurch gekennzeichnet,
dass die Koeffizienten sequentiell in einem Speicher abgelegt werden.
4. Verfahren nach Anspruch 2 oder 3,
dadurch gekennzeichnet,
dass eine einen horizontalen oder vertikalen Vektor der Autokorrelationsmatrix darstellende Untergruppe eines Datensatzes mit p Koeffizienten durch eine Programmschleife ausgelesen wird, wobei ein die Speicherstelle des ersten Koeffizienten bezeichnender Wert und eine konstante Schrittweite bis zur nächsten Speicherstelle vorgegeben werden.
5. Verfahren nach Anspruch 4,
dadurch gekennzeichnet,
dass für die Datensätze der Gruppe der Kombinationen benachbarter Tracks die Schrittweite eins gewählt wird.
6. Verfahren nach Anspruch 4,
dadurch gekennzeichnet,
dass für die Datensätze der Gruppe der Kombinationen nicht benachbarter Tracks die Schrittweite p gewählt wird.
7. Verfahren nach einem der vorangehenden Ansprüche,
dadurch gekennzeichnet,
dass für die Gruppe der Kombinationen gleicher Tracks t Dreiecksmatrizen sequentiell gespeichert werden.
8. Verfahren nach Anspruch 7,
dadurch gekennzeichnet,
dass der Zugriff auf die Koeffizienten der Gruppe gleicher Tracks über eine Nachschlagetabelle erfolgt.
9. Verfahren nach einem der vorangehenden Ansprüche,
dadurch gekennzeichnet,
dass die Koeffizienten der Hauptdiagonale sequentiell gespeichert werden.
10. Verfahren nach einem der vorangehenden Ansprüche,

dadurch gekennzeichnet,

dass innerhalb eines Zeitabschnitts 40 Sprachsignalabtastungen erfasst werden.

- 5 11. Verfahren nach einem der vorangehenden Ansprüche,
dadurch gekennzeichnet,
dass die Autokorrelationsmatrix eine 40×40 -Matrix ist.
- 10 12. Verfahren nach einem der vorangehenden Ansprüche,
dadurch gekennzeichnet,
dass ein Zeitabschnitt in fünf Tracks mit je acht möglichen Pulspositionen zerlegt wird.
- 15 13. Verfahren nach einem der Ansprüche 1 bis 11,
dadurch gekennzeichnet,
dass ein Zeitabschnitt in vier Tracks mit je zehn möglichen Pulspositionen zerlegt wird.
- 20 14. Verfahren nach einem der vorangehenden Ansprüche,
dadurch gekennzeichnet,
dass für die Gruppe der Kombinationen benachbarter Tracks 320 Koeffizienten ermittelt werden.
- 25 15. Verfahren nach einem der vorangehenden Ansprüche,
dadurch gekennzeichnet,
dass für die Gruppe der Kombinationen nicht benachbarter Tracks 320 Koeffizienten ermittelt werden.
- 30 16. Verfahren nach einem der vorangehenden Ansprüche,
dadurch gekennzeichnet,
dass für die Gruppe der Kombinationen gleicher Tracks 140 Koeffizienten ermittelt werden.
- 40 17. Verfahren nach einem der vorangehenden Ansprüche,
dadurch gekennzeichnet,
dass insgesamt 820 Koeffizienten ermittelt werden.
- 45 18. Verfahren nach einem der vorangehenden Ansprüche,
dadurch gekennzeichnet,
dass Koeffizientengruppen in verschiedenen RAM-Speicherbänken eines mehrere RAM Speicherbänke aufweisenden Speichers gespeichert werden.
- 50 19. Kommunikationseinrichtung mit einem Sprachsignalkodierer, insbesondere Mobiltelefon,
dadurch gekennzeichnet,
dass es ein Betriebssystem mit Mitteln zur Implementierung jedes einzelnen der Schritte eines Ver-
- 55

fahrens nach einem der Ansprüche 1 bis 18 aufweist.

Claims

1. A method for the algebraic codebook search of a speech signal encoder, preferably using the Code Excited Linear Prediction process, in which, in order to calculate coefficients of the triangular matrix of the auto-correlation matrix of the Toeplitz type, a time interval comprising n speech signal samplings is broken down into an integral number of tracks t with p possible pulse positions each, **characterized in that** the coefficients are stored in a memory grouped in
 - combinations of adjacent tracks;
 - combinations of non-adjacent tracks;
 - combinations of identical tracks; and
 - coefficients of main diagonals of the auto-correlation matrix.
2. A method as claimed in claim 1, **characterized in that** for the groups of combinations of adjacent and non-adjacent tracks in each case t data records with p x p coefficients each are stored.
3. A method as claimed in claim 1 or 2, **characterized in that** the coefficients are stored sequentially in a memory.
4. A method as claimed in claim 2 or 3, **characterized in that** a sub-group of a data record with p coefficients, representing a horizontal or vertical vector of the auto-correlation matrix, is read through a program loop where a value indicating the memory point of the first coefficient and a constant step width to the next memory point are prespecified.
5. A method as claimed in claim 4, **characterized in that** for the data records of the group of combinations of adjacent tracks the step value 1 is selected.
6. A method as claimed in claim 4, **characterized in that** for the data records of the group of combinations of non-adjacent tracks the step value p is selected.
7. A method as claimed in any of the previous claims, **characterized in that** for the group of combinations of identical tracks t triangular matrices are stored sequentially.
8. A method as claimed in claim 7, **characterized in that** access to the coefficients of the group of identical tracks takes place via a lookup table.

9. A method as claimed in any of the previous claims, **characterized in that** the coefficients of the main diagonals are stored sequentially.
10. A method as claimed in any of the previous claims, **characterized in that** 40 speech signal samplings are performed within a time interval.
11. A method as claimed in any of the previous claims, **characterized in that** the auto-correlation matrix is a 40 x 40 matrix.
12. A method as claimed in any of the previous claims, **characterized in that** a time interval is divided into five tracks of eight possible pulse positions each.
13. A method as claimed in any of claims 1 to 11, **characterized in that** a time interval is divided into four tracks of ten possible pulse positions each.
14. A method as claimed in any of the previous claims, **characterized in that** for the group of combinations of adjacent tracks 320 coefficients are determined.
15. A method as claimed in any of the previous claims, **characterized in that** for the group of combinations of non-adjacent tracks 320 coefficients are determined.
16. A method as claimed in any of the previous claims, **characterized in that** for the group of combinations of identical tracks 140 coefficients are determined.
17. A method as claimed in any of the previous claims, **characterized in that** a total of 820 coefficients are determined.
18. A method as claimed in any of the previous claims, **characterized in that** coefficient groups are stored in various RAM memory banks of a memory having several RAM memory banks.
19. A communication device with a speech signal encoder, in particular a mobile phone, **characterized in that** it includes an operating system with means enabling each one of the steps to be implemented, using a method as claimed in any of the claims 1 to 18.

Revendications

1. Procédé pour la recherche algébrique de table de codes pour un codeur de signaux de parole, de préférence selon le procédé Code Excited Linear Prediction dans lequel, pour le calcul de coefficients de la matrice triangulaire de la matrice d'autocorrélation du type Toeplitz, un segment temporel compre-

- nant n balayages du signal de parole est décomposé en un nombre entier de pistes t avec p positions d'impulsion possibles chacune, **caractérisé en ce que** les coefficients sont groupés dans une mémoire en combinaisons de pistes adjacentes, combinaison de pistes non adjacentes, combinaisons de pistes identiques et coefficients des diagonales principales de la matrice d'autocorrélation.
2. Procédé selon la revendication 1, **caractérisé en ce que** t jeux de données avec p x p coefficients chacun sont enregistrés pour les groupes de combinaisons des pistes adjacentes et non adjacentes.
 3. Procédé selon l'une des revendications 1 ou 2, **caractérisé en ce que** les coefficients sont enregistrés séquentiellement dans une mémoire.
 4. Procédé selon l'une des revendications 2 ou 3, **caractérisé en ce qu'un** sous-groupe d'un jeu de données avec p coefficients représentant un vecteur horizontal ou vertical de la matrice d'autocorrélation est lu par une boucle de programme, une valeur désignant l'emplacement de mémoire du premier coefficient et une largeur de pas constante étant préalablement déterminées jusqu'à l'emplacement de mémoire suivant.
 5. Procédé selon la revendication 4, **caractérisé en ce que** la largeur de pas un est choisie pour les jeux de données du groupe des combinaisons de pistes adjacentes.
 6. Procédé selon la revendication 6, **caractérisé en ce que** la largeur de pas p est choisie pour les jeux de données du groupe des combinaisons de pistes non adjacentes.
 7. Procédé selon l'une des revendications précédentes, **caractérisé en ce que** t matrices triangulaires sont enregistrées séquentiellement pour le groupe des combinaisons de pistes identiques.
 8. Procédé selon la revendication 7, **caractérisé en ce que** l'accès aux coefficients du groupe de pistes identiques est assuré par l'intermédiaire d'un tableau de référence.
 9. Procédé selon l'une des revendications précédentes, **caractérisé en ce que** les coefficients de la diagonale principale sont enregistrés séquentiellement.
 10. Procédé selon l'une des revendications précédentes, **caractérisé en ce que** 40 balayages de signaux de parole sont enregistrés dans un segment de parole.
 11. Procédé selon l'une des revendications précédentes, **caractérisé en ce que** la matrice d'autocorrélation est une matrice 40 x 40.
 12. Procédé selon l'une des revendications précédentes, **caractérisé en ce que** le segment temporel est décomposé en cinq pistes avec huit positions d'impulsion possibles chacune.
 13. Procédé selon l'une des revendications 1 à 11, **caractérisé en ce qu'un** segment temporel est décomposé en quatre pistes avec dix positions d'impulsion possibles chacune.
 14. Procédé selon l'une des revendications précédentes, **caractérisé en ce que** 320 coefficients sont déterminés pour le groupe des combinaisons de pistes adjacentes.
 15. Procédé selon l'une des revendications précédentes, **caractérisé en ce que** 320 coefficients sont déterminés pour le groupe des combinaisons de pistes non adjacentes.
 16. Procédé selon l'une des revendications précédentes, **caractérisé en ce que** 140 coefficients sont déterminés pour le groupe des combinaisons de pistes identiques.
 17. Procédé selon l'une des revendications précédentes, **caractérisé en ce que** 820 coefficients sont déterminés au total.
 18. Procédé selon l'une des revendications précédentes, **caractérisé en ce que** des groupes de coefficients sont enregistrés dans différentes bases de mémoire RAM d'une mémoire présentant plusieurs bases de mémoire RAM.
 19. Dispositif de communication avec un codeur de signaux de parole, en particulier un radiotéléphone mobile, **caractérisé en ce qu'il** présente un système d'exploitation avec des moyens pour la mise en oeuvre de chacune des différentes étapes d'un procédé selon l'une des revendications 1 à 18.

track	puls position									
1	0	4	8	12	16	20	24	28	32	36
2	1	5	9	13	17	21	25	29	33	37
3	2	6	10	14	18	22	26	30	34	38
4	3	7	11	15	19	23	27	31	35	39

Fig.1

	lp 0	lp 1	lp 2	lp 3	lp 4	lp 5	lp 6	lp 7
iteration 1	2	3	4	0	1	2	3	4
iteration 2	2	4	0	1	2	3	4	0
iteration 3	2	0	1	2	3	4	0	1
iteration 4	2	1	2	3	4	0	1	3

Fig.2

adjacent tracks	non-adjacent tracks
Tr 0 -Tr 1	Tr 0 -Tr 2
Tr 1 -Tr 2	Tr 1 -Tr 3
Tr 2 -Tr 3	Tr 2 -Tr 4
Tr 3 -Tr 4	Tr 3 -Tr 0
Tr 4 -Tr 0	Tr 4 -Tr 1

Fig.3

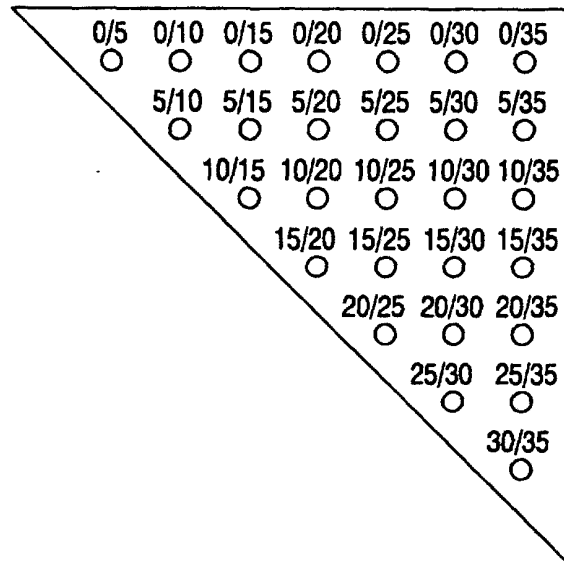


Fig.4

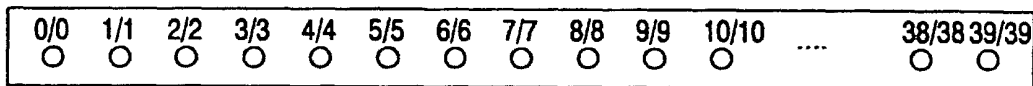


Fig.5

block 1

track 0 - track 1 :
track 1 - track 2 :
track 2 - track 3 :
track 3 - track 4 :
track 4 - track 0 :

8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8

block 2

track 0 - track 2 :
track 1 - track 3 :
track 2 - track 4 :
track 3 - track 0 :
track 4 - track 1 :

8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8

block 3

track 0 - track 0 :
track 1 - track 1 :
track 2 - track 2 :
track 3 - track 3 :
track 4 - track 4 :

7	6	5	4	3	2	1
7	6	5	4	3	2	1
7	6	5	4	3	2	1
7	6	5	4	3	2	1
7	6	5	4	3	2	1

block 4

main diagonals

5	5	5	5	5	5	5	5
---	---	---	---	---	---	---	---

Fig.6

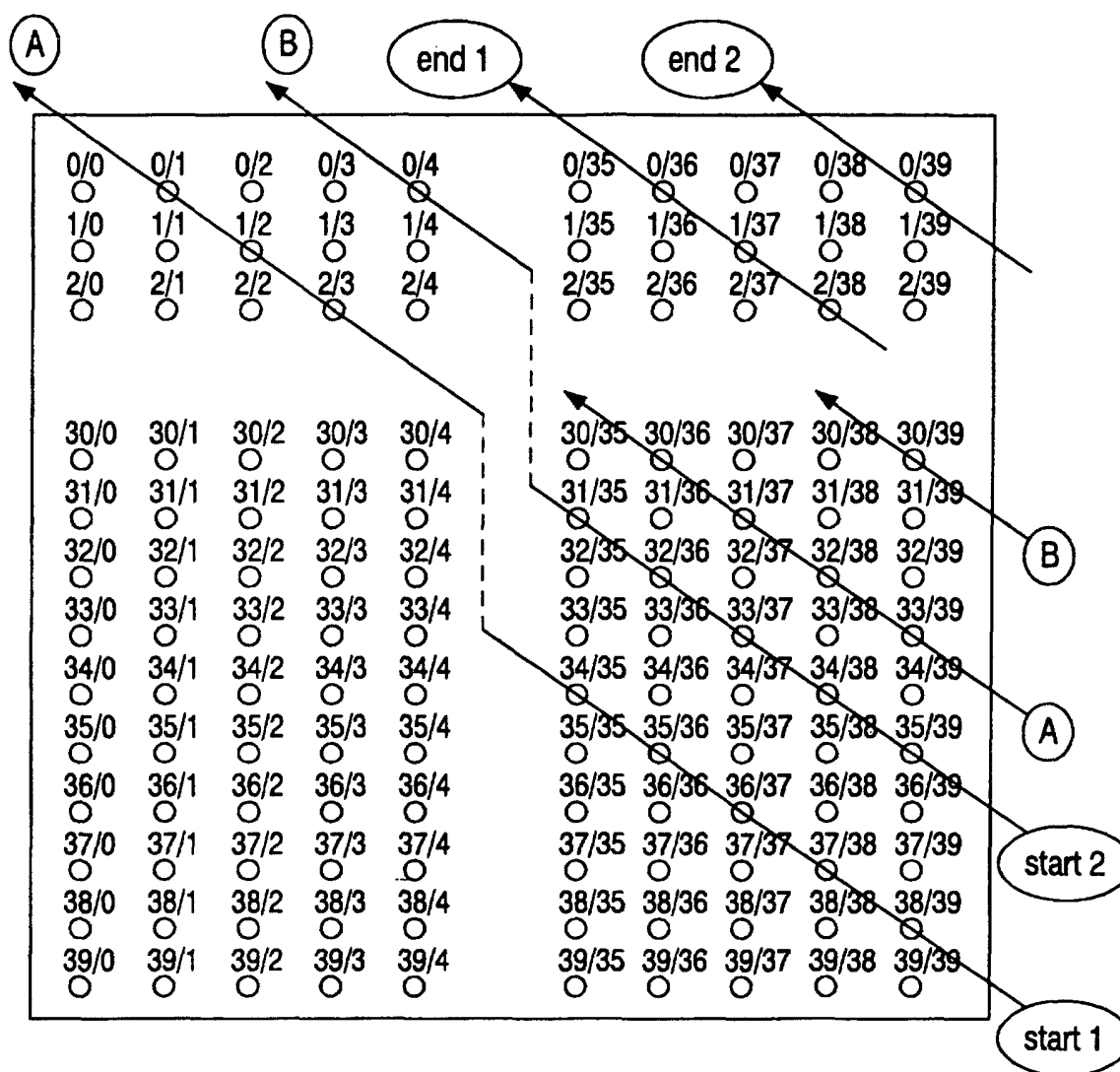


Fig.7

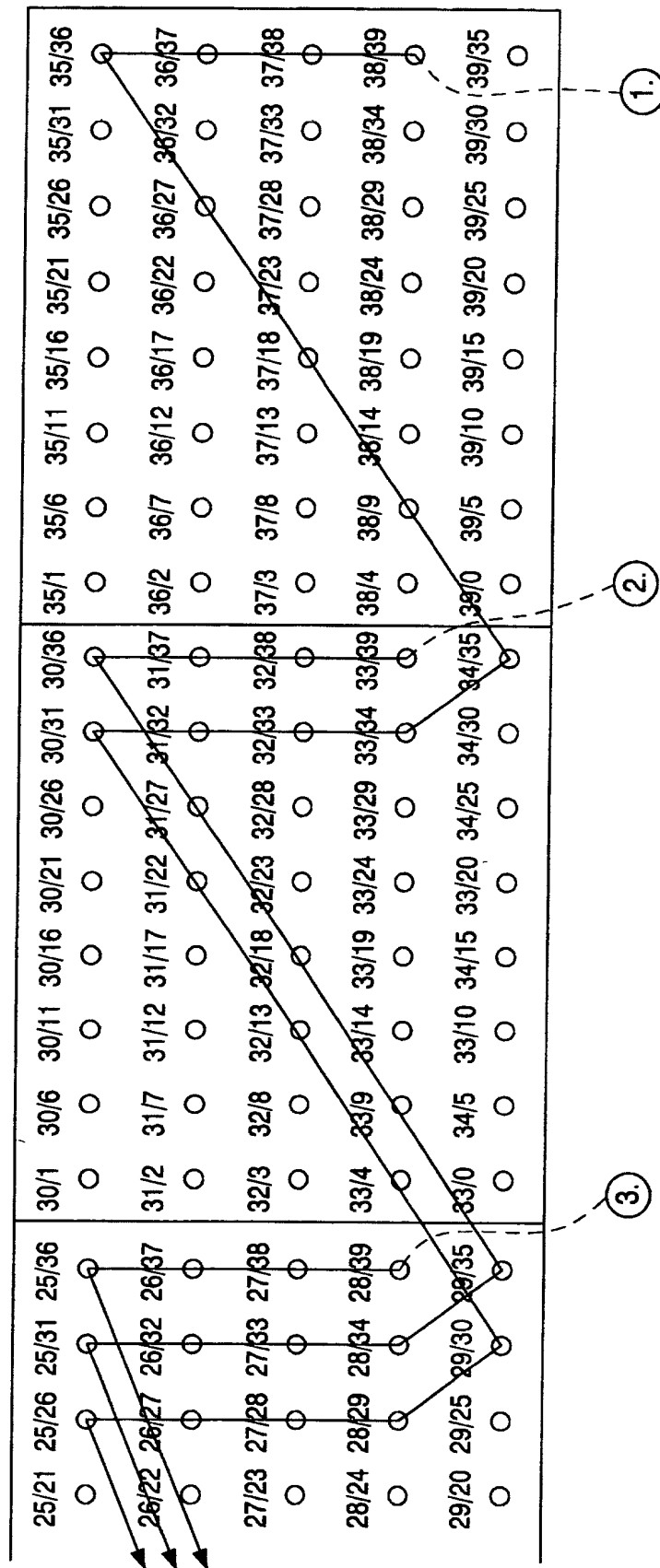


Fig.8

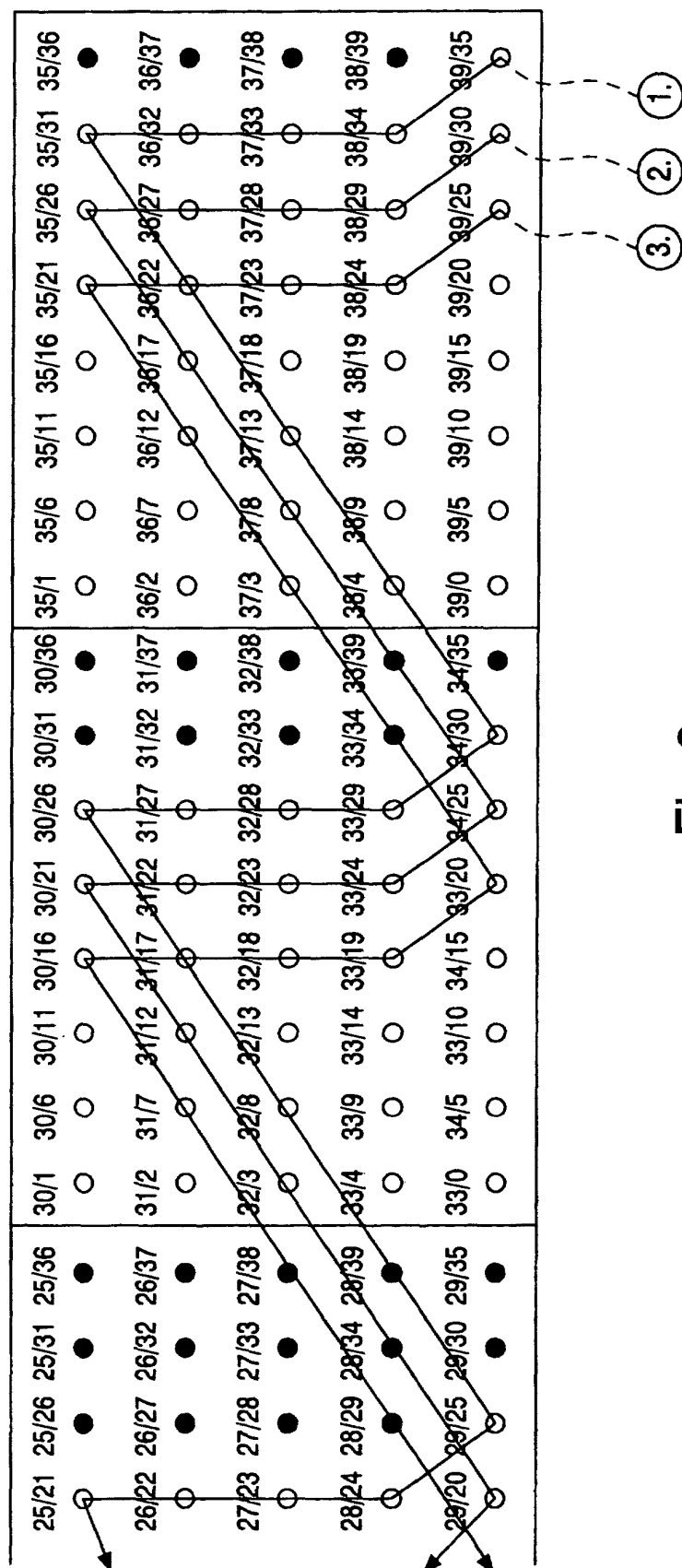


Fig. 9

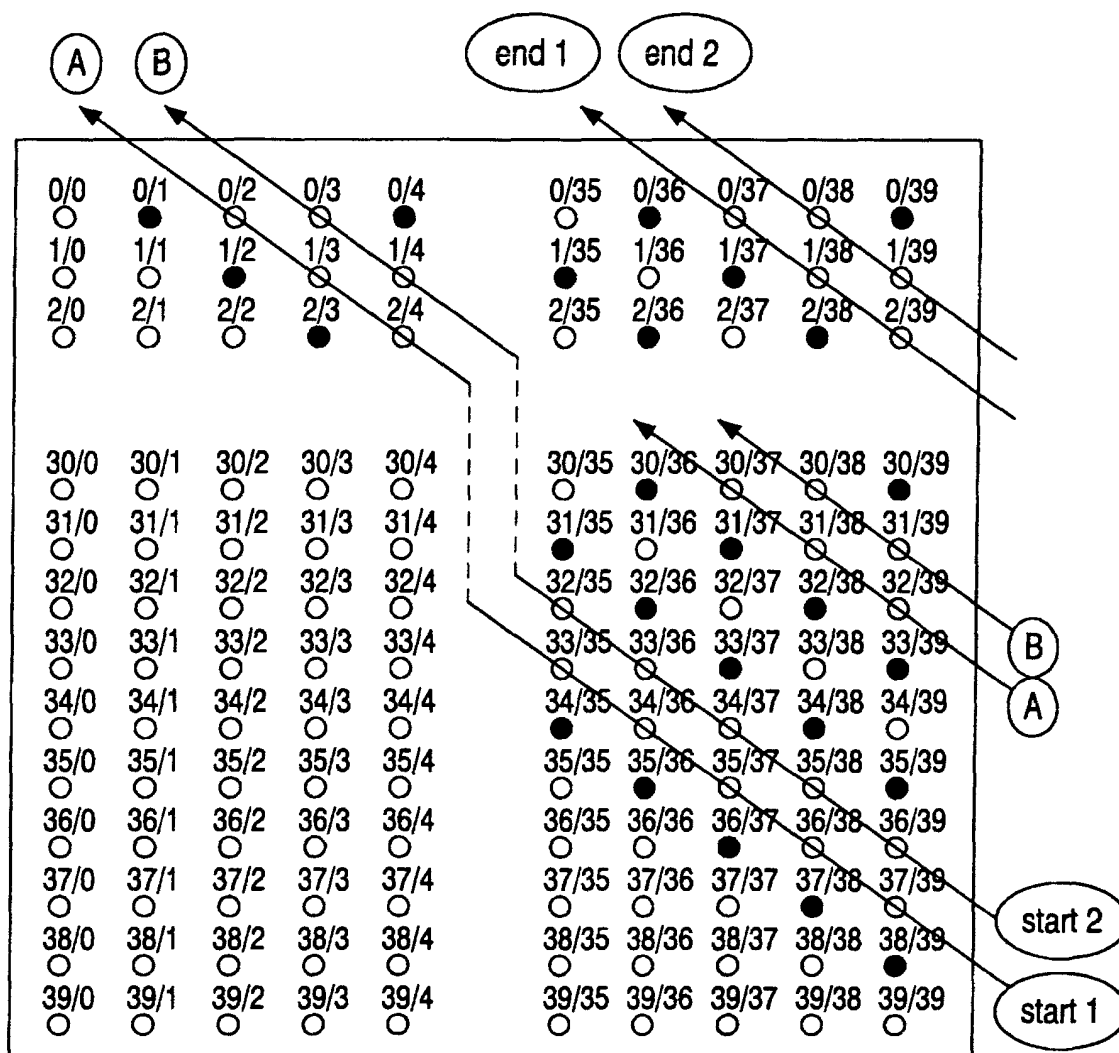


Fig.10

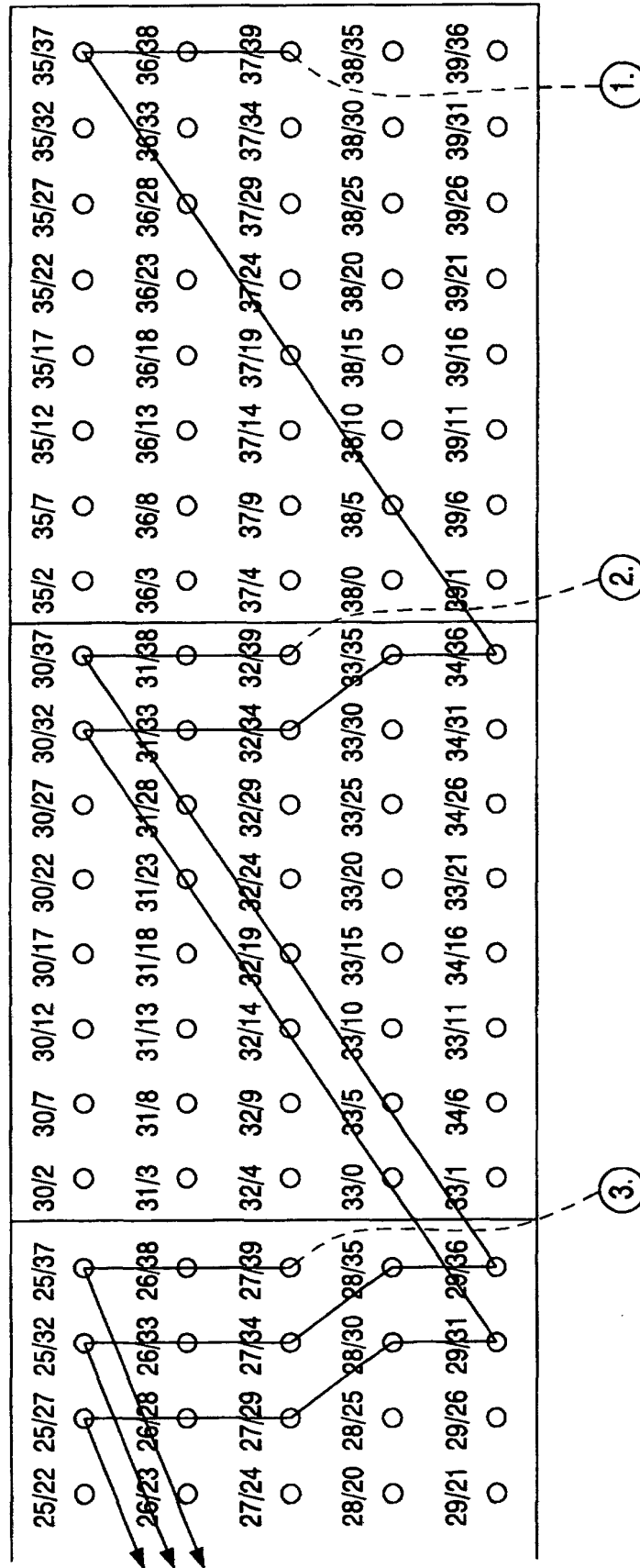


Fig.11

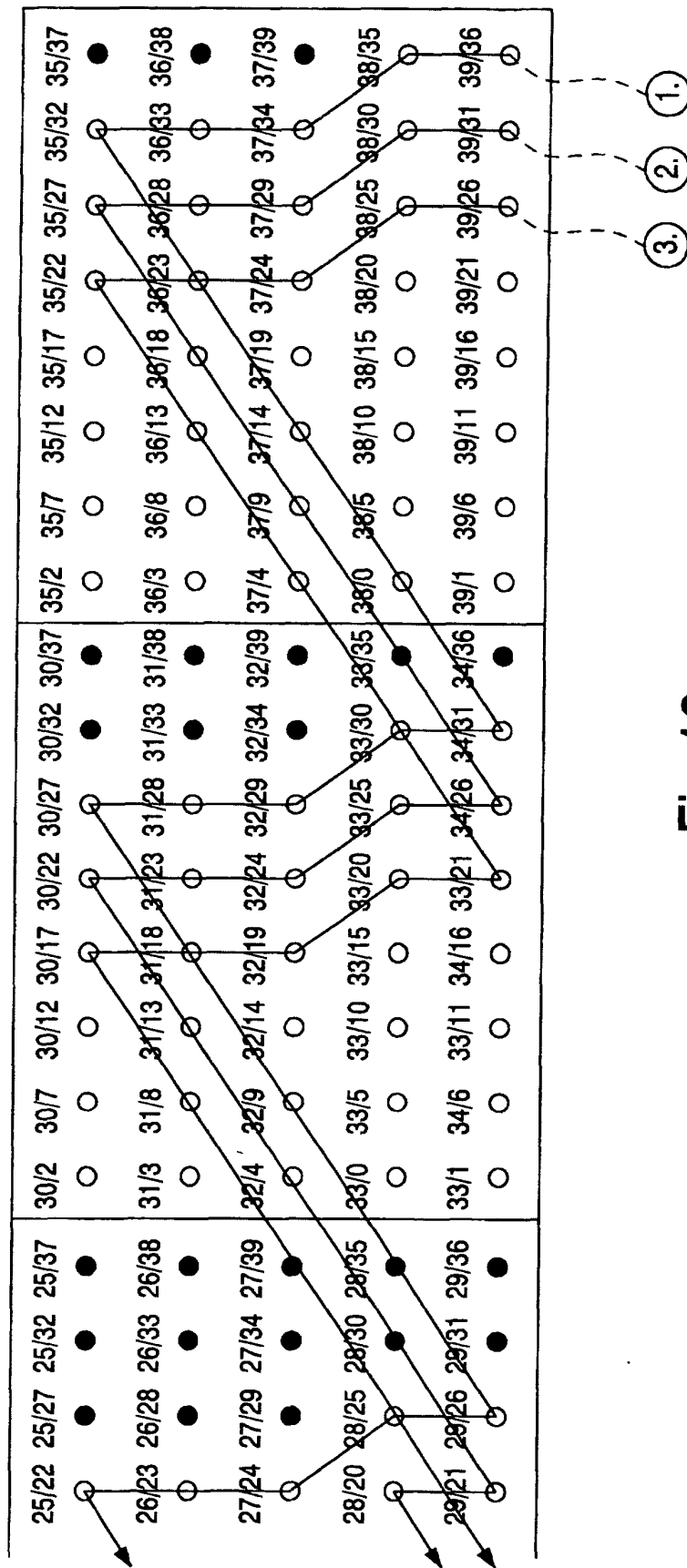


Fig.12

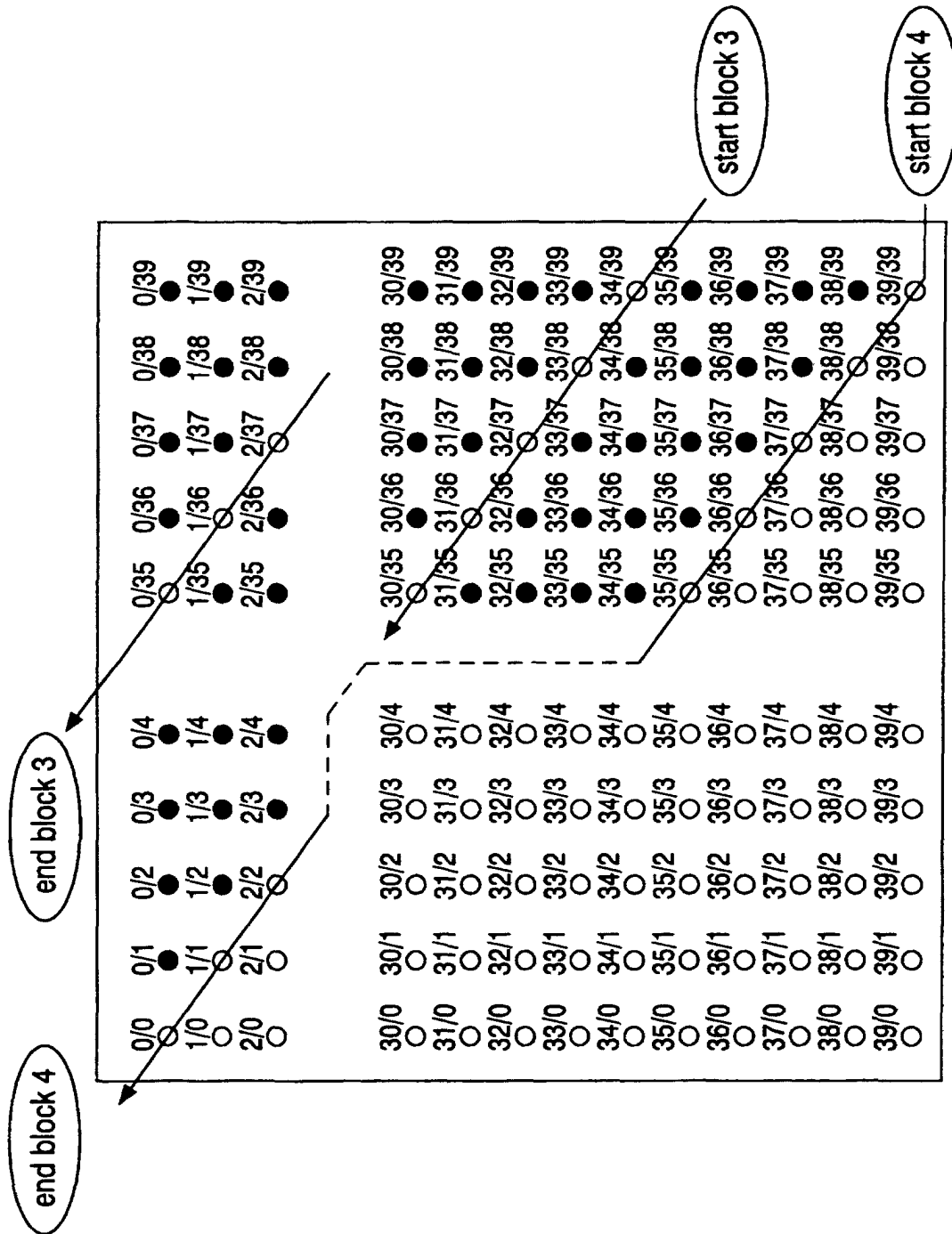


Fig.13

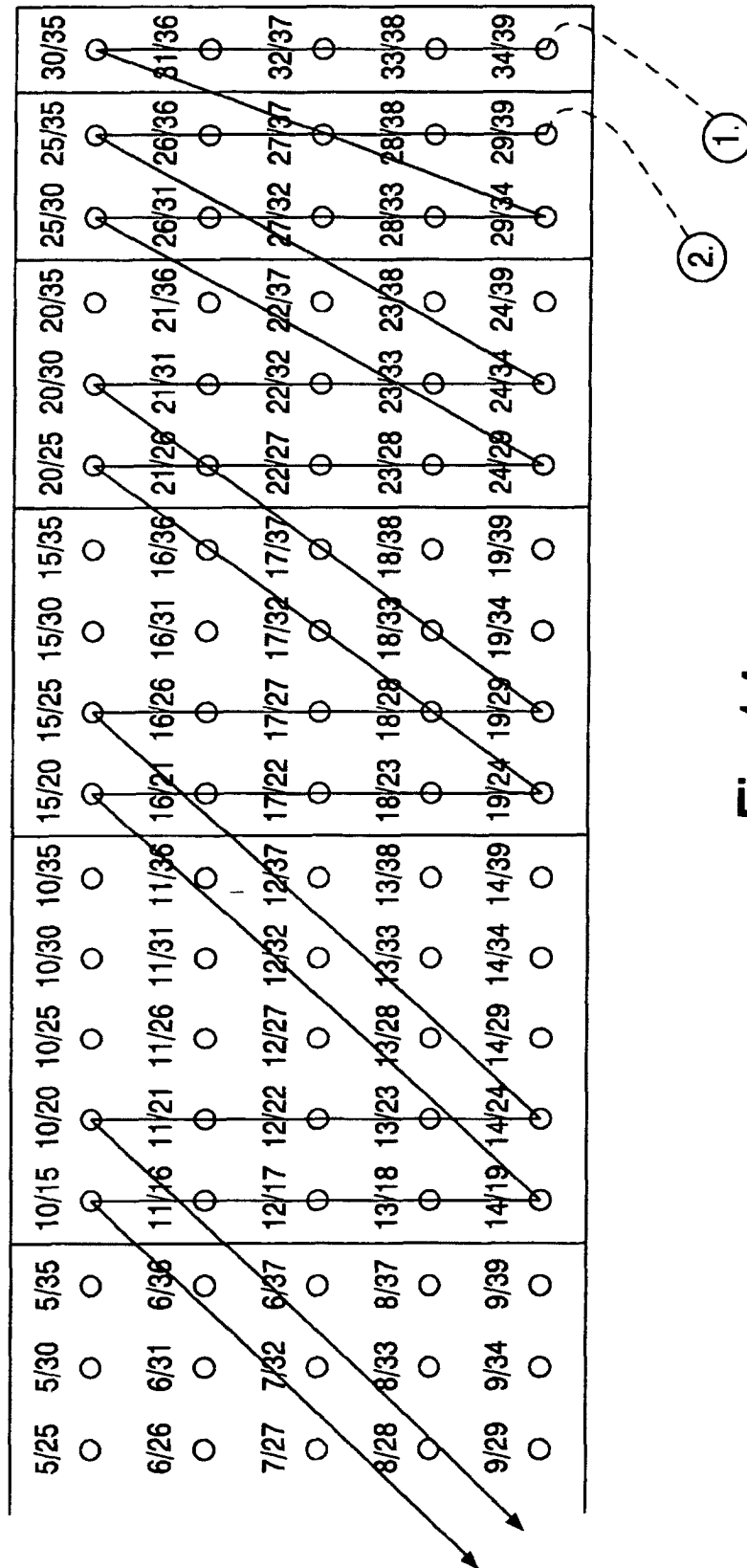


Fig.14