



(19)  
Bundesrepublik Deutschland  
Deutsches Patent- und Markenamt

(10) **DE 695 33 074 T2** 2004.09.16

(12)

## Übersetzung der europäischen Patentschrift

(97) **EP 1 030 263 B1**

(21) Deutsches Aktenzeichen: **695 33 074.8**

(96) Europäisches Aktenzeichen: **00 107 920.1**

(96) Europäischer Anmeldetag: **01.09.1995**

(97) Erstveröffentlichung durch das EPA: **23.08.2000**

(97) Veröffentlichungstag

der Patenterteilung beim EPA: **19.05.2004**

(47) Veröffentlichungstag im Patentblatt: **16.09.2004**

(51) Int Cl.7: **G06K 9/20**

(30) Unionspriorität:  
**303929            09.09.1994    US**

(73) Patentinhaber:  
**Xerox Corp., Rochester, N.Y., US**

(74) Vertreter:  
**Grünecker, Kinkeldey, Stockmair &  
Schwanhäusser, 80538 München**

(84) Benannte Vertragsstaaten:  
**DE, FR, GB**

(72) Erfinder:  
**Saund, Eric, San Carlos, California 94070, US**

(54) Bezeichnung: **Verfahren zum Interpretieren handgeschriebener schematischer Benutzerschnittstellenbefehle**

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 das Steuern einer Vorrichtung, besonders einer Rechenvorrichtung, durch handgezeichnete Markierungen auf einer weißen oder schwarzen Wandtafel. Insbesondere betrifft die Erfindung Bildanalyseverfahren zum Interpretieren von Markierungen zum Zweck des Steuerns von Vorrichtungen.

[0002] In zusammenarbeitenden Arbeitsumgebungen wünschen oftmals mehrere Benutzer, angezeigte Information gleichzeitig zu betrachten und zu bearbeiten. Weiße und schwarze Wandtafeln (nachstehend "Tafeln") werden weitverbreitet verwendet, um handgezeichnete Text- und Grafikbilder auf einer "wandgroßen" Fläche zu bewahren. Das Tafelmedium bietet bestimmte Eigenschaften, die eine Vielfalt von interaktiven Arbeitsverfahren fördern: Markierungen sind groß genug, um von mehreren Leuten betrachtet zu werden; Markierungen können durch Löschen und Neuzeichnen editiert werden; die Fläche ist ortsfest, wird also nicht verloren, zerstückelt, zerrissen oder vom Wind verweht; die Fläche ist leicht zu löschen, vollständig wiederverwendbar und hat (praktisch) keinen Verschleiß. Ein Nachteil des Gebrauchs einer Tafel ist jedoch, dass Information nicht ohne weiteres auf andere Medien zu übertragen ist. Gegenwärtig ist es daher nicht möglich, sich mit jemandem zu unterhalten, dabei die Unterhaltung in Text und Grafik auf einer Tafel aufzuzeichnen, und die Aufzeichnung dann schnell, einfach und automatisch auf Papier oder ein anderes tragbares und lagerbares Medium zu übertragen.

[0003] Bestehende Verfahren, die diese Aufgabe erfüllen, sind mühsam, zeitraubend und unbequem. Man kann den Text und die Grafik, die sich auf der Tafel befinden, ganz oder teilweise einfach von Hand auf Papier übertragen. Dies kann zeitraubend sein und leidet unter Fehlern, die auf Fehler beim menschlichen Lesen und Schreiben zurückzuführen sind. Oder man kann die Tafel mit einer Kamera fotografieren. Dies erfordert, dass eine Kamera zur Hand ist, bringt die Verzögerung der Filmentwicklung ein, kann teuer sein, wenn eine "Instant"-Kamera benutzt wird, und unterliegt schlechter Wiedergabequalität infolge von Unschärfe und ungeeigneter Belichtung. Eine Kamera erzeugt außerdem ein Bild mit stark vermindelter Größe, das schwer zu lesen sein kann.

[0004] "Design issues for hand driven text editing/annotation systems" von Guy Hardock beschreibt ein Computersystem, das handgeschriebene Markierungen zu erkennen vermag, um dem System zu ermöglichen, Texteditierprozeduren durchzuführen. Dies wird erreicht, indem man einen Operator auf ein elektronisches Tablett schreiben lässt. Der Computer analysiert dann die handgeschriebenen Markierungen und bestimmt, welche Operationen auszuführen sind.

[0005] Alternativ führen "wandgroße" Papierbögen, z. B. Plakatblöcke, zu einer relativ dauerhaften und

tragbaren Aufzeichnung des Geschriebenen, aber diese Papierbögen sind groß und schwerfällig und gestatten kein Löschen während der Erstellung des Bildes.

[0006] Eine Kopiertafel-Vorrichtung stellt eine Schreibfläche bereit, die in eine Papierkopie umgesetzt werden kann. Diese wird aber gegenwärtig als bemerkenswerte tragbare Weißtafel verstanden, die bestehende eingebaute Tafeln verdrängt, anstatt sie zu stärken.

[0007] Die oben erörterten bekannten Verfahren helfen außerdem nicht bei dem Umsetzen des Bildes von der Tafel in eine elektronisch verwendbare Form.

[0008] EP-A-0546343 offenbart ein Diagramm-Erkennungssystem, das in der Lage ist, ein Diagramm, das z. B. durch eine manuelle Operation oder durch eine Zeichensoftware in einer Weise, die unabhängig von einem CASE-Werkzeug oder dergleichen ist, erstellt wurde, automatisch zu erkennen und eine erkannte Logik ohne die Hilfe einer manuellen Operation an das CASE-Werkzeug zu liefern.

[0009] EP-A-0572031 beschreibt ein elektronisches Tafelsystem mit einem verkleinerten Anzeigebereich, in dem Zeichen, Editierbefehle und dergleichen mit einer Zeigeeinrichtung handgezeichnet werden können.

[0010] Die gleichzeitig eingereichte EP-A-0701225, die der US-Patentanmeldung Seriennummer 08/303,918 entspricht, liefert spezifische technische Details für eine Einrichtung zum Umsetzen von Markierungen auf einer Tafel in elektronische Form. Zusammengefasst, eine Videokamera ist auf einem Schwenk-/Neigekopf montiert. Hochauflösende Fliesen werden durch Zoomen in der Kamera auf Flecken des Bildes erhalten. Diese werden später zusammengestückt, um ein großes zusammengesetztes Bild mit hoher Auflösung zu bilden. Perspektivische Verzerrung, Effekte von ungleichmäßiger Beleuchtung und Fliesenüberschneidung werden durch Bildverarbeitungsoperationen gehandhabt.

[0011] Eine solche Umsetzvorrichtung ist nützlich, weil eine elektronische Version eines Tafelbildes Vielfältigkeit und Flexibilität bei der weiteren Verwendung der Bilddaten bereitstellt. Ein elektronisches Bild kann z. B. hart kopiert, mit Fax gesendet, in einer Datei gespeichert, auf eine elektronische Workstation übertragen oder auf einen Schirm projiziert werden. Außerdem kann vor jeder dieser Operationen das Bild selbst bearbeitet werden, z. B. Auswählen nur eines Bereichs des Bildes, Auswählen nur bestimmter Farben, Verstärken oder Berichtigen des Linienwerks, Neuordnen von Elementen in einer Liste usw. [0012] Die Fülle von Operationen, die durch die Grundfähigkeit, ein Tafelbild umzusetzen, verfügbar gemacht wird, wirft die Frage nach der Steuerung auf: wie soll der Benutzer auszuführende Operationen spezifizieren und wann?

[0013] Da die Tafelumsetzungs- und Bildbearbeitungsoperationen computerbasiert sind, besteht für Benutzer eine Möglichkeit, sich zu ihren Computern

zu begeben, um diese Funktionen zu steuern. Diese Lösung ist aus mehreren Gründen nicht wünschenswert. Erstens, sie zwingt Benutzer, den Takt ihrer Arbeit an der Tafel zu unterbrechen, um sich einer Computerkonsole zuzuwenden. Zweitens, entweder muss am Ort der Tafel eine Konsole bereitgestellt werden oder Benutzer müssen eine unbestimmte Strecke dorthin zurücklegen, wo eine zur Verfügung steht. Drittens, viele Tafelbenutzer sind wahrscheinlich mit Computern im Allgemeinen oder mit den einzelnen Tastatur- und Mausbefehlen, die zum Bedienen des Programms nötig sind, nicht vertraut und/oder haben ein ungutes Gefühl dabei.

[0014] Ein zweiter Typ von Benutzerschnittstelle besteht aus einer dedizierten Bedienkonsole, die angrenzend an die Tafel montiert ist. Wenn die Bedienkonsole aus beschrifteten Tasten besteht, können diese mit einem maßvollen Satz von möglichen Operationen in Verbindung gebracht werden, z. B. Richten der umgesetzten Bitmap an einen einer handvoll Drucker oder Dateiverzeichnissen. Größere Flexibilität könnte erhalten werden, indem der Bedienkonsole eine Tastatur beigegeben wird, aber dies beginnt, neuen Benutzern ein einschüchterndes Gefüge zu präsentieren. Trotzdem ist für einige Verkörperungen einer Tafelumsetzungsvorrichtung eine dedizierte Bedienkonsole wahrscheinlich angemessen.

[0015] In dem System der vorliegenden Erfindung ist jedoch ein dritte Alternative vorhanden, die auf mehrfache Weise für nahtlose Erstellung, Erfassung und elektronisch vermittelten Gebrauch von Bildern, die auf einer Tafel entstehen, ideal geeignet ist. Die Benutzerschnittstelle soll aus Markierungen bestehen, die vom Benutzer auf der Tafel selbst gezeichnet werden. Im einfachsten Fall könnte der Benutzer z. B. ein spezielles "Tasten"-Symbol zeichnen und dann innen eine Prüfmarke zeichnen, wenn die Taste zu "drücken" ist. Das System, das die Tasten kennt, würde auf den Daten, die auf dem Tastendruck basieren, arbeiten. Eine gesteigerte Funktionalität kann erreicht werden, indem das Tastensymbol mit weiteren Anweisungen kommentiert wird, oder indem verschiedene Arten von Tasten für verschiedene Operationen spezifiziert werden.

[0016] Die vorangehend beschriebene Schnittstelle beseitigt nicht die Notwendigkeit, dass Benutzer Kenntnis davon besitzen, wie mit dem System in dieser schematischen Weise zu kommunizieren ist, aber diese Kenntnis kann für viele Tafelbenutzer erreichbarer und leichter assimilierbar sein als jede Bedienkonsole oder Computerkonsolen-basierte Schnittstelle.

[0017] Es besteht daher Bedarf an einem höheren Entwicklungsstand in der Bildanalyse, die eine schematische Benutzerschnittstelle zu einem Tafelumsetzungssystem unterstützt. Die Leistung einer Schnittstelle kann durch die Anwendung von Computersichtverfahren in der geometrischen Analyse der Markierungen auf der Tafel stark erhöht werden. Benutzer sollten in der Lage sein, Tasten auf der Tafel ir-

gendwo und jederzeit zu zeichnen, sie sollten imstande sein, Tasten zu drücken, indem sie darin Kreuze (X) oder Prüfmarken machen, und sie sollten in der Lage sein, komplexere Diagramme zu erstellen, um symbolische und geometrische Funktionalität zu steuern, z. B. durch Spezifizieren des Druckers, auf den zu kopieren ist, oder des zu extrahierenden Bereichs einer Tafel.

[0018] Interpretation von schematischen Benutzerschnittstellen ist wegen der in bedeutungsvollen handgezeichneten Befehlen gefundenen großen Variationsbreite ein schwieriges Computersichtproblem. Symbole und Text können an jeder Stelle und in jedem räumlichen Maßstab (Größe) vorkommen, scheinbar gerade Linien sind selten wirklich gerade, vermeintlich durchgehende Linien haben unechte Lücken und Verzweigungen, und die Konventionen der formalen Geometrie (z. B., dass ein Quadrat aus zwei Sätzen von parallelen Liniensegmenten besteht, die sich an vier 90° Ecken treffen) werden selten beachtet. Bestehende Computersicht- und Dokumentenbildanalyseverfahren arbeiten auf der schematischen Tafelbenutzerschnittstellen-Analysetask unter normalen Betriebsbedingungen unzulänglich, bei denen sich Benutzer wahrscheinlich nicht besonders um die Präzision und Genauigkeit ihrer schematischen Befehlszeichnungen kümmern. Außerdem müssen wegen der Echtzeitnatur dieser Task die Bildanalyseverfahren als Ganzes inhärent effektiv sein (z. B. Vermeiden von kombinatorischer Komplexität in der Zahl von Markierungen auf der Tafel).

[0019] Eine Aufgabe dieser Erfindung ist es, menschliche Benutzer mit der Fähigkeit zu versehen, Recheneinrichtungen und Umsetzungseinrichtungen durch handgezeichnete Bilder auf einer Tafel zu steuern. Diese Bilder können an jeder Stelle auf der Tafel zu jeder Zeit gezeichnet werden und liefern ein Verfahren zum Auswählen von Steueroperationen von der Tafel.

[0020] Symbolische Tokens basieren auf geometrischen Symbolen, die leicht von Hand gezeichnet und durch räumliche und symbolische Analyseprozeduren erfasst werden können.

[0021] Eine weitere Aufgabe der Erfindung ist es, die Erkennung von Steuerdiagrammen in handgezeichneten Bildern sowie ein Verfahren zum Steuern von symbolischer und geometrischer Funktionalität für den Tafelbenutzer bereitzustellen.

[0022] Die vorliegende Erfindung beschreibt eine neuartige Anwendung von Computersichtverfahren, die die Interpretation von handgezeichneten Befehlen unter einer unbegrenzten Klasse von schematischen Benutzerschnittstellen-Konstruktionen unterstützen. Die Komponenten dieser Schnittstellenkonstruktionen können handgezeichnete Symbole und krummlinige Verknüpfungen umfassen. Die Verfahren dieser Erfindung unterstützen folglich das Erkennen von handgezeichneten Befehlssymbolen und Verfolgen von krummlinigen Verknüpfungen. Die hierin angebotenen Verfahren führen außerdem zu

einem höheren Grad an Robustheit bei der Maschineninterpretation von handgezeichneten Diagrammen als vorher in der Technik an den Tag gelegt wurde.

[0023] Erfindungsgemäß umfasst ein Verfahren zum Identifizieren eines handgezeichneten Befehlsbezeichners, der von Hand auf einer Oberfläche, die keinen Ausgang in elektronischer Form erzeugt, gezeichnet wird, wobei ein elektronisches Kamerasystem ein Bild von handgezeichneten Markierungen auf der Oberfläche erfasst, um eine elektronische Wiedergabe der Oberfläche zu erzeugen:

- a) Bestimmen einer ersten geometrischen Form und einer zweiten geometrischen Form des handgezeichneten Befehlsbezeichners;
- b) Verwenden von räumlicher Analyse von symbolischen Tokens, um die elektronische Wiedergabe der Oberfläche zu analysieren, um die erste geometrische Form zu finden, und
- c) Verwenden von räumlicher Analyse von symbolischen Tokens, um die elektronische Wiedergabe der Oberfläche zu analysieren, um die in der ersten geometrischen Form verschachtelte zweite geometrische Form zu finden.

[0024] In einer Ausführung wird ein Verfahren zum Steuern von Einrichtungen durch Interpretieren von handgezeichneten Markierungen auf einer gescannten Oberfläche bereitgestellt. Das Verfahren umfasst das Bestimmen eines handgezeichneten symbolischen Tokens, der eine Aktion darstellt, Bestimmen, wann das symbolische Token ausgewählt wird, und Durchführen der Aktion, wenn es ausgewählt ist. Das Verfahren sorgt des Weiteren für das Anzeigen eines räumlichen Bereichs der mit dem symbolischen Token verbundenen abgetasteten Oberfläche und für das Durchführen der Aktion in Bezug auf diesen räumlichen Bereich.

[0025] Bestimmte Ausführungen werden in den daran angehängten abhängigen Ansprüchen dargelegt.

[0026] Eine schematische Schnittstelle nutzt bestehende Fähigkeiten des Benutzers im Zeichnen auf der Tafel aus und kann außerdem einen äußerst einfachen Zugriff auf die Grundfunktionalität bieten und schrittweise die größere schematische Komplexität einführen, die zum Feinabstimmen von größerer Funktionalität nötig ist. Schließlich ist eine schematische Schnittstelle, die aus auf der Tafel selbst gezeichneten Markierungen besteht, bestens geeignet, räumliche Steueranweisungen bereitzustellen, z. B. extrahieren eines Bereichs des Bildes.

[0027] Eine schematische Benutzerschnittstelle zu einem Tafel-Umsetzungssystem hängt von der Fähigkeit eines Computerprogramms ab, Markierungen auf der Tafel erfolgreich zu interpretieren. Ein zusätzlicher Vorteil der vorliegenden Erfindung umfasst die Toleranz gegen Veränderlichkeit und von einem menschlichen Benutzer gemachte unechte Markierungen.

[0028] Ausführungen der Erfindung werden nun als

Beispiel mit Bezug auf die begleitenden Zeichnungen beschrieben. Inhalt der Zeichnungen:

[0029] **Fig. 1** zeigt ein allgemeines Oberflächen-Abtastsystem.

[0030] **Fig. 2** ist ein Flussdiagramm, das ein allgemeines Verfahren zum Erzeugen einer Binärwiedergabe einer Tafel aus einem Satz von abgetasteten Bildabschnitten.

[0031] **Fig. 3** zeigt ein Tafelbild mit einer handgezeichneten Taste, die aus zwei verschachtelten Quadraten besteht.

[0032] **Fig. 4** zeigt eine komplexe handgezeichnete Befehlsstruktur mit Linien, die zu symbolischen Textanmerkungen führen.

[0033] **Fig. 5** zeigt ein Bild mit einem ausgewählten Bereich, der handgeschriebene Textzeilen enthält.

[0034] **Fig. 6** zeigt ein Bild mit Abgrenzungslinien, die benutzt werden können, um einen Bereich des Bildes zu bezeichnen.

[0035] **Fig. 7** beschreibt die Grundschriffe, die beim Verwenden der in **Fig. 3** bis **6** illustrierten Merkmale in Frage kommen.

[0036] **Fig. 8** präsentiert die Gesamtfunktionsarchitektur **180** der vorliegenden Erfindung.

[0037] **Fig. 9–14** sind Flussdiagramme, die Schritte in dem Steuermodul der höchsten Task zum Befehlen der im Flussdiagramm von **Fig. 7** beschriebenen allgemeinen Operation zeigen.

[0038] **Fig. 15** ist ein Flussdiagramm, das Schritte zeigt, die zum Identifizieren von Kurvenfragmenten verwendet werden.

[0039] **Fig. 16** zeigt einige typische handgezeichnete verschachtelte Kastentasten (NBB).

[0040] **Fig. 17** illustriert allgemeine Merkmale, die von einer NBB verlangt werden.

[0041] **Fig. 18** illustriert den Abgleich von krummlinigen Tokens.

[0042] **Fig. 19** ist ein Flussdiagramm, das Schritte beim Bestimmen einer NBB zeigt, und ob er ausgewählt wird oder nicht, wobei die Schritte durch **Fig. 20–27** veranschaulicht werden.

[0043] **Fig. 28** ist ein Flussdiagramm, das Schritte zur einfachen Kurvenverfolgung zeigt. Einfache Kurvenverfolgung wird durch **Fig. 29** veranschaulicht.

[0044] **Fig. 30** ist ein Flussdiagramm, das Schritte zur Strahlsuch-Konturverfolgung zeigt. **Fig. 31–33** veranschaulichen Strahlsuch-Konturverfolgung.

[0045] **Fig. 34** ist ein Flussdiagramm, das Schritte zum Verfolgen von Verzweigungskonturen zeigt.

[0046] **Fig. 35** zeigt Beispiele von anderen möglichen Befehlssymbolen.

## ALLGEMEINE SYSTEMBESCHREIBUNG

[0047] **Fig. 1** zeigt ein allgemeines Tafel-Abtastungssystem **50**, in dem die Verfahren der vorliegenden Erfindung eingesetzt werden können. Es wird klar sein, dass die hierin beschriebenen Verfahren auch auf andere Arten von Systemen angewandt werden können. Zum Beispiel können die beschrie-

benen Verfahren verwendet werden, um Steuerbefehle über Markierungen auf Papier in einer Papier-basierten Benutzerschnittstelle zu interpretieren. Die Verfahren können auch als Befehlseingabe-Schnittstelle für Griffel-basierte Rechensysteme verwendet werden. Die Erörterung hierin wird jedoch hauptsächlich in Form des unten beschriebenen Tafel-Abtastungssystems beschrieben.

[0048] Eine ausführlichere Beschreibung der Merkmale und Arbeitsweise des Systems in **Fig. 1** und **2** kann in der gleichzeitig eingereichten EP-A-0701224, die der US-Patentanmeldung Seriennummer 08/303,929 entspricht, gefunden werden.

[0049] Eine Tafel **52** empfängt Markierungen von einem Benutzer **51**. Ein Kamera-Untersystem **54** erfasst ein Bild oder Bilder der Tafel, die einem Computer **56** über ein Netzwerk **58** zugeführt werden. Im Allgemeinen wird die Auflösung einer elektronischen Kamera, z. B. einer Videokamera, nicht ausreichend sein, um ein ganzes Tafelbild mit genügend Einzelheiten zu erfassen, um die Markierungen auf der Tafel klar zu unterscheiden. Deshalb müssen mehrere eingezoomte Bilder von kleineren Unterbereichen der Tafel, genannt "Bildfliesen" unabhängig erfasst und dann zusammengestückelt werden.

[0050] Das Kamera-Untersystem **54** ist auf einem computergesteuerten Schwenk-Neigekopf **55** montiert und wird unter Programmsteuerung sequentiell auf verschiedene Unterbereiche gerichtet, wenn ein Bilderfassungsbefehl ausgeführt wird. Das Kamera-Untersystem **54** kann alternativ eine Anordnung von festen Kameras umfassen, die jeweils auf einen anderen Unterbereich oder Unterbereiche gerichtet werden. Für die Erörterung hierin kann auf das Kamera-Untersystem **54** einfach als Kamera **54** verwiesen werden.

[0051] Die "rohen" Bildfliesen werden sich im Allgemeinen gegenseitig überschneiden, Perspektivverzerrung infolge des achsenversetzten Sichtpunkts der Kamera enthalten sowie gewisse ungleichmäßige Helligkeitsstufen über der Tafel zwischen Vordergrund (geschriebene Markierungen) und Hintergrund (unmarkierte Tafel) infolge von unkontrollierten Beleuchtungsbedingungen und Reflexionen enthalten. Das Flussdiagramm von **Fig. 2** zeigt ein allgemeines Verfahren zum Erzeugen einer Binärwiedergabe aus einem Satz von abgetasteten Bildabschnitten, einschließlich der Kompensation der oben beschriebenen Effekte. In Schritt **100** werden die abgetasteten Bildabschnitte als "Fliesen" erfasst. Jede Fliese ist ein Teil des von einer Tafelkamera, z. B. Kamera **54**, abgetasteten Tafelbildes. Eine Tafel wird als eine Serie von Fliesen erfasst. Die Fliesen überschneiden sich etwas mit benachbarten Fliesen, sodass das ganze Bild ohne "fehlende" Zwischenräume abgetastet wird. Die Stelle jeder Fliese ist aus der Position und Richtung der Kamera auf dem Schwenk-/Neigekopf bekannt, wenn die Fliese abgetastet wird. Die Fliesen können als "Rohbild"- oder "Kamerabild"-Fliesen bezeichnet werden, weil keine Verarbeit-

ung auf ihnen durchgeführt wurde, um sie entweder zu interpretieren oder in dem Digitalbild genau zu lokalisieren. In Schritt **102** wird eine Mittenumgebungs-Verarbeitung auf jeder Kamerabildfliese durchgeführt. Die Mittenumgebungs-Verarbeitung kompensiert die Helligkeitsabweichungen zwischen und innerhalb Fliesen.

[0052] Schritt **104** findet "Landmarken" in sich überschneidenden Fliesen. "Landmarken" werden als Markierungen auf der Tafel bezeichnet, die die richtige Ausrichtung und Überschneidungsposition von benachbarten Fliesen liefern. Landmarken, die in einem Überschneidungsbereich zwischen zwei Fliesen erscheinen, werden benutzt, um diese zwei Fliesen "passend zu machen", um das Bild zu rekonstruieren. [0053] Schritt **106** sorgt für Perspektivverzerrungs-Korrekturen, die globale Landmarken-Fehlpassungsfunktionen optimieren. Dieser Schritt korrigiert Fehler, die im D. R. (dead reckoning) der Fliesenstelle in dem Bild vorkommen. Die Transformation wird durch ein Vertrauen in die Stelle jeder Landmarke gewichtet, die in dem vorherigen Schritt erhalten wird. Schritt **108** führt perspektivische Korrekturen auf allen Fliesen unter Verwendung der in Schritt **106** bestimmten Perspektiv-Transformation durch.

[0054] In Schritt **110** werden die korrigierten Daten für alle Fliesen in ein Graustufen-Tafel-Wiedergabebild geschrieben. In Schritt **112** wird dieses Graustufenbild schwellwertverarbeitet, um eine binäre, elektronisch handhabbare Wiedergabe des Tafelbildes zu erzeugen.

[0055] In der folgenden Erörterung werden die tatsächlichen Schritte des Umwandeln eines Tafelbildes in ein Digitalbild als implementiert angenommen, wie zuvor beschrieben.

#### ALLGEMEINE MERKMALE UND ARCHITEKTUR-ÜBERSICHT

[0056] **Fig. 3-6** veranschaulichen allgemeine Merkmale der vorliegenden Erfindung. Obwohl die Verfahren dieser Erfindung eine unbegrenzte Klasse von Entwurfskonventionen für Benutzer unterstützen, um Parameter schematisch zu spezifizieren und Befehle auszuführen, werden sie zum Zweck der Veranschaulichung der Verfahren der Erfindung mit den in **Fig. 3-6** gezeigten einzelnen Befehlskonventionen demonstriert.

[0057] Des Weiteren wird für eine in der Technik erfahrene Person klar sein, dass, während die Erörterung unten das Agieren auf einer Auswahl in Form des Umsetzens von Teilen des Bildes beschreibt, das System alternativ andere Arten von Aktionen nach Empfang eines Drucktastenbefehls abhängig von den Wünschen des Entwicklers durchführen kann. Außerdem können andere Formen von Tasten möglicherweise für verschiedene Befehle benutzt werden. Dies würde jedoch die Komplexität der Analyse erhöhen und kann die Systemleistung nachteilig beeinflussen.

[0058] **Fig. 3** zeigt ein Tafelbild **150**. Das Herz eines Benutzerbefehls ist in der vorliegenden Ausführung ein Befehlsbezeichner, der eine handgezeichnete Taste umfasst, die aus zwei "verschachtelten" Quadraten – eines direkt in dem anderen – besteht. Diese wird als eine "Verschachtelter-Kasten-Taste" oder NBB **152** bezeichnet. Zeichnen eines "x" oder einer Prüfmarke in der Taste, wie durch Taste **154** gezeigt, leitet die Bildumsetzung ein.

[0059] Natürlich können andere geometrische Formen als Befehlsbezeichner zusätzlich zu den Kästen oder als Ersatz dafür verwendet werden. Zum Beispiel liefern alle der in **Fig. 35** gezeigten geometrischen Formen einen unerschöpflichen Satz von Beispielen von Formen, die, abhängig von den Wünschen des Systemdesigners, geeignete Tastenformen können. Einige Formen, besonders verschachtelte ähnliche Formen, sind leichter zuverlässig zu erfassen. Verschachtelte Kästen haben jedoch einen zusätzlichen Vorteil, dass sie einfach zuverlässig zu konstruieren sind, und machen es auch für einen ungeübten Benutzer leicht, eine Schnittstelle zu erstellen.

[0060] Optional können eine Anzahl von Hilfsparametern oder Anweisungen über krummlinige Linien an die Taste angefügt werden. Diese können zu Bildabgrenzungssymbolen, zu alphanumerischen Anweisungszeichen oder zu Text führen. **Fig. 4** zeigt z. B. eine komplexe NBB **158**, die Linien enthält, die zu symbolischen Textanmerkungen führen, die anweisen, dass die Kopie an einen Drucker "daily planet" zu senden ist, dass eine zu speichernde Bilddatei "text im" heißt, und dass nur rote und blaue Farben umzusetzen sind. Die Analyse des mit diesen krummlinigen Linien verbundenen Bereichs kann durch eine Vielfalt von Verfahren durchgeführt werden. Zum Beispiel liefert die gleichzeitig eingereichte, gemeinsam abgetretene Anmeldung EP-A-0701226, die der US-Patentanmeldung Seriennummer 08/303,962 entspricht, ein Verfahren zum Bestimmen der Segmentierung von Zeilen in dem Bild. Solche Information kann verwendet werden, um Text aus dem Bild zur Erkennung durch ein Texterkennungssystem zu extrahieren.

[0061] **Fig. 5** zeigt ein Bild **160**, einen Teil von "A list" in "Frog and Toad Together" von Arnold Lobel 1972, HarperCollins Publishers. Das Bild **160** hat einen durch NBB **164** ausgewählten Bereich **162**, der von krummlinigen Linien umschlossen ist. In **Fig. 6** ist ein Teil von Bild **166**, gezeigt durch Abgrenzungslinien **168**, mit NBB **170** verbunden. In **Fig. 6** können z. B. diese Abgrenzungslinien benutzt werden, um anzugeben, dass nur der durch Abgrenzungslinien **168** umschlossene oder darin liegende Bereich umzusetzen ist.

[0062] **Fig. 7** beschreibt die bei Verwendung der in **Fig. 3–6** gezeigten Merkmale in Frage kommenden Grundschritte. Schritt **174** findet ein NBB, z. B. NBB **152**. Schritt **175** prüft, ob die NBB ausgewählt ist. Wenn ja, bestimmt Schritt **176** räumliche Bereiche,

die mit der ausgewählten NBB verbunden sind, z. B. Bereich **162** in **Fig. 5**. Schritt **177** bestimmt Textsegmente, die mit der ausgewählten NBB verbunden sind, z. B. Textfolge **159** in **Fig. 4**. Schritt **178**, schließlich, prüft andere oder neue NBBs, für die die Analyse durchzuführen ist.

[0063] **Fig. 8** zeigt die Gesamtfunktionsarchitektur **180** der vorliegenden Erfindung. Das Kernbildanalyseverfahren zum Interpretieren von schematischen Befehlen unter der vorliegenden Erfindung ist räumlich fokussierte Manipulation von "symbolischen Tokens", kompakten symbolischen Bezeichnern von Teilen des Bildes, in diesem Fall Markierungen und Kombinationen von Markierungen. Information über den Inhalt von Bild **182** wird als symbolische Tokens unterhalten, die in der Datenstruktur der Teilstrichtafel (SSB) **184** liegen. Diese symbolischen Tokens dienen als eine Ressource für räumliche Analyseprozeduren **186**, die manchmal neue Tokens zu der Tafel-datenstruktur beisteuern. Auf der obersten Stufe weist ein Task-Steuermodul **190** die räumlichen Analyseprozeduren **186** an, Information über das Bild **182** zu sammeln, um Beschreibungen der Aspekte des Bildes, die für die augenblickliche Task relevant sind, zu erstellen.

[0064] Die Teilstrichtafel-Datenstruktur **184** erleichtert räumliche und geometrische Berechnungen durch Bereitstellen von Indexierung von symbolischen Tokens auf der Basis der räumlichen Lage und des Maßstabs. Diese Tokens sind kompakte symbolische Bezeichner von abstrakten Bildereignissen, z. B. das Vorkommen von Kurven, Ecken, verschachtelten Kästen, Zeichen usw. Die SSB-Datenstruktur **184** wird ferner in Saund, E., "Symbolic Construction of a 2-D Scale Space Image", IEEE TPAMI, V. 12 Nr. 8, August 1990, beschrieben. Zur Bequemlichkeit können in der Praxis mehrere SSB-Datenstrukturen unterhalten werden, wobei jede verschiedene Arten von Informationen hält, die verschiedene Verarbeitungsebenen widerspiegeln.

[0065] Eine große und erweiterbare visuelle Merkmalsbibliothek **188** wird in Verbindung mit den räumlichen Analyseprozeduren **186** unterhalten, mit einer Wissensbasis von prototypischen räumlichen Ereignissen, die durch Platzieren neuer Tokens auf der SSB ausdrücklich bezeichnet werden können. Tabelle 1 listet Typen von wichtigen Analyseprozeduren auf, die in der vorliegenden Erfindung eingesetzt werden. Diese räumlichen Analyseprozeduren unterliegen den Haupt-Bildanalysetasks des Erkennens von Befehlssymbolen, des Verfolgens von krummlinigen Pfaden und des Segmentierens von handgeschriebenem Text.

Tabelle 1

## Wichtige visuelle Analyseprozeduren

[0066] Identifizieren von Tokens nahe der angegebenen Stelle

[0067] Identifizieren von Tokens in angegebener Richtung von gegebener Raumstelle

[0068] Identifizieren von Tokens in Linie mit angegebenem Token

[0069] Tokens mit angegebener Ausrichtung auswählen

[0070] Tokens mit angegebener Krümmung auswählen

[0071] Tokens mit angegebenem Maßstab auswählen

[0072] Paare von Linientokens bezeichnen, die eine Eckenkonfiguration bilden

[0073] Paare von Linientokens bezeichnen, die eine Parallelkonfiguration bilden

[0074] Paare von Linientokens bezeichnen, die eine T-Konfiguration bilden

[0075] Paare von Linientokens bezeichnen, die eine X-Konfiguration bilden

[0076] Paare von Eckentokens bezeichnen, die eine verschachtelte Eckenkonfiguration bilden

[0077] Ketten von In-Linie-Tokens bezeichnen, die einen krummlinigen Bogen bilden

[0078] Gruppen von Tokens bezeichnen, die einen dichten Textbereich bilden

[0079] Bestimmen, ob geeignete verschachtelte Ecken- und Paralleltokens vorhanden sind, um zwei verschachtelte Kästen zu bilden

[0080] Ketten von Tokens verfolgen, die sich Ende-an-Ende treffen

[0081] Zum Beispiel kann eine Prozedur zum Finden von Ecken krummlinige Tokens in einem spezifizierten Bereich der SSB finden und Paare von Tokens bezeichnen, die eine eckenförmige Konfiguration bilden. Für jedes derartige Paar kann ein neues ECKEN-Token erzeugt und an der geeigneten Stelle auf der SSB in der SSB-Datenstruktur **184** platziert werden.

[0082] Das oberste Task-Steuermodul **190** weist die Anwendung von räumlichen Analyseprozeduren **186** entsprechend einer Task-Ausführungsstrategie an, die für die spezifische schematische Benutzerschnittstellen-Spezifikation definiert ist. Verbunden mit dem Task-Steuermodul **190** ist ein Zwischenspeicher, Raumbereich-Cache **192**, der einen Zustand dahingehend unterhält, welche Tastenkästen oder andere Taskstufen-Objekte in dem Bild vermutet werden, sowie Information über Stellen des Bildes, die im Einzelnen untersucht werden müssen, sobald Verarbeitungs-Ressourcen verfügbar werden.

#### TASK-STEUERUNG

[0083] Das in **Fig. 9–14** gezeigte oberste Task-Steuermodul befiehlt die Anwendung von räumlichen Analyseprozeduren, um die im Flussdiagramm von **Fig. 7** beschriebene allgemeine Operation entsprechend einer Task-Ausführungsstrategie zu implementieren, die für die in **Fig. 3–6** gezeigte schematische Benutzerschnittstellen-Spezifikation definiert ist. Weitere Einzelheiten zu spezifischen Analysepro-

zeduren werden in den folgenden Abschnitten beschrieben.

[0084] In **Fig. 9** verarbeitet Schritt **420** das visuelle NBB Aussehen für jede NBB auf der Tafel, wie bezüglich **Fig. 10** beschrieben wird. Schritt **422** sucht nach neuen NBBs auf der Tafel. Dieser Schritt kann periodisch durchgeführt werden, indem alle Tokens auf der Tafel überprüft werden, oder kann für Abschnitte der Tafel durchgeführt werden, wo bekannt ist, dass Aktivität stattgefunden hat.

[0085] Schritt **424** von **Fig. 10** verifiziert, dass die NBB auf der Tafel existiert, und entfernt, wenn die NBB in Schritt **426** nicht mehr vorhanden ist, die NBB in Schritt **428** aus dem Cache und kehrt zurück. Wenn die NBB noch da ist, prüft Schritt **430** den Tastendruckzustand. Wenn die Taste in Schritt **432** neu gedrückt wird, wird die Befehlsparameteranalyse, wie mit Bezug auf **Fig. 11** beschrieben, durchgeführt. Schritt **436** gibt vor dem Rückkehren einen Bild-Umwandeln-Befehl aus.

[0086] **Fig. 11** analysiert mit der NBB verbundene Befehlsparameter. Schritt **438** führt eine räumliche Unterbereich-Indikatoranalyse durch, wie mit Bezug auf **Fig. 12** beschrieben wird. Schritt **440** führt eine Symbolanmerkungs-Analyse durch, wie mit Bezug auf **Fig. 13** beschrieben.

[0087] In **Fig. 12** werden die räumlichen Unterbereichsindikatoren analysiert, um einen mit der NBB verbundenen räumlichen Unterbereich zu bestimmen. Schritt **442** sucht nach räumlichen krummlinigen Unterbereichsindikatorstrichen. Wenn in Schritt **444** räumliche Unterbereichsindikatorstriche gefunden sind, werden die Striche in Schritt **446** verfolgt. Das durch das Verfolgen angegebene räumliche Unterbereichspolygon wird in Schritt **448** der mit dieser einzelnen NBB verbundenen Befehlsparameter-Spezifikation hinzugefügt.

[0088] Eine Bestimmung und Analyse von mit der NBB verbundenen symbolischen Anmerkungsbereichen wird durch die Schritte von **Fig. 13** durchgeführt. Schritt **450** sucht nach krummlinigen Strichen, die von der Bodenmitte der gerade analysierten NBB absteigen. Wenn in Schritt **452** ein Strich gefunden ist, folgt Schritt **454** der krummlinigen Linie bis zu der nächsten Verbindung. Schritt **456** folgt bis zum Ende von sich nach rechts erstreckenden Ästen, und Schritt **458** segmentiert symbolische Anmerkungs-textbereiche heraus, z. B. unter Verwendung des in der gleichzeitig eingereichten, gemeinsam abgetretenen Anmeldung EP A-0701226, die der US-Patentanmeldung Seriennummer 08/303,962 entspricht, beschriebenen Verfahrens. Schritt **460** sendet die extrahierte Bitmap an ein Texterkennungsmodul, und Schritt **456** fügt der mit dieser einzelnen NBB verbundenen Befehlsparameter-Spezifikation symbolische Anmerkungssteuerparameter hinzu.

[0089] Wenn die Verbindung eine T-Verbindung war, kehrt das System in Schritt **464** zurück, um den letzten Abschnitt des krummlinigen Striches zu verfolgen.

[0090] **Fig. 14**, schließlich, sucht nach neuen NBBs auf der Tafel. Schritt **466** gruppiert in Linie liegende Tokens, die der Tafel neu hinzugefügt wurden. Schritt **468** erfasst verschachtelte obere linke Ecken unter den neu hinzugefügten Tokens. In Schritt **470** wird jede verschachtelte obere linke Ecke in dem Cache auf eine NBB geprüft. Wenn in Schritt **472** eine NBB gefunden wird, wird sie in Schritt **474** in dem Cache installiert, und ihre visuelles Aussehen wird in Schritt **420** analysiert.

## VERARBEITUNGSVERFAHREN

[0091] Die folgende Beschreibung und Figuren beschreiben Bildverarbeitungsverfahren, die benutzt werden können, um Tokens zu identifizieren, Kurven und Ecken zu verfolgen und handgeschriebene Textzeilen zu segmentieren, um die Verfahren der Task-Steuerung, wie oben beschrieben, zu implementieren.

### i. Krummlinien-Grundverarbeitung

[0092] Der primitivste Tokentyp wird ein Primitiv-Kurvenfragment genannt, das relativ gerade Abschnitte von Kurven ohne interne Verbindungen oder Kreuzungen mit anderen Kurven darstellt. Die in **Fig. 15** gezeigten Schritte werden zum Identifizieren von Kurvenfragmenten verwendet.

[0093] Schritt **200** verdünnt zuerst die Linien der Zeichnung mittels morphologischer Erosion, ein den Anwendern von Computervision wohl bekanntes Verfahren. Schritt **202** erfasst Kurvenverbindungen mittels einer morphologischen Nit-Miss-Transformation mit Masken, die Verbindungskonfigurationen von  $3 \times 3$  Pixeln widerspiegeln.

[0094] Schritt **204** verfolgt Kuren zwischen Verbindungen unter Verwendung eines von vielen Kurvenverfolgungsverfahren, die in der Literatur von Computervision bekannt sind. Schritt **206** erfasst Ecken auf Kurven. Verfahren dafür sind ebenfalls in der Computervision-Literatur vorhanden.

[0095] Schritt **208** bricht Kurven an Ecken, um elementare Fragmente übrigzulassen, die als Primitiv-Kurvenfragmenttyp-Tokens zu bezeichnen sind, die in die SSB-Datenstruktur geschrieben werden.

[0096] Diese Verarbeitungsschritte werden weiter in der mitanhängigen, mitabgetretenen Patenanmeldung Seriennummer 101,646 "Method for Dynamically Maintaining Multiple Structural Interpretations in Graphics Systems" von Saund et al., hierin durch Verweis eingeschlossen, beschrieben.

### ii. Befehlssymbole

[0097] In der vorliegenden Erfindung nehmen Befehlssymbole die Form von verschachtelten Kästen und zugehörigen krummlinigen Linien an, wie bezüglich **Fig. 3–6** beschrieben. Tokens werden wie vorangehend beschrieben identifiziert, wobei jedes eine

Stelle, Ausrichtung, Länge und Krümmung hat. In dem Raum aller möglichen Positionen kann jedes Paar von Tokens gemessen werden, um zu bewerten, eine wie gute Ecke sie bilden. (Dies Bewertung "wie gut eine Ecke ist" kann auf Schwellenparametern basieren, die vom Systemdesigner festgelegt werden.)

[0098] In der vorliegenden Ausführung werden, wie oben erklärt, verschachtelte Kästen verwendet, um Befehlstasten zu bezeichnen. Die unten beschriebenen Verfahren zeigen nur die Erfassung von NBB-Befehlssymbolen, aber das Verfahren lässt sich ohne weiteres auf eine breite Klasse von möglichen Befehlssymbolformen, wie z. B. die in **Fig. 35** gezeigten, erweitern.

[0099] Eine NBB kann als zwei Kästen beschrieben werden, die ineinander verschachtelt sind. Symbolische Tokens bestehen gewöhnlich aus anderen kleineren Untertokens. **Fig. 17** zeigt ein Beispiel von Untertokens, die ein symbolisches verschachteltes Kastentasten-Token bilden. Eine NBB **210** wird errichtet, wenn Verschachtelte-Ecken-Untertokens, z. B. **212**, und Parallellinien-Untertokens, z. B. **214**, in geeigneter Nähe und Ausrichtung in Bezug auf eine hypothetisch angenommene Tastenmitte **216** gefunden werden. Benutzer sorgen sich jedoch nicht sonderlich um die Präzision und Genauigkeit der schematischen Befehlszeichnungen. **Fig. 16** zeigt einige typische handgezeichnete verschachtelte Kastentasten, die von einem Benutzer eingegeben werden können, um einige der Schwierigkeiten beim Interpretieren einer schematischen Benutzerschnittstelle in typischen Computervisionssystemen zu veranschaulichen. Zum Beispiel können sich einige Ecken, wie **262** und **264**, nicht ganz treffen, ebenso wie Ecken **274** und **286**, die einige Linien haben, die sich über die Ecken hinaus erstrecken. Einige Linien, z. N. **280**, können mehr gekrümmt als gerade sein oder können, wie **282**, Unterbrechungen enthalten. Die Verfahren der vorliegenden Erfindung erlauben es, alle in **Fig. 16** gezeigten NBBs richtig zu interpretieren.

[0100] **Fig. 19** beschreibt die Schritte beim Bestimmen einer NBB, und ob sie ausgewählt wird oder nicht, mit den durch **Fig. 20–27** veranschaulichten Schritten. Schritt **222** beginnt mit dem Finden einer Anwärterstelle. In der vorliegenden Ausführung wird nach einer "oben links" verschachtelten Ecke gesucht. Filter können auf den verschiedenen Eckenpaaren benutzt werden, um alle Tokens zu entfernen, die nicht Teil einer oberen linken Ecke einer NBB sein können. Obere linke Ecken, die – eine fast in der anderen – verschachtelt sind, werden ausgewählt. Die Mitte eines möglichen verschachtelten Begrenzungskastens, die durch die Stelle und Größe der verschachtelten oberen linken Ecke angenommen wird, wird als eine Anwärterstelle gewählt. In **Fig. 20** ist die Stelle **282** die hypothetische Tastenstelle für eine vermutete NBB und wird als eine Anwärterstelle ausgewählt.

[0101] Schritt **224** gewinnt alle Primitiv-Kurvenfrag-



ment-Tokens, die in einem vorbestimmten maßstabsnormalisierten Abstand von der Anwärterstelle liegen, wie in **Fig. 21** gezeigt.

[0102] Schritt **226** bildet Linearstrich-Tokens durch Gruppieren von Primitiv-Kurvenfragment-Tokens, die klar miteinander in Linie sind, gezeigt in **Fig. 22**. Um "klar ausgerichtete" Tokens zu bestimmen, werden Tokens paarweise untersucht. Wie in **Fig. 18** gezeigt, gibt es für allein genommene Tokenpaare, z. B. Tokens **257** und **258**, die sich Ende-an-Ende mit der gleichen Ausrichtung treffen, kaum eine Frage bezüglich ihrer Ausrichtung. Wenn sich ein Token einem anderen in einem leichten Winkel nähert, wie z. B. Token **259** und Token **258**, bringt dies jedoch eine Mehrdeutigkeit ein, weil das Token **258** entweder mit Token **257** oder **259** in Linie gebracht werden könnte. Das System muss daher auf Paare von Tokens schauen und sich dann in der Nachbarschaft von ausgerichteten Paaren umsehen, um irgendwelche anderen Anwärter zum Ausrichten zu finden. Wenn das "zweitbeste" Token viel weniger ausgerichtet ist als das Erste, und wenn die ersten zwei Tokens gegenseitig die "beste" Ausrichtung sind, sind die ersten zwei klar ausgerichtet. Wenn jedoch der Zweitbeste nur etwas weniger gut ausgerichtet ist, können alle drei Tokens nicht als "klar ausgerichtet" bestimmt werden.

[0103] Zu **Fig. 19** zurückkehrend bildet Schritt **228** Rundbogentyp-Tokens durch Gruppieren von Primitiv-Kurvenfragment-Tokens, die entlang einem runden Bogen miteinander in Linie liegen, wie in **Fig. 23** gezeigt. Anders als Linearstrich-Tokens können sich Rundbogen-Tokens überschneiden und Unterstützung auf der Stufe von Primitiv-Kurvenfragment-Tokens erhalten.

[0104] Schritt **230** bildet Eckentyp-Tokens, gezeigt in **Fig. 24**, durch Finden von Paaren von Linearstrich- und/oder Rundbogen-Tokens, die eine Eckenkonfiguration bilden. Um eine Ecke zu unterstützen, muss ein Rundbogen-Token nur einen flachen Bogen bilden. In Schritt **232** werden verschachtelte Eckentyp-Tokens durch Finden von verschachtelten Paaren von Eckentoken gebildet. In **Fig. 25** zeigen gestrichelte Linien **294** und **295** in verschachtelte Ecken gruppierte Tokens.

[0105] Schritt **234** bildet Parallellinientyp-Tokens durch Finden von parallel liegenden Paaren von Linearstrich- und/oder Rundbogen-Tokens. Gestrichelte Linien **296** und **297** in **Fig. 26** zeigen Parallellinientyp-Tokens.

[0106] In Schritt **236** prüft das System, ob verschachtelte Ecken- und Parallellinien-Tokens an ungefähren Stellen und Ausrichtungen gefunden werden, die mit dem Vorhandensein von zwei verschachtelten Quadraten vereinbar sind. Wenn ja, wird in Schritt **240** eine neue Verschachtelter-Kasten-Taste oder NBB-Token gebildet. Wenn ein NBB gefunden wurde, prüft Schritt **242** das Vorhandensein von Linearstrich- und/oder Rundbogen-Tokens, die eine "X"- oder Prüfmarken-Konfiguration bilden, an der unge-

fähr geeigneten Stelle in Bezug auf die Stelle und Größe der NBB, wie in **Fig. 27** gezeigt. Alternativ könnten andere Symbole, wie z. B. einfache Schrägstriche, auch benutzt werden, um Tasten auszuwählen. Die Flexibilität der Parameter erlaubt dem Systemdesigner eigenes Ermessen beim Wählen von spezialisierten Diagrammen für jedes beliebige System, auf dem die Benutzerschnittstelle verwendet werden soll.

[0107] Dieser Token-Gruppierungsweg unterscheidet sich von modellbasierten Übereinstimmungslösungen, die herkömmlich in der Computervision-Gemeinschaft eingesetzt werden, und ist erfolgreich im Erkennen von ungenau gezeichneten Befehlssymbolen in Anwesenheit von verwirrenden Daten, wie z. B. Prüfmarken und die krummlinigen Linien, die die verschachtelten Kastentasten-Formen verbinden.

### iii. Verfolgen von krummlinigen Pfaden für räumliche Unterbereich-Analyse

[0108] Krummlinige Pfade sind nützlich in schematischen Benutzerschnittstellen zum Einkreisen von Bereichen und zum Bilden von Links zwischen Befehlssymbolen und anderen Anweisungen, z. B. Textanmerkungen. Der dieser Erfindung zugrunde liegende Grund-Tokengruppierungsweg, oben beschrieben, unterstützt das Verfolgen von einfachen krummlinigen Pfaden, krummlinigen Pfaden, die von anderen Linien gekreuzt werden, abzweigende krummlinige Pfade und unterbrochene krummlinige Pfade.

[0109] Kurvenverfolgung macht Gebrauch von Primitiv-Kurvenfragment-Tokens und Kurvenende-Tokens, die mit jedem Ende jedes Primitiv-Kurvenfragments verbunden sind.

[0110] Die Eingabe in eine Kurvenverfolgungsroutine besteht aus einem Primitiv-Kurvenfragment **312** in **Fig. 28** und einem seiner zugehörigen Kurvenenden **314**, das das aktuelle Anfangsende genannt wird. Einfache Kurvenverfolgung arbeitet wie durch das Flussdiagramm in **Fig. 28** gezeigt und durch **Fig. 29** veranschaulicht.

[0111] In Schritt **300** wird das andere Ende **316** von dem Anfangsende **314** des Primitiv-Kurvenfragments **312** identifiziert. Dieses wird das aktuelle ferne Ende genannt.

[0112] Schritt **302** lokalisiert Kurvenenden, die zu anderen Primitiv-Kurvenfragmenten nahe dem aktuellen fernen Ende gehören. In **Fig. 29** werden Kurvenenden **317** und **318** erfasst.

[0113] Schritt **304** prüft die Ausrichtung der anderen Primitiv-Kurvenfragmente mit dem aktuellen Primitiv-Kurvenfragment-Token basierend auf der relativen Stelle und Ausrichtung des Kurvenendes. Wenn keine Segmente deutlich ausgerichtet sind, kann eine "Strahlsuche" durchgeführt werden, wie in **Fig. 30** beschrieben wird.

[0114] Schritt **308** wählt aus, welches von Primitiv-Kurvenfragmenten, wenn vorhanden, klar mit dem aktuellen Primitiv-Kurvenfragment-Token ausgerich-

tet ist, und Schritt **310** macht das in Linie liegende Primitiv-Kurvenfragment das aktuelle Kurvensegment und das in Linie liegende Kurvenende das neue aktuelle Anfangsende und kehrt zu Schritt **300** zurück, um das Verfolgen fortzusetzen. In **Fig. 29** liegt das Primitiv-Kurvenfragment **319** in Linie mit dem Primitiv-Kurvenfragment **312**, Primitiv-Kurvenfragment **319** wird das aktuelle Kurvensegment und Kurvenende **318** wird das neue aktuelle Anfangsende.

[0115] Das Verfolgen von unterbrochenen krummlinigen Konturen, z. B. Verfolgen der Ecken eines Polygons, macht Gebrauch von dem obigen einfachen Kurvenverfolgungsalgorithmus. Wenn jedoch ein Kontursegment keine klare Fortsetzung über sein fernes Ende, wie in Schritt **304** bestimmt, hinaus hat, wird eine Strahlsuche verwendet, um die SSB nach Tokens abzusuchen, die mit dem aktuellen Token genügend ausgerichtet sind, damit der Pfad einen Sprung macht. Das Flussdiagramm von **Fig. 30** zeigt dieses Konturverfolgen in Bezug auf **Fig. 31–33**.

[0116] In Schritt **304** von **Fig. 28** wurde festgestellt, dass es keine deutlich mit dem aktuellen Primitiv-Kurvenfragment ausgerichteten Segmente gab. Zum Beispiel hat im Segment **332** von **Fig. 31** das ferne Ende von NBB **330** kein klar ausgerichtetes Segment in seiner unmittelbaren Nachbarschaft. Desgleichen hat Segment **334** kein anderes klar ausgerichtetes Segment in der Nachbarschaft seines unteren Endes. Schritt **320** projiziert dann ein Segment von dem aktuellen fernen Ende in der Ausrichtung des aktuellen Primitiv-Kurvenfragments, und projiziert dann in Schritt **322** ein "Taschenlampenstrahl"-Muster von dem aktuellen fernen Ende, das das projizierte Segment umgibt. In **Fig. 32** wird die Strahlprojektion **336** um das projizierte Segment **335** herum projiziert, von dem aktuellen fernen Ende von Segment **332**. Desgleichen wird die Strahlprojektion **340** um das projizierte Segment **339** herum projiziert, das mit dem aktuellen fernen Ende von Segment **334** ausgerichtet ist.

[0117] In Schritt **324** werden Kurvenende-Tokens in der projizierten Fläche lokalisiert. Wie in **Fig. 32** gezeigt, wird dem Kurvenende **338** durch die Strahlprojektion **336** begegnet werden. Die Strahlprojektion **340** wird mehreren Kurvenende-Segmenten auf ihrem Weg – Segment **341–344** begegnen, die alle in die Strahlprojektion fallen.

[0118] Schritt **326** prüft die lokalisierten Tokens auf ausreichende Ausrichtung. Im Fall der Strahlprojektion **340** ist das Segment **341** klar nicht in der richtigen Ausrichtung, um mit dem Segment **334** in Linie zu sein. Desgleichen würde das Segment **341** kein Anwärter auf Ausrichtung sein, da das Ende des Segments, das dem Segment **334** am nächsten ist, bereits mit einem anderen Segment verbunden ist. Sowohl **342** als auch **344** sind richtig ausgerichtet, aber Segment **342** liefert eine bessere Ausrichtung mit Segment **334** als Segment **344**. Schritt **338** wählt das am besten ausgerichtete Primitiv-Kurvenfragment aus und macht das neue Segment das aktuelle Primi-

tiv-Kurvenfragment für das weitere Verfolgen.

[0119] **Fig. 33** zeigt die entstandene NBB **350** und ihren zugehörigen räumlichen Unterbereich, dargestellt als Polygon **348** und durch das oben beschriebene Strahlsuchen bestimmt. Ein zweiter räumlicher Unterbereich **352**, verbunden mit einer anderen NBB **354**, kann ebenfalls unter Verwendung der Strahlsuche gefunden werden.

[0120] Das Verfolgen von verzweigenden Konturen, gezeigt im Flussdiagramm **34**, umfasst eine einfache Modifikation des einfachen Konturverfolgungsalgorithmus. In Schritt **360** wird das aktuelle ferne Ende des aktuellen Primitiv-Kurvenfragments identifiziert. Schritt **362** lokalisiert Kurvenenden, die zu anderen Primitiv-Kurvenfragmenten nahe dem aktuellen fernen Ende gehören. Schritt **364** sucht nach geeigneten nahen und ausgerichteten Kurvenende-Tokens, die eine Abzweigung von dem momentan verfolgten Weg bilden, und in Schritt **366** wird eine neue unabhängige Spur für alle angetroffenen Zweige nach den mit Bezug auf **Fig. 28** beschriebenen Verfolgungsschritten hervorgebracht.

[0121] Solche Verfolgungsverfahren führen zur Identifikation von Stellen in dem Bild, die der Anfang einer Zeile von Text sind. Textsegmentierung grenzt dann den Bereich des Bildes ab, der den Text enthält. Ein Verfahren zum Segmentieren von Text wird in der vorangehend zitierten gleichlaufend eingereichten Anmeldung EP-A-0701226, die der US-Patentanmeldung Seriennummer 08/303,962 entspricht, beschrieben. Aus dieser Segmentierung ist es ein Leichtes, den entsprechenden Bereich der ursprünglichen Bitmap zu extrahieren, der an ein getrenntes Handschrift-Erkennungsmodul zum Erkennen des Texts zu senden ist.

## Patentansprüche

1. Verfahren zum Identifizieren eines handgezeichneten Befehlsbezeichners (**152**), der von Hand auf einer Oberfläche (**52**) gezeichnet wird, die keinen Ausgang in elektronischer Form erzeugt, wobei ein elektronisches Kamerasystem (**54**) ein Bild von handgezeichneten Markierungen auf der Oberfläche erfasst, um eine elektronische Wiedergabe der Oberfläche zu erzeugen, wobei das Verfahren umfasst:
  - a) Bestimmen einer ersten geometrischen Form und einer zweiten geometrischen Form des handgezeichneten Befehlsbezeichners;
  - b) Verwenden von räumlicher Analyse von symbolischen Tokens, um die elektronische Wiedergabe der Oberfläche (**52**) zu analysieren, um die erste geometrische Form zu finden, und
  - c) Verwenden von räumlicher Analyse von symbolischen Tokens, um die elektronische Wiedergabe der Oberfläche (**52**) zu analysieren, um die in der ersten geometrischen Form verschachtelte zweite geometrische Form zu finden.

2. Verfahren nach Anspruch 1, wobei die erste

geometrische Form und die zweite geometrische Form rechteckig sind.

3. Verfahren nach Anspruch 1, wobei der Schritt b) die folgenden Unterschritte umfasst:

b1) Analysieren der elektronischen Wiedergabe der Oberfläche (52), um einen Teil der ersten geometrischen Form zu finden, und

b2) Analysieren der elektronischen Wiedergabe der Oberfläche (52), um andere Teile der ersten geometrischen Form zu finden, die mit dem Teil der ersten geometrischen Form verbunden sind, und

wobei der Schritt c) die folgenden Unterschritte umfasst:

c1) Analysieren der elektronischen Wiedergabe der Oberfläche (52), um einen Teil der zweiten geometrischen Form in unmittelbarer Nähe zu der ersten geometrischen Form zu finden, und

c2) Analysieren der elektronischen Wiedergabe der Oberfläche, um andere Teile der zweiten geometrischen Form zu finden, die mit dem Teil der zweiten geometrischen Form verbunden sind.

4. Verfahren nach Anspruch 1, wobei die erste geometrische Form ein Rechteck ist, und die zweite geometrische Form ein anderes Rechteck ist, das in der ersten geometrischen Form verschachtelt ist.

5. Verfahren nach Anspruch 1, wobei der Schritt b) die folgenden Unterschritte umfasst:

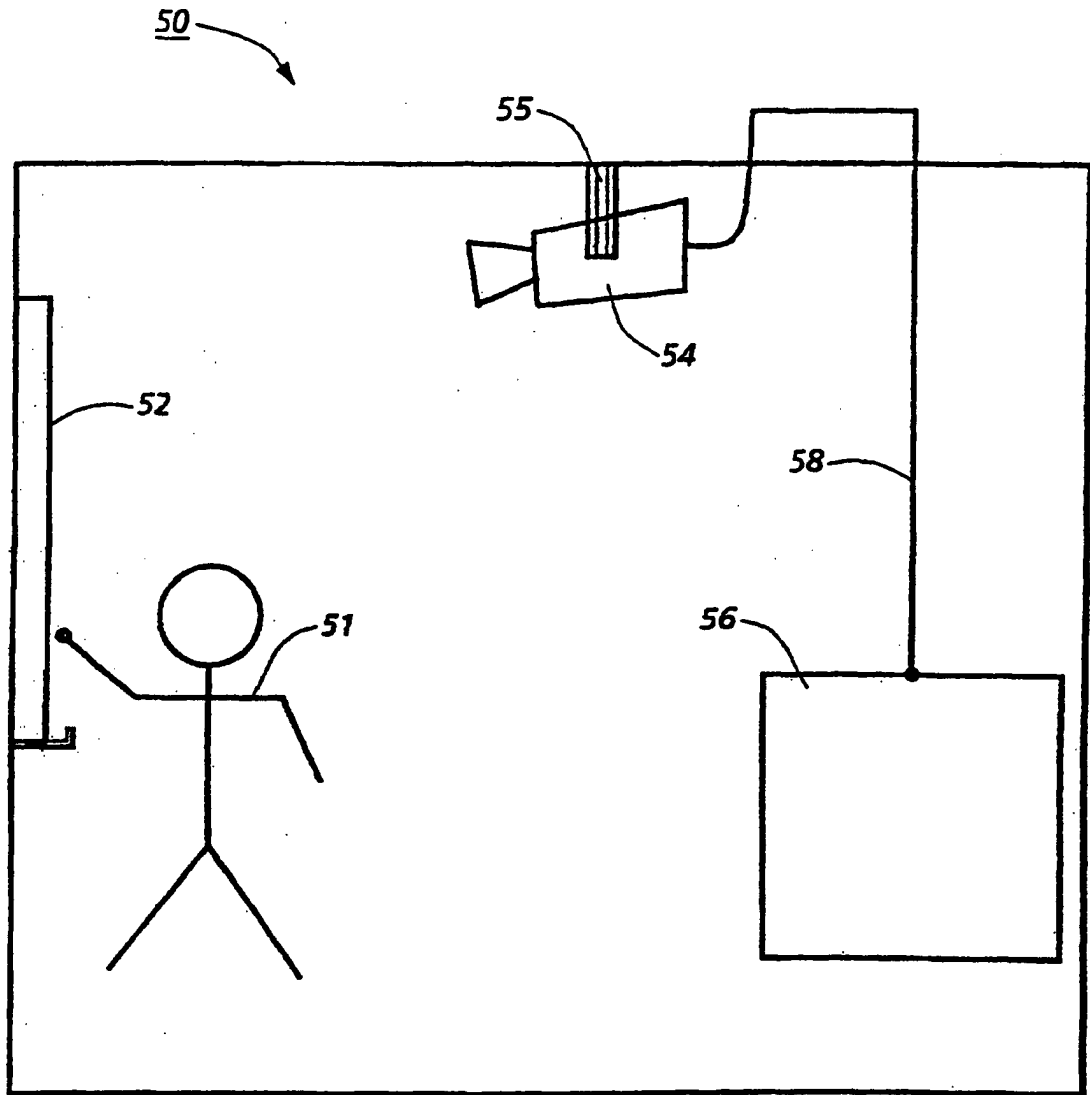
b1) Analysieren der elektronischen Wiedergabe der Oberfläche, um alle handgezeichneten Linien in einem Bereich der Oberfläche zu finden, und Erzeugen eines Tokens, um jede gefundene handgezeichnete Linie darzustellen,

b2) Bestimmen der Tokens, die eine obere linke Ecke bilden,

b3) Isolieren von Tokens in einer Nachbarschaft der oberen linken Ecke, und

b4) Analysieren der isolierten Tokens in der Nachbarschaft der oberen linken Ecke, um weitere Ecken zu finden, die eine obere rechte Ecke, eine untere rechte Ecke und eine untere linke Ecke darstellen.

Es folgen 23 Blatt Zeichnungen



**Fig. 1**

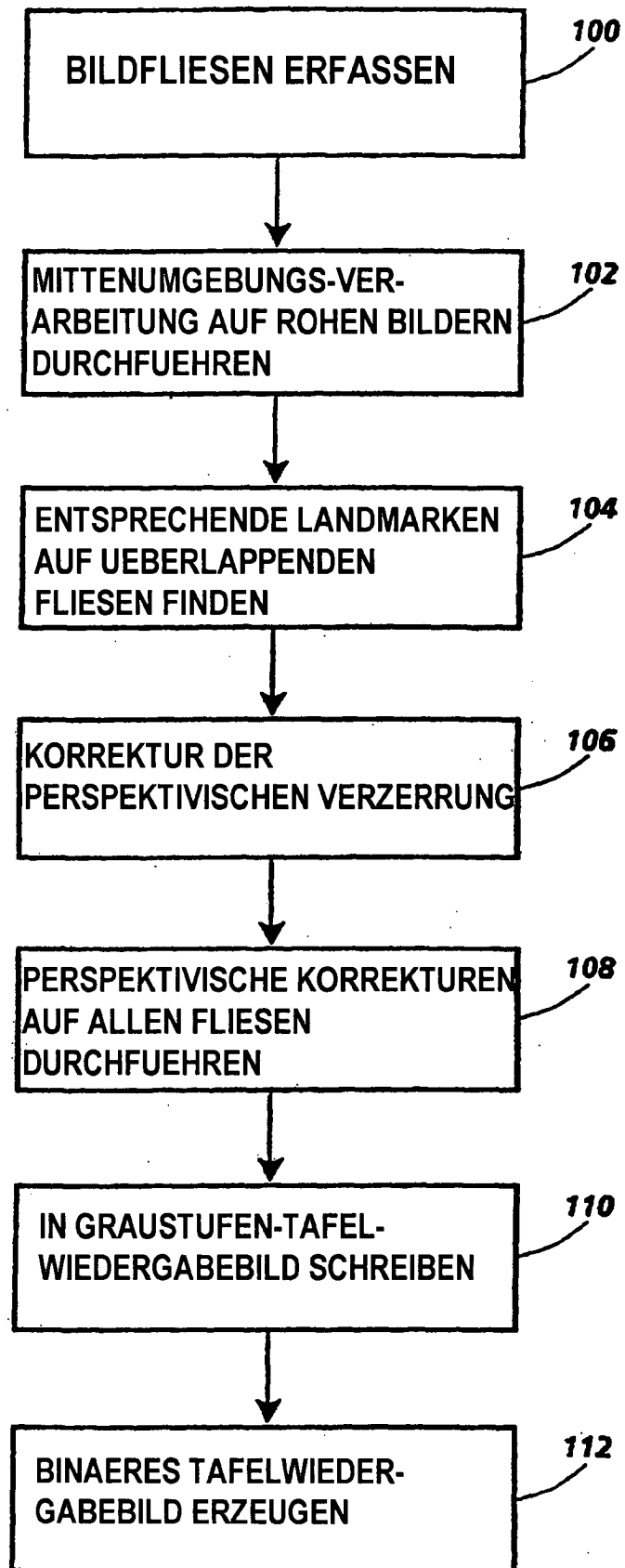
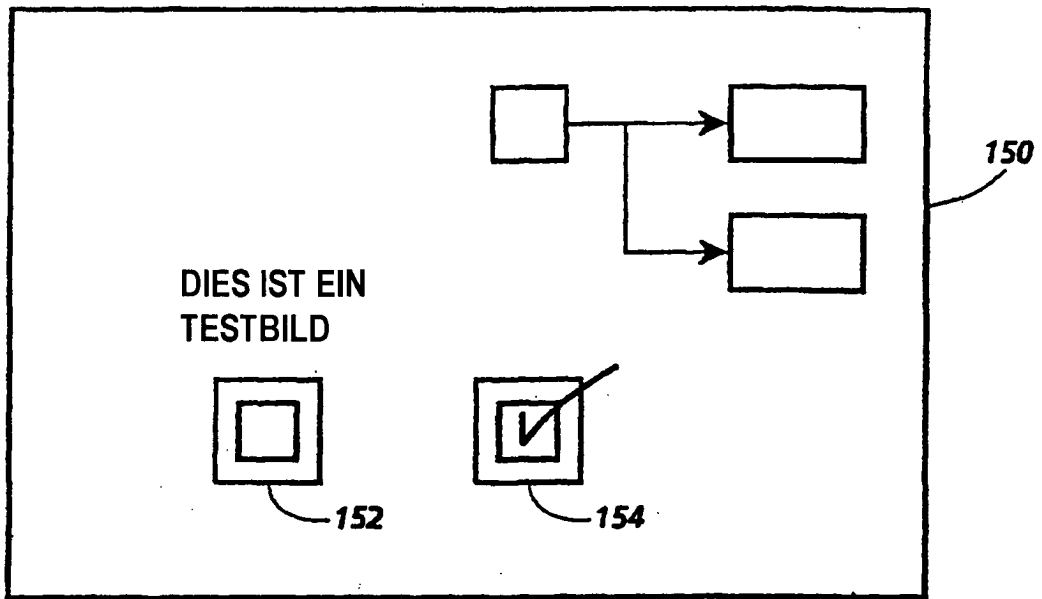
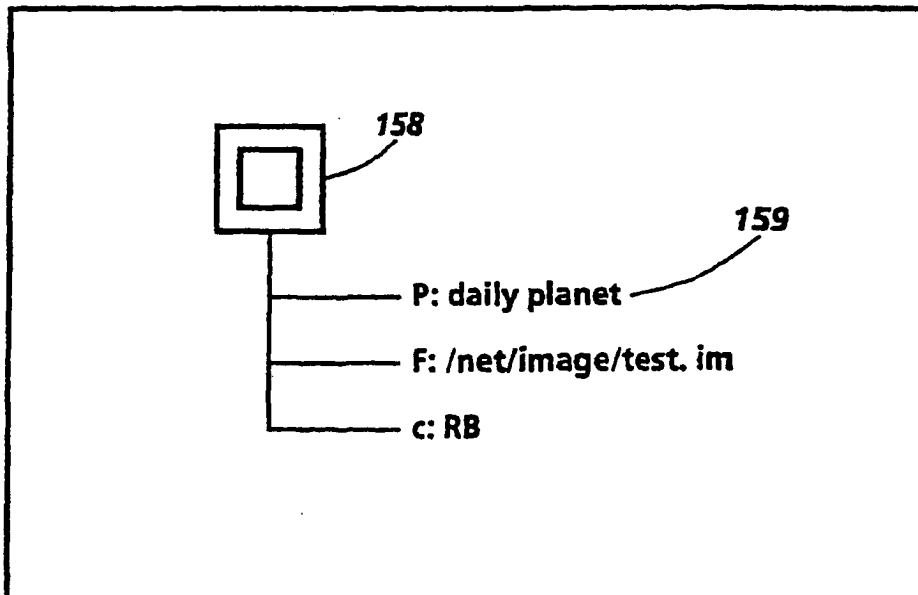


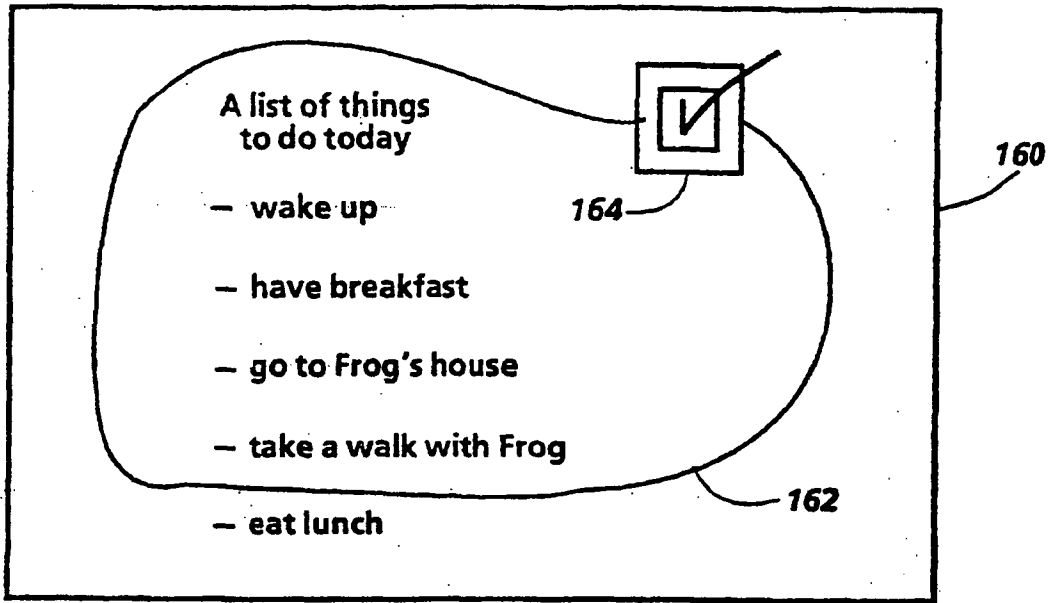
FIG. 2



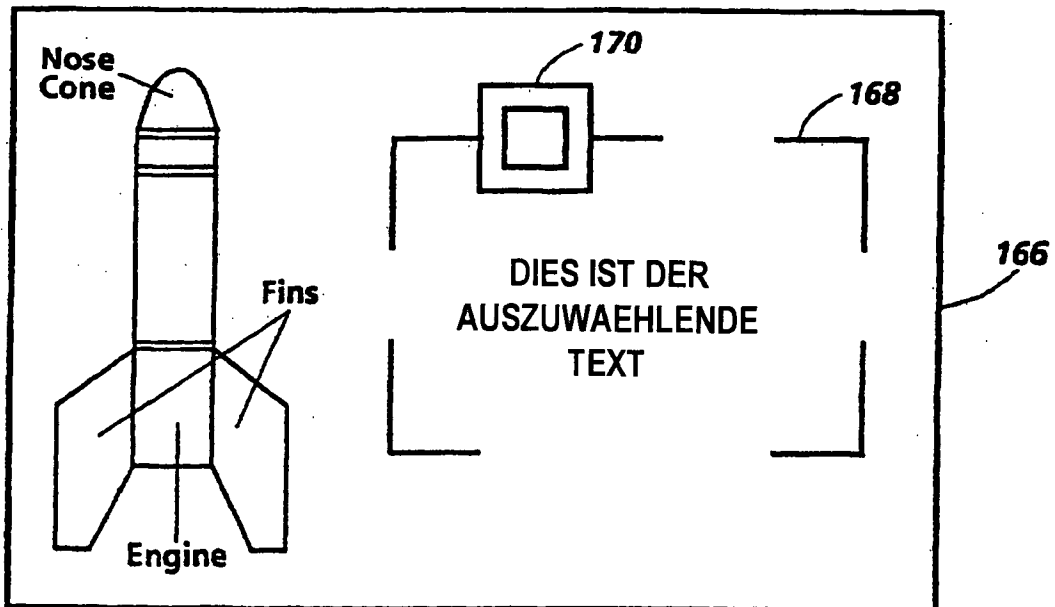
**Fig. 3**



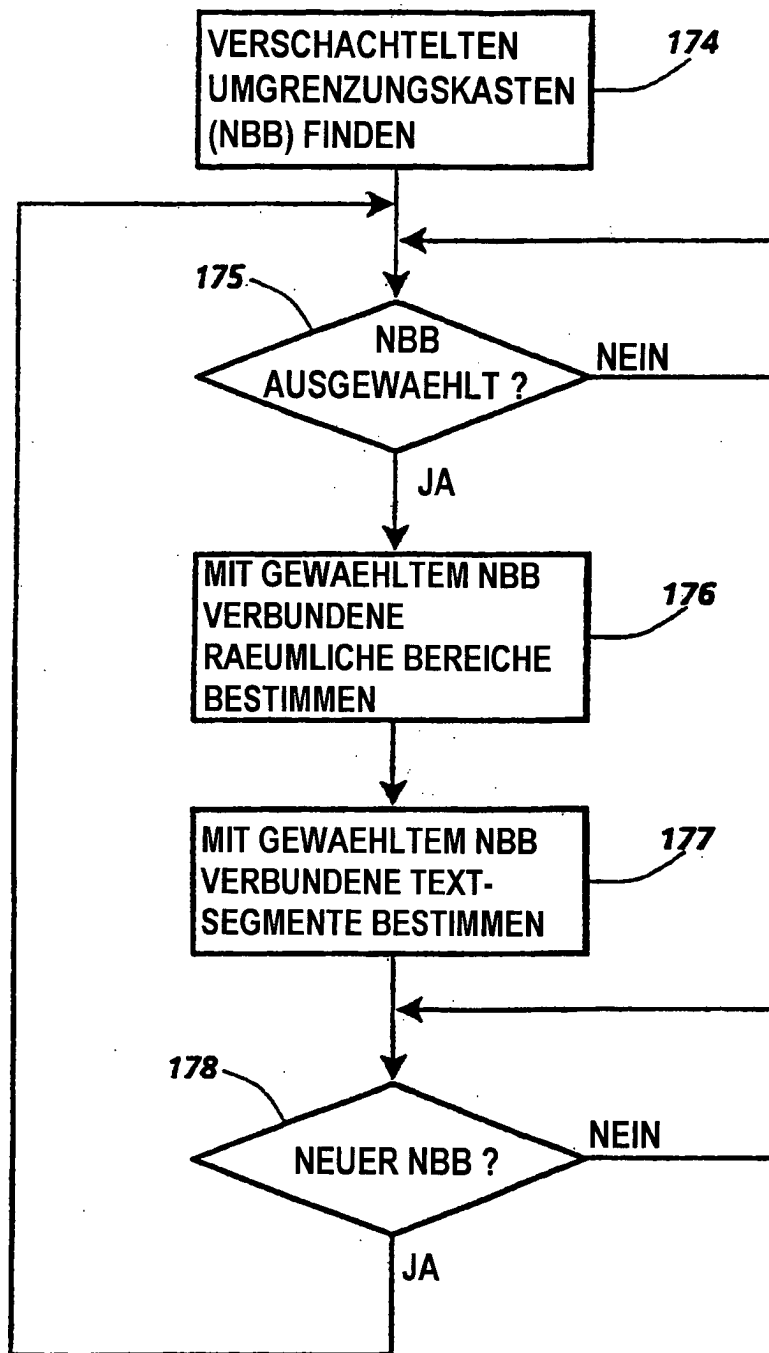
**Fig. 4**



**Fig. 5**

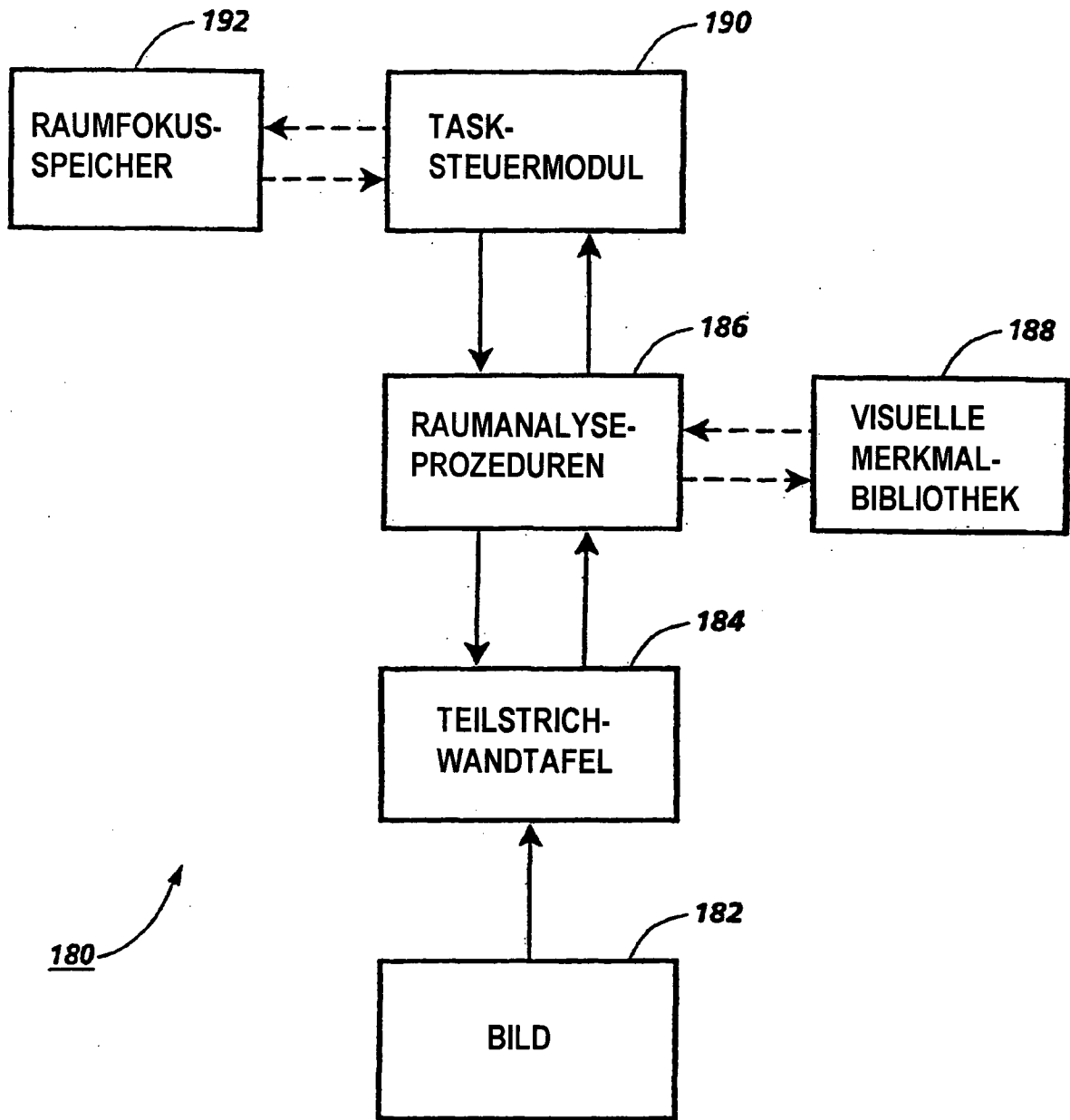


**Fig. 6**

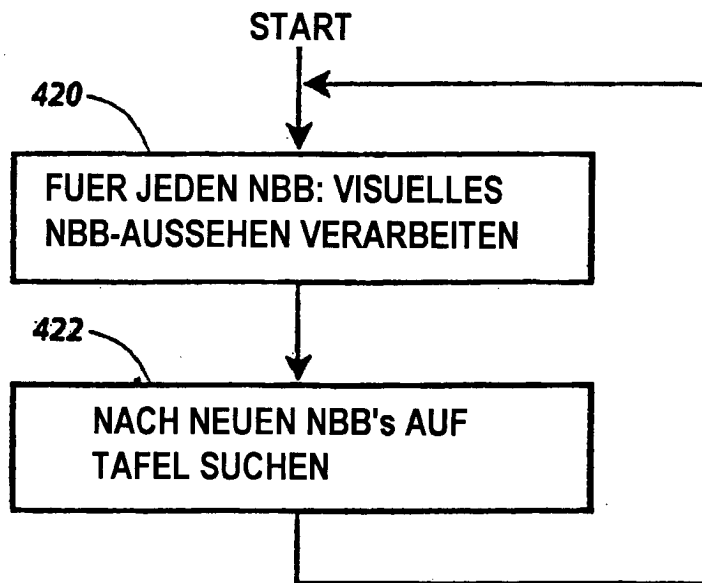


**Fig. 7**

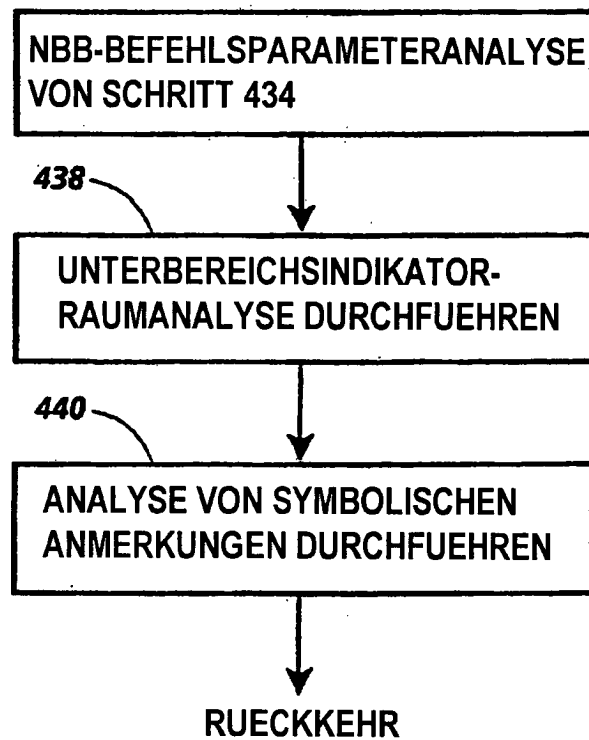




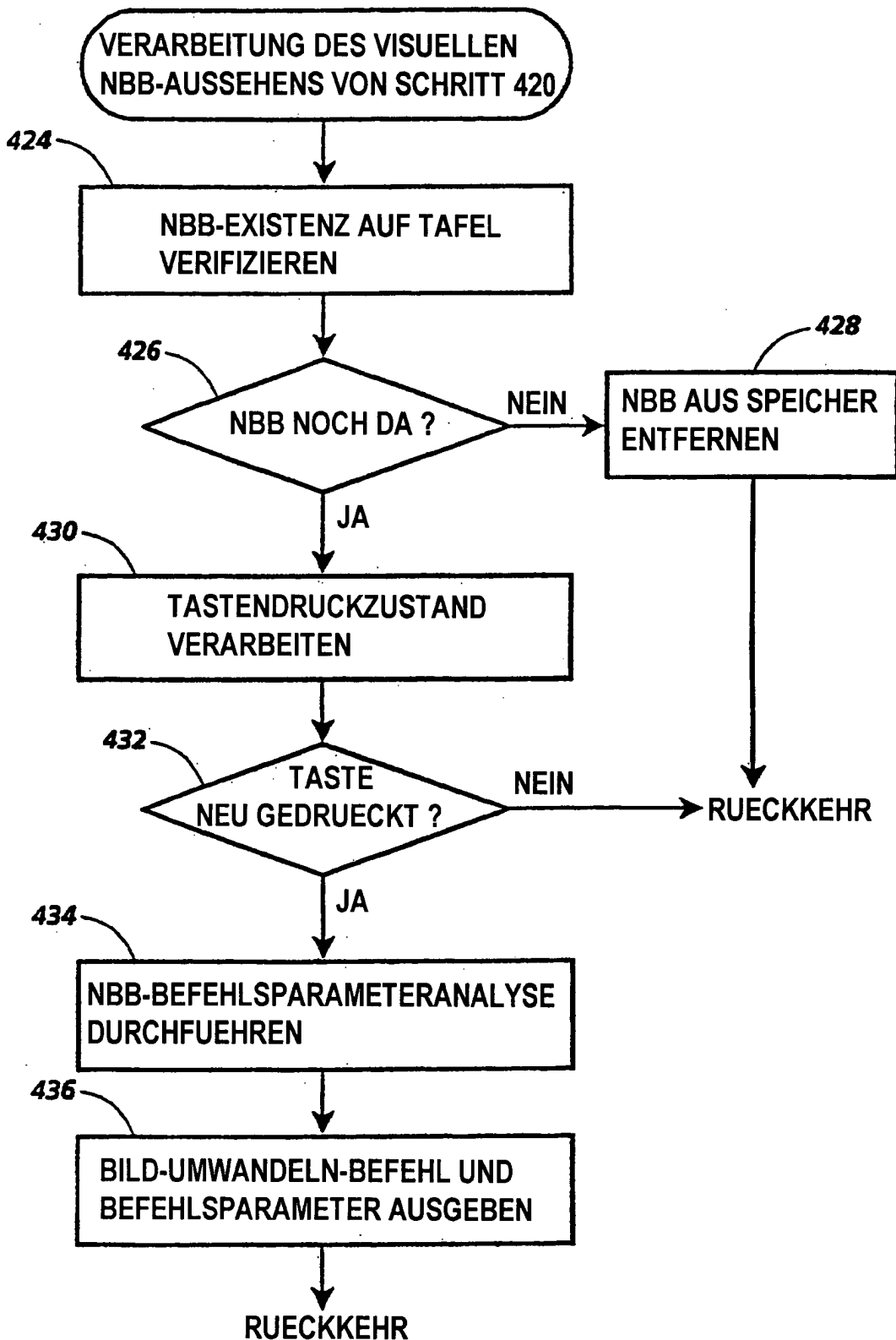
**Fig. 8**



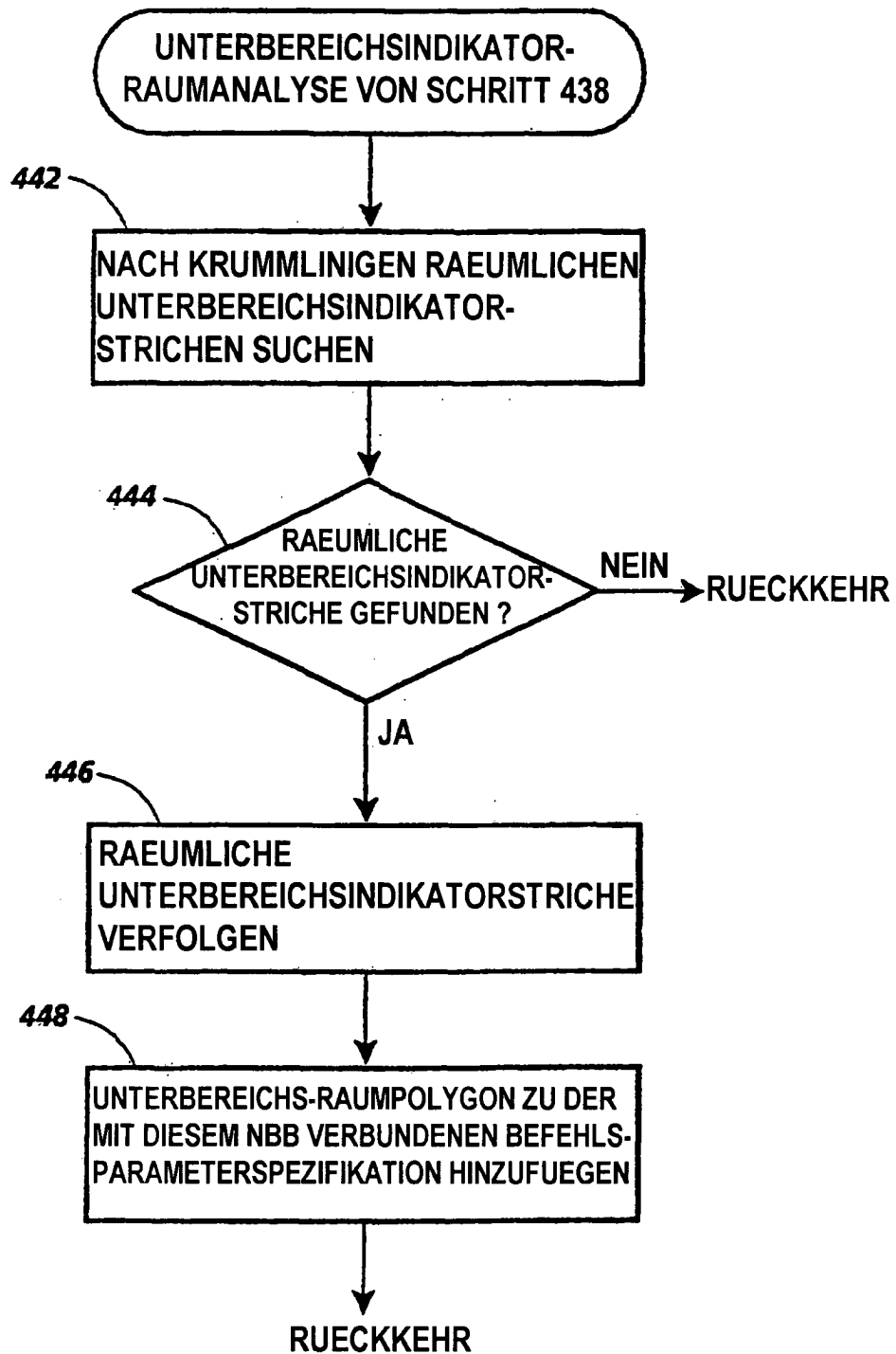
**Fig. 9**



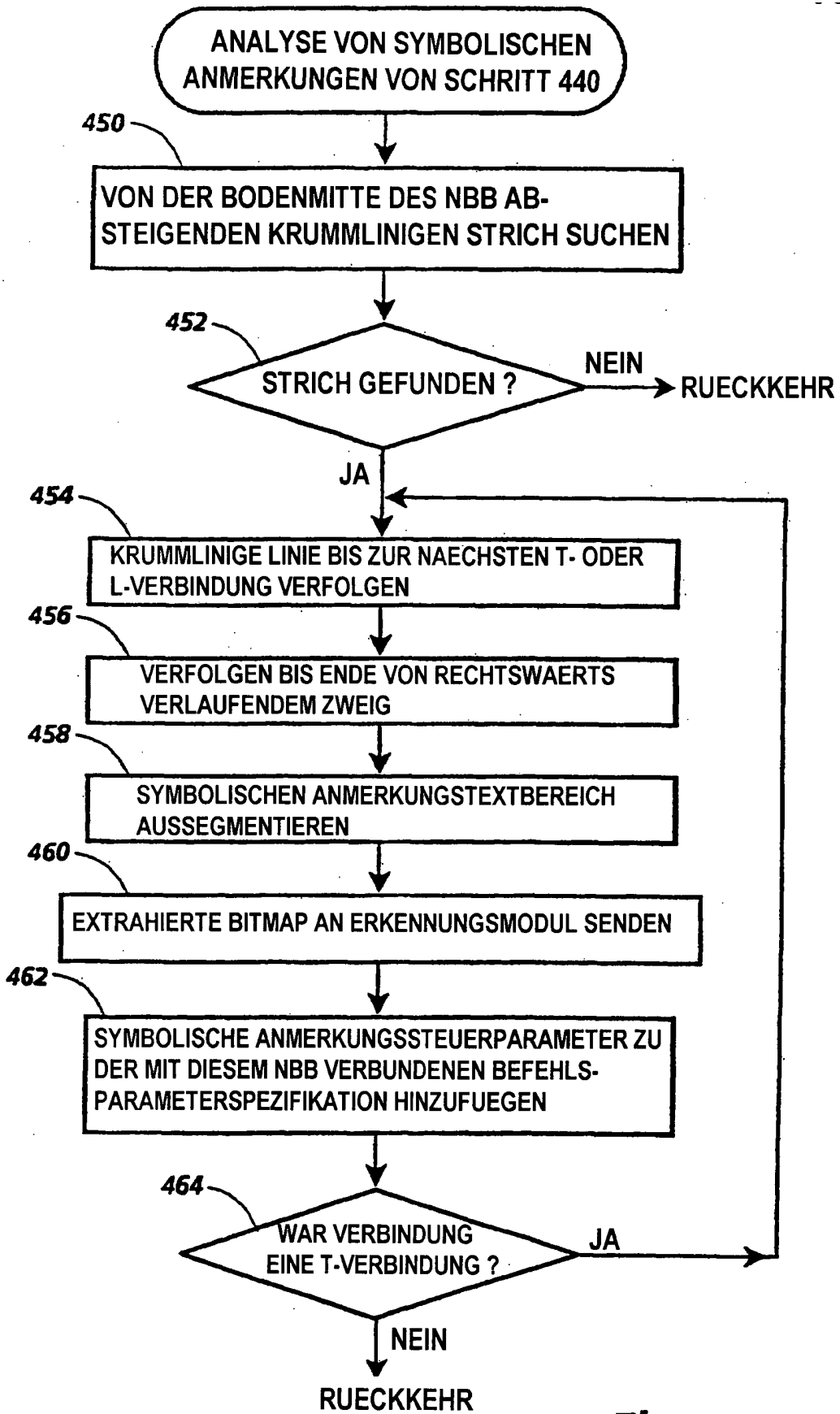
**Fig. 11**



**Fig.10**



**Fig.12**



**Fig.13**

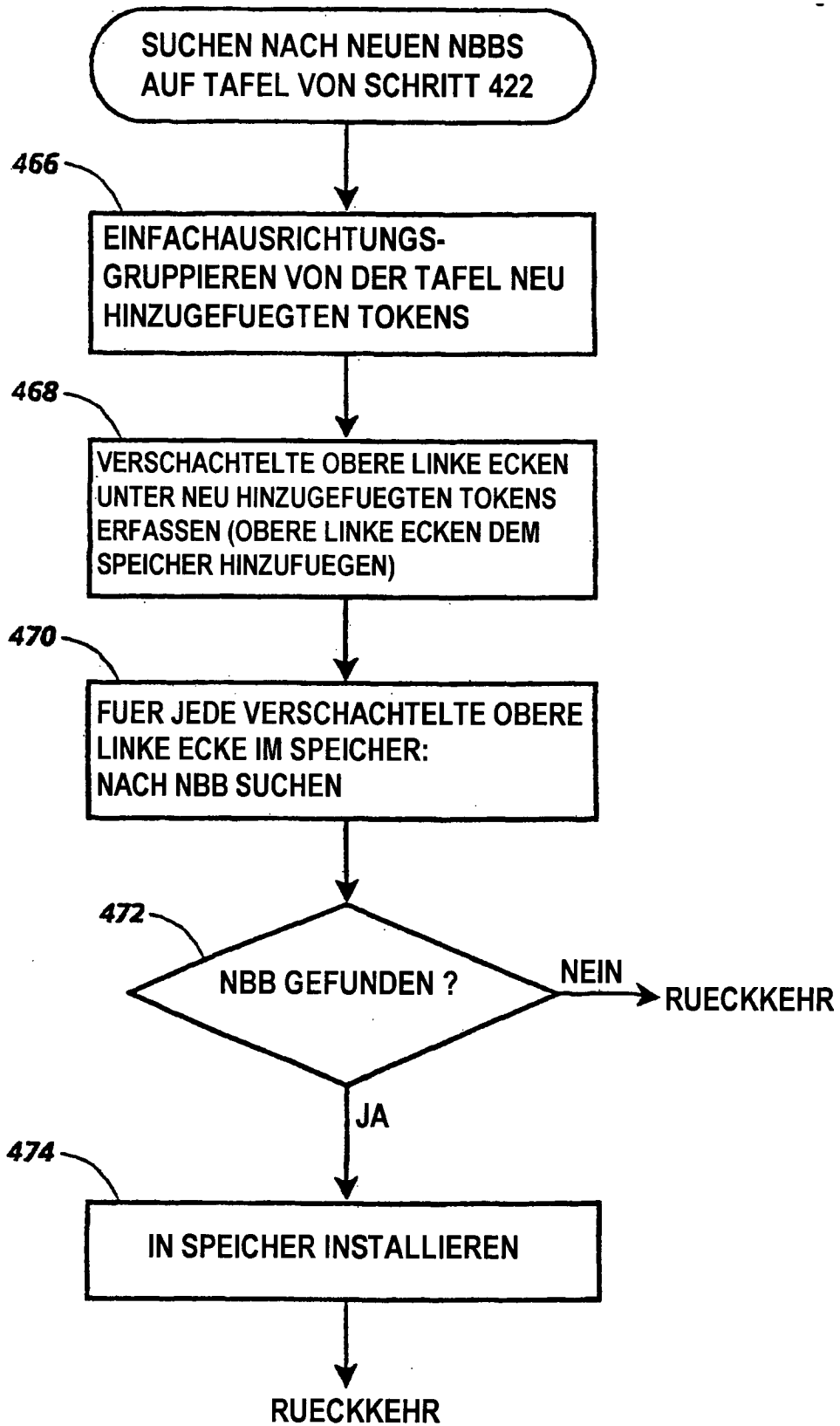
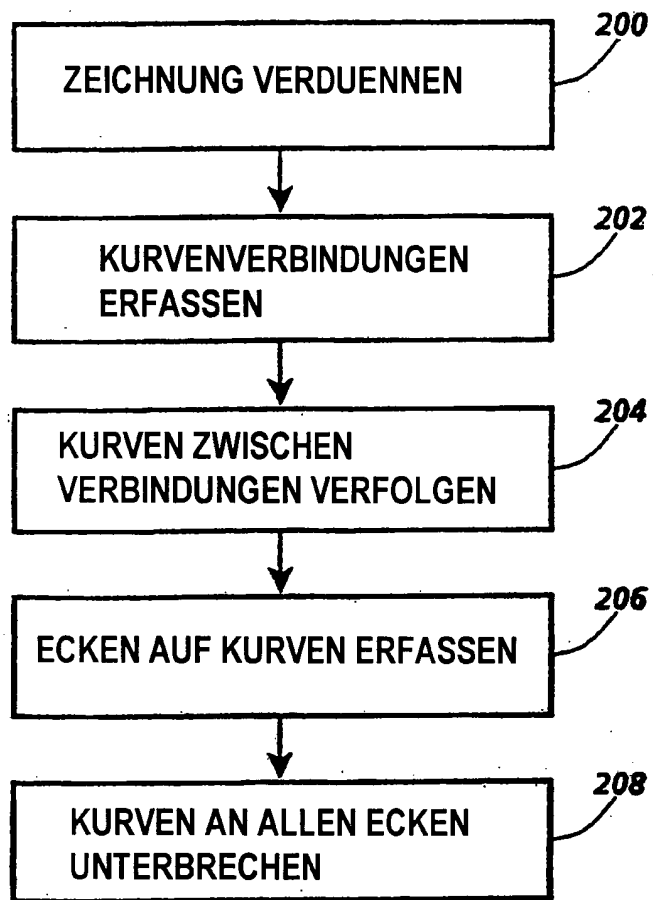
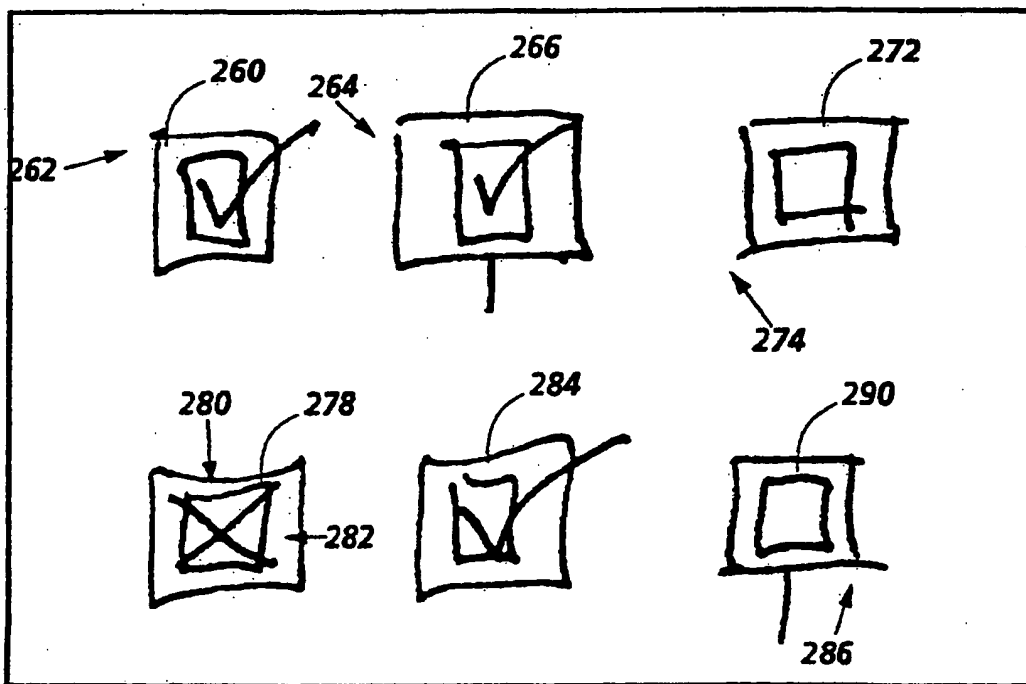


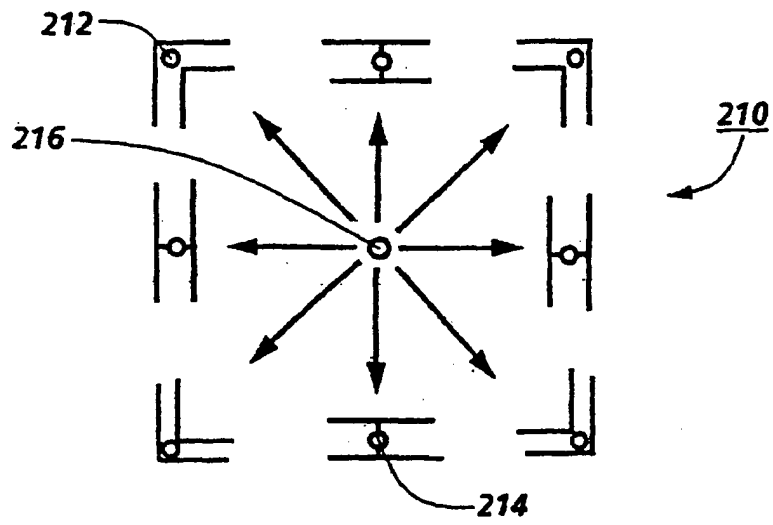
FIG. 14



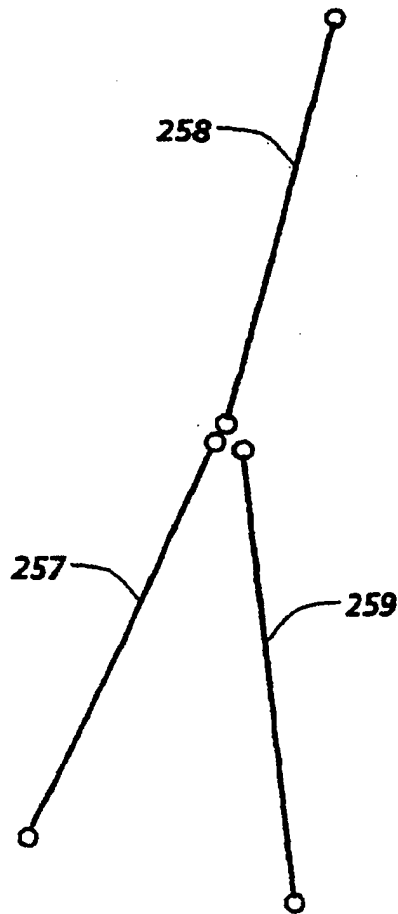
**Fig.15**

**FIG. 16**



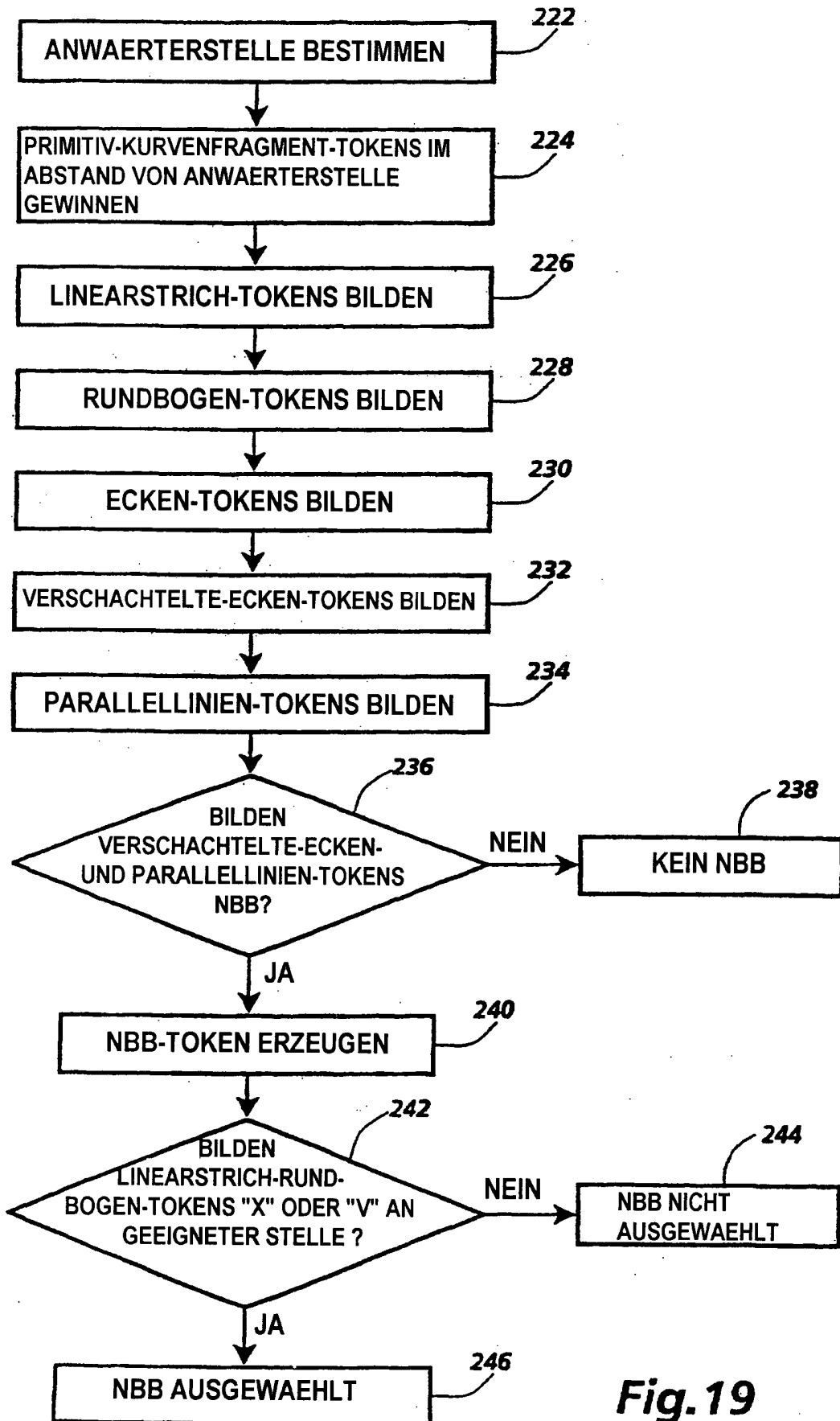


**Fig.17**

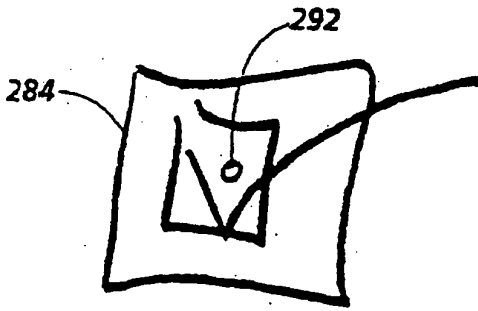


**Fig.18**

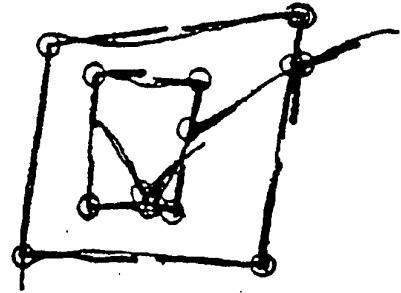




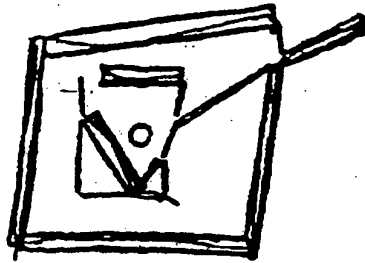
**Fig. 19**



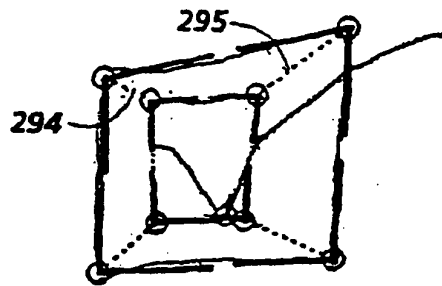
**Fig. 20**



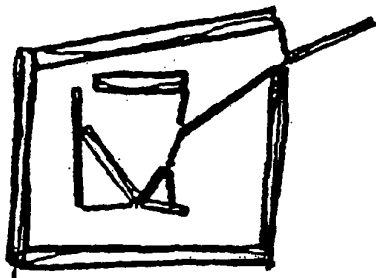
**Fig. 24**



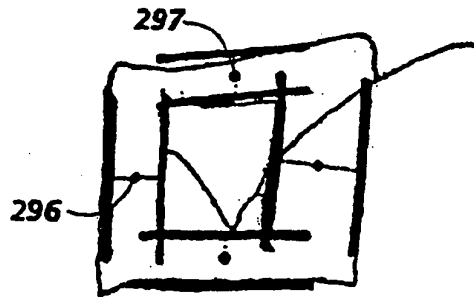
**Fig. 21**



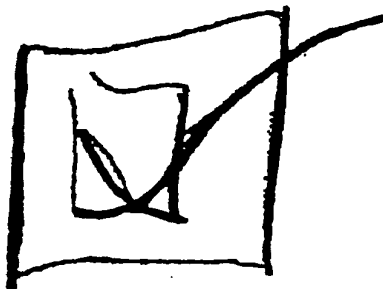
**Fig. 25**



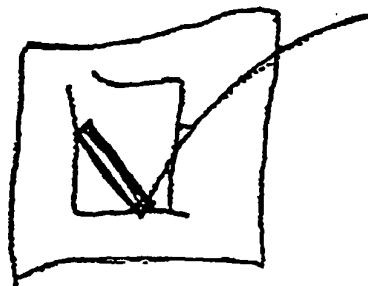
**Fig. 22**



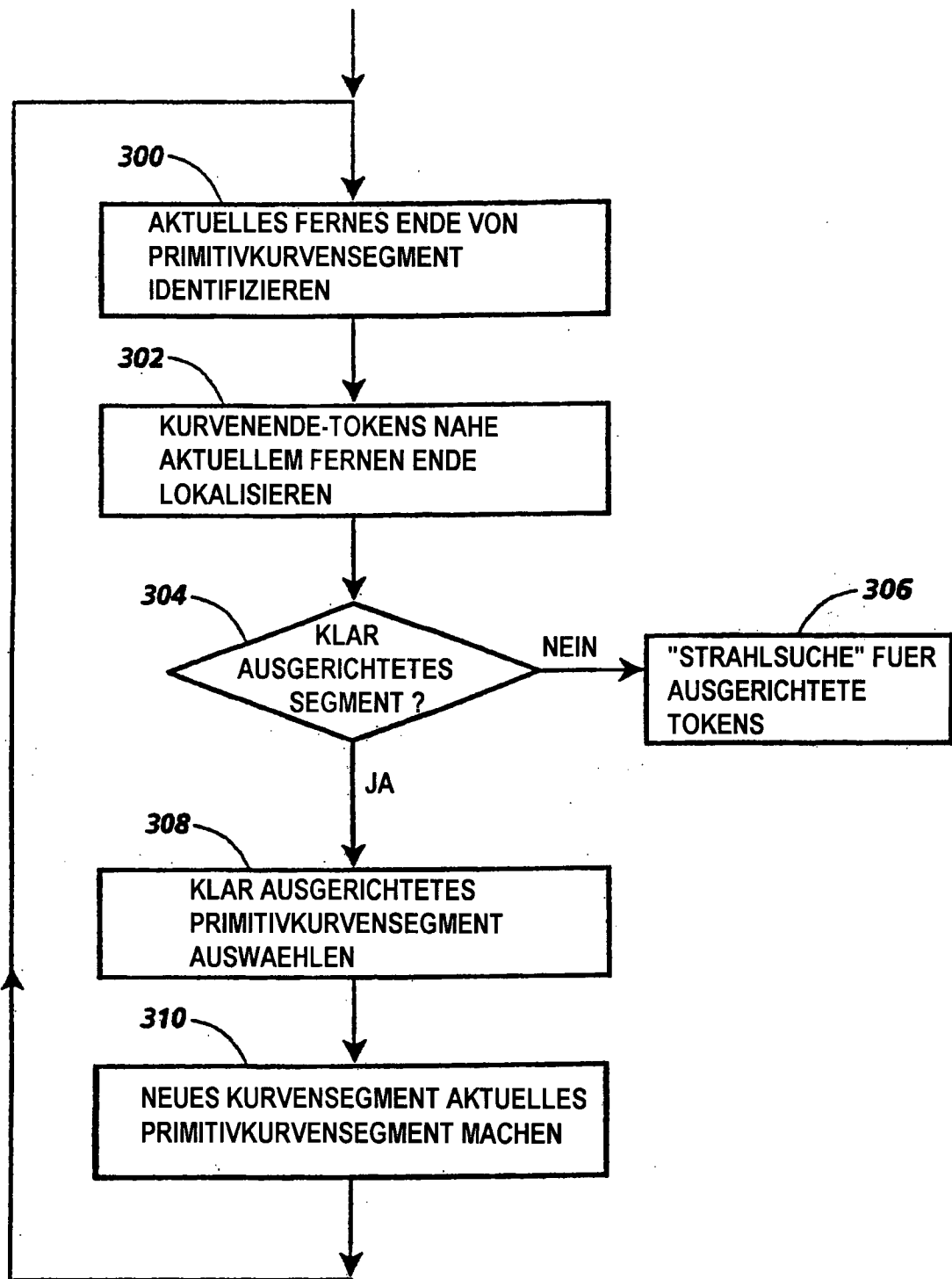
**Fig. 26**



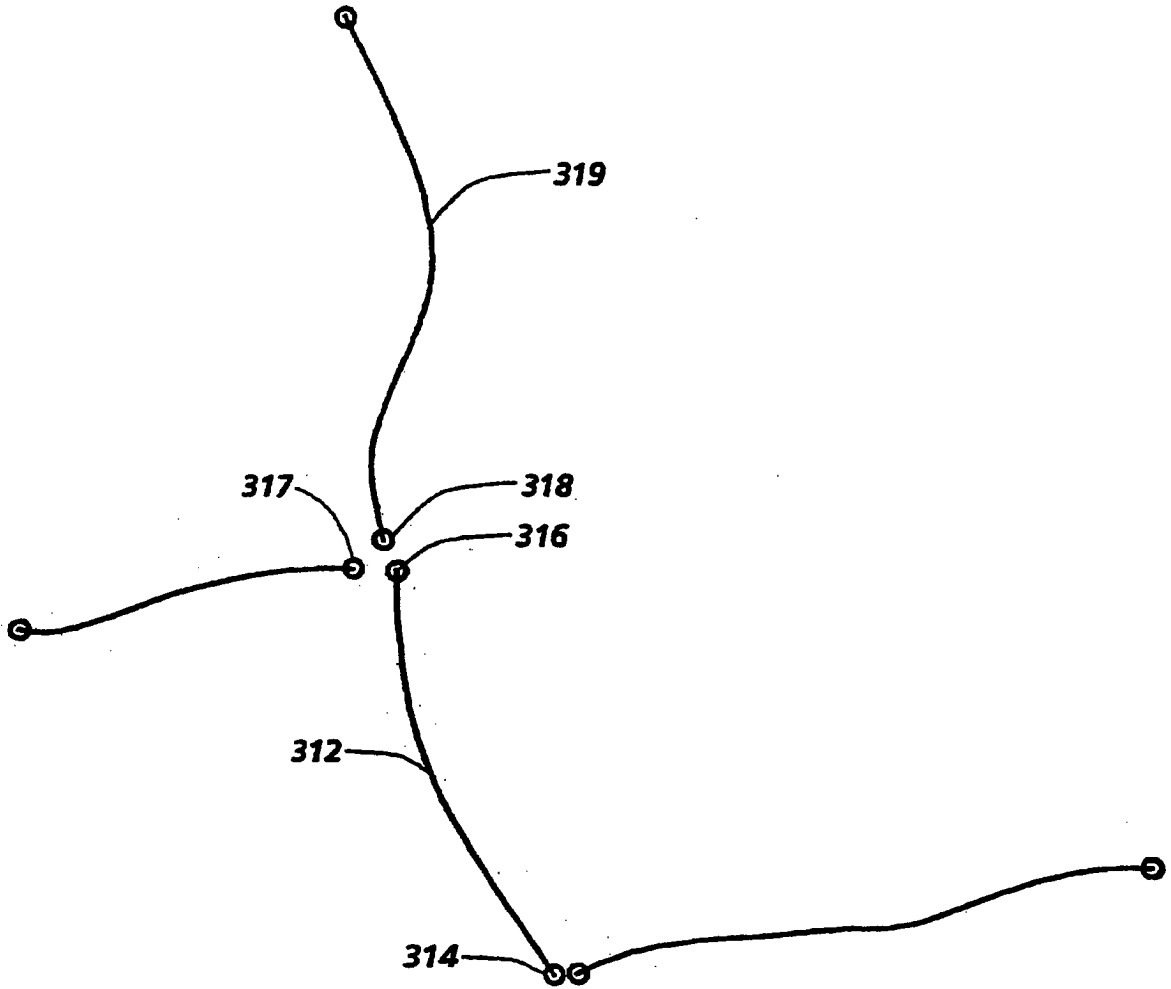
**Fig. 23**



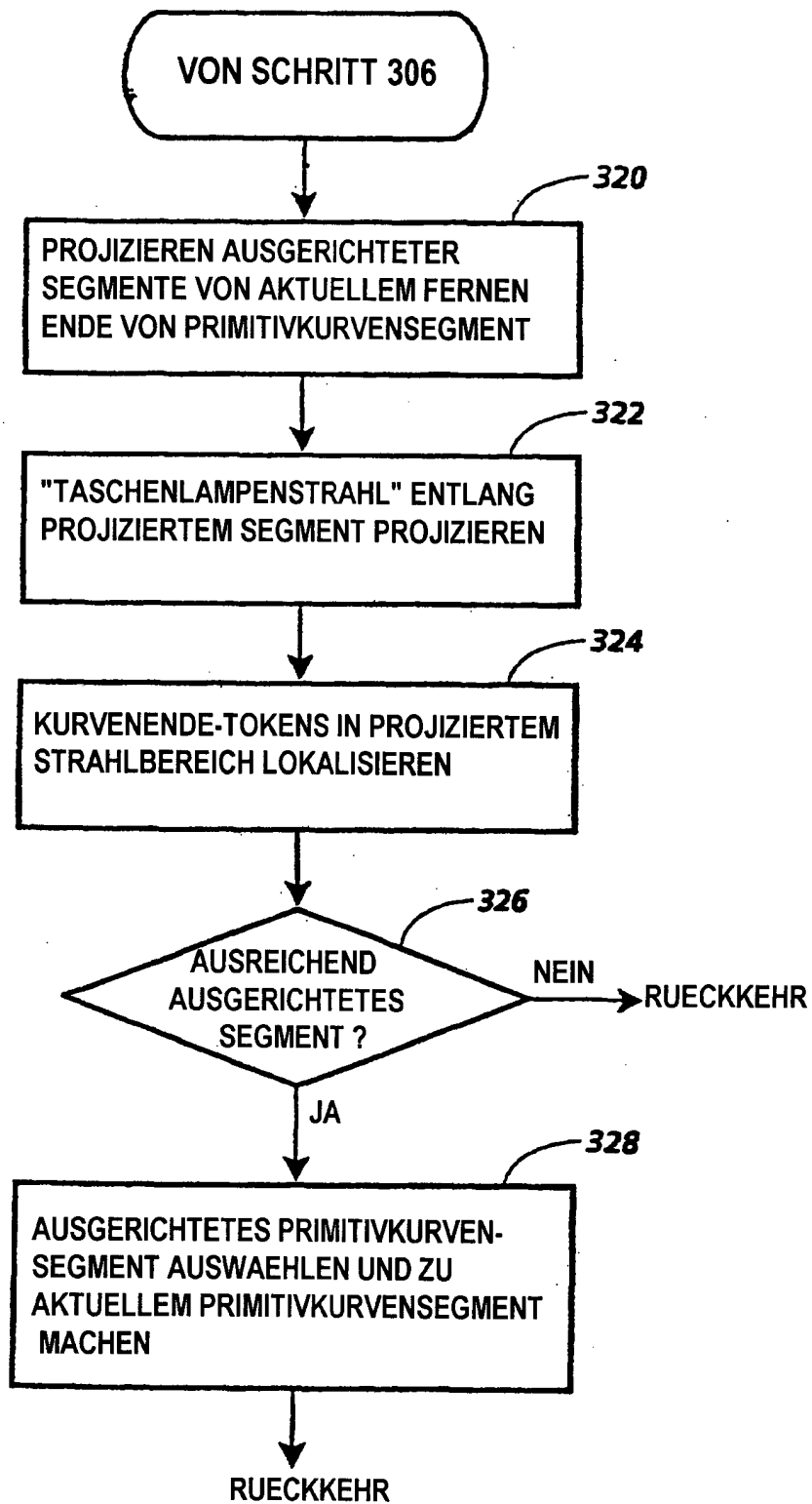
**Fig. 27**



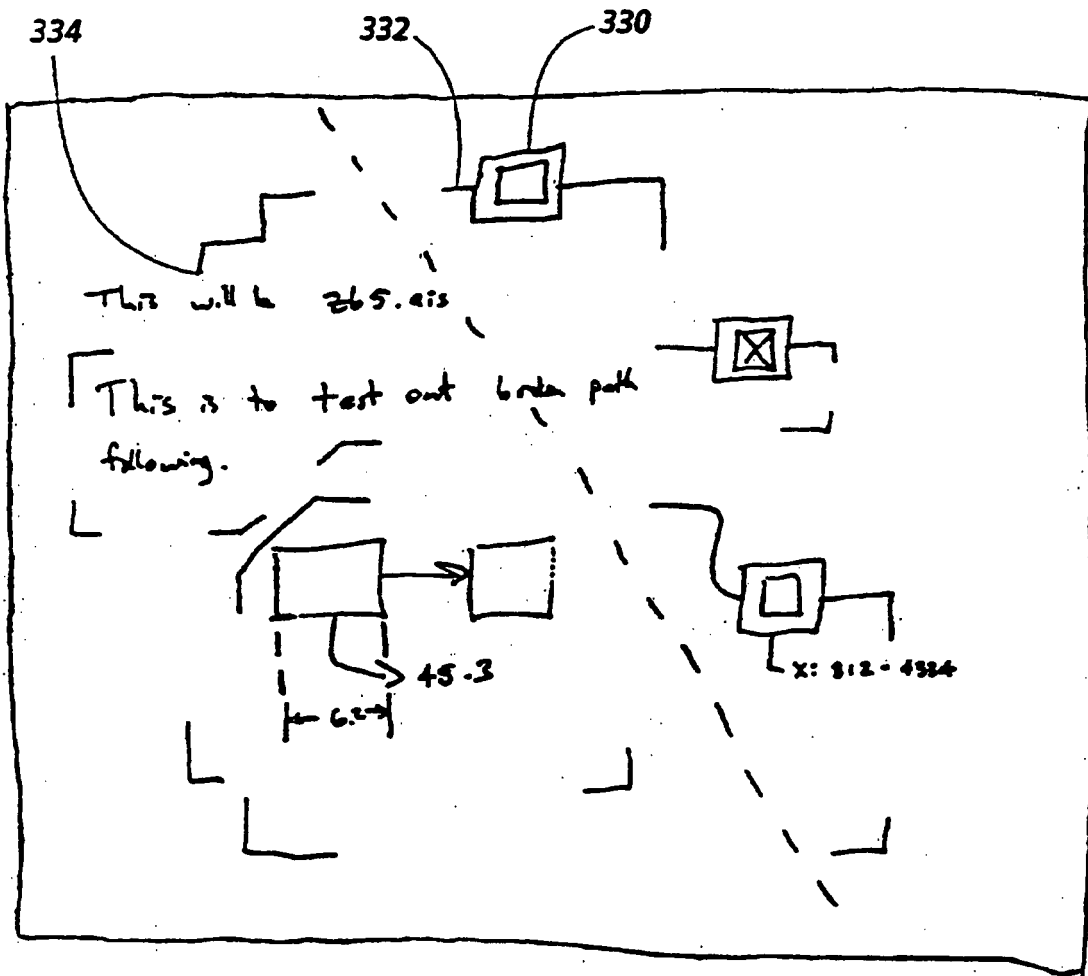
**Fig.28**



**Fig.29**



**Fig.30**



**Fig.31**

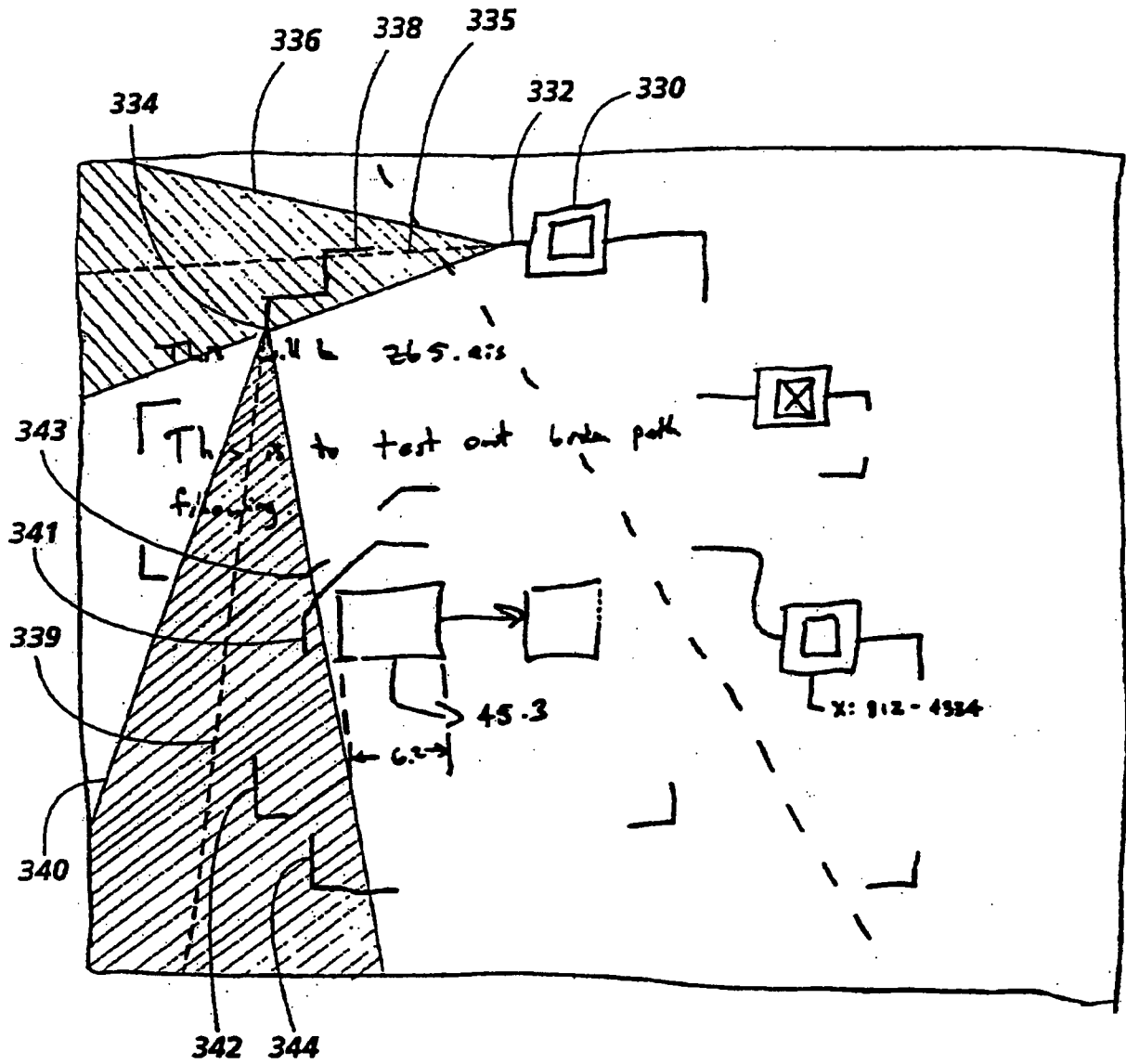
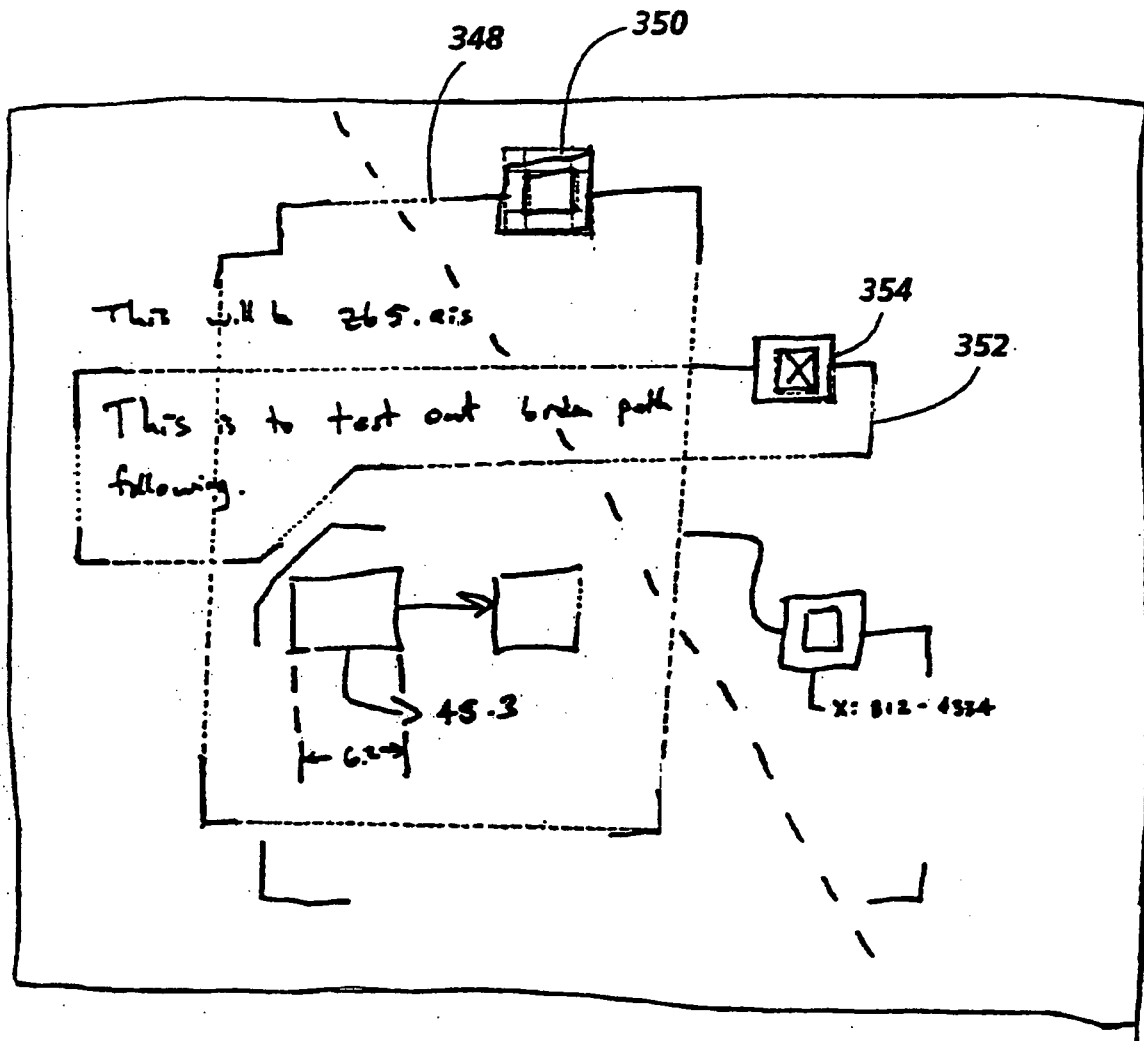


Fig.32



**Fig.33**



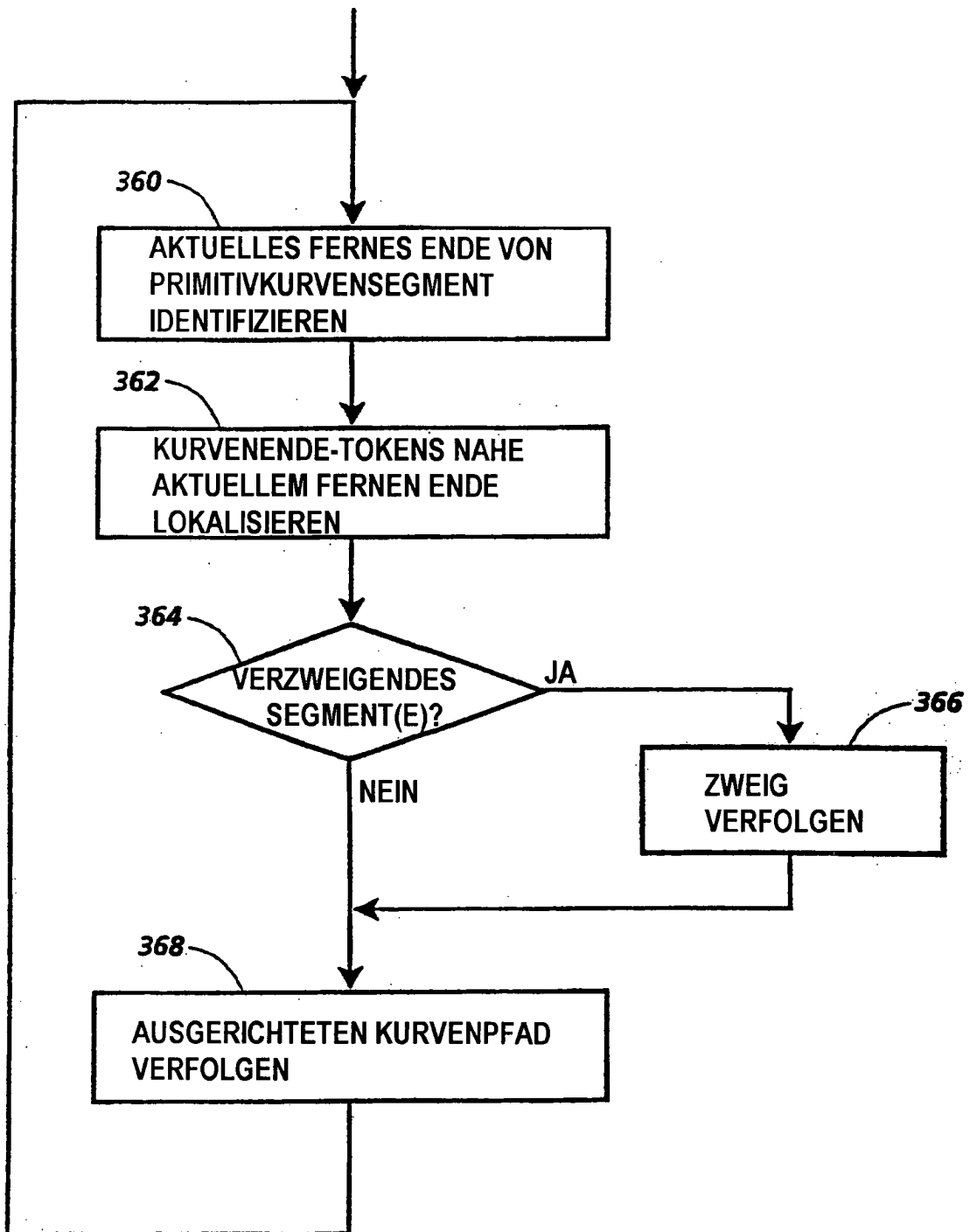
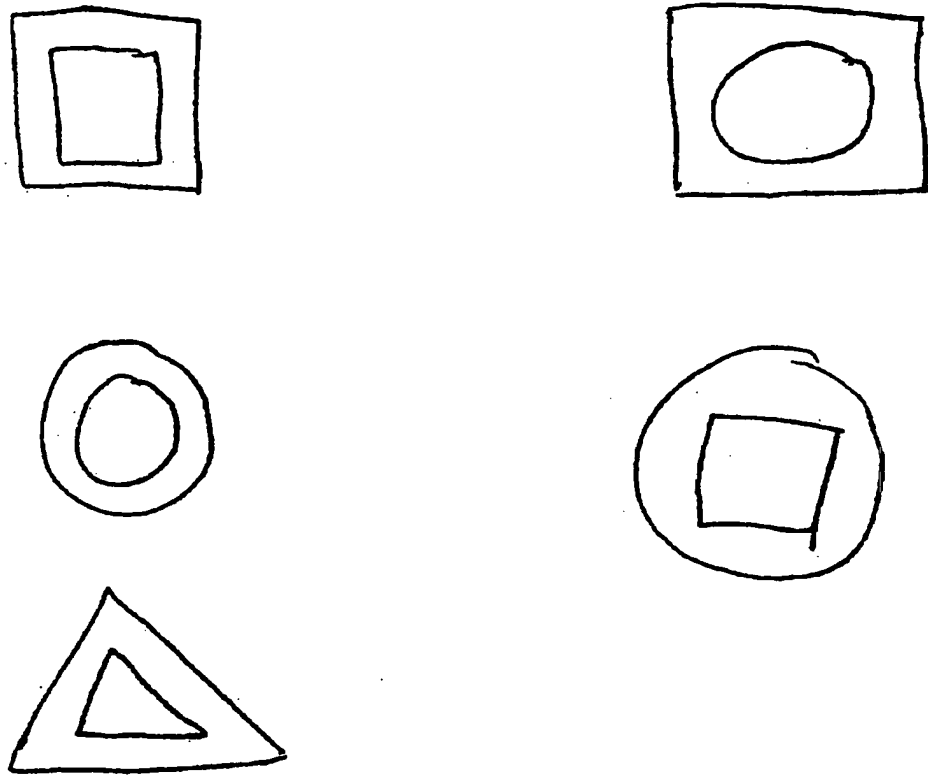


FIG. 34



**Fig.35**