



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 697 31 425 T2 2005.11.10**

(12)

Übersetzung der europäischen Patentschrift

(97) **EP 0 809 213 B1**

(51) Int Cl.7: **G06T 11/60**

(21) Deutsches Aktenzeichen: **697 31 425.1**

(96) Europäisches Aktenzeichen: **97 303 479.6**

(96) Europäischer Anmeldetag: **21.05.1997**

(97) Erstveröffentlichung durch das EPA: **26.11.1997**

(97) Veröffentlichungstag

der Patenterteilung beim EPA: **03.11.2004**

(47) Veröffentlichungstag im Patentblatt: **10.11.2005**

(30) Unionspriorität:

PO002196 22.05.1996 AU

(84) Benannte Vertragsstaaten:

DE, FR, GB

(73) Patentinhaber:

Canon K.K., Tokio/Tokyo, JP

(72) Erfinder:

Politis, George, New South Wales 2113, AU

(74) Vertreter:

TBK-Patent, 80336 München

(54) Bezeichnung: **Verfahren und Apparat zur Optimierung des Expressionsbaumes zur Darstellung von Bildern**

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

Gebiet der Erfindung

[0001] Die vorliegende Erfindung bezieht sich auf die Erzeugung computergenerierter Bilder sowohl in der Form von Stehbildern als auch einer Videoabbildung, und insbesondere auf ein Verfahren, eine Vorrichtung und ein System zur Erzeugung eines durch die Zusammensetzung einer Vielzahl von Komponenten aufgebauten Bildes.

Hintergrund

[0002] Computergenerierte Bilder bestehen typischerweise aus vielen verschiedenen Komponenten oder Graphikelementen, die zur Erzeugung eines endgültigen Bildes gerendert und zusammengesetzt werden. Bisher wurde allgemein ein „Opazitätskanal (der auch als „Matt“- „Alphakanal“ oder einfach als „Opazität“ bekannt ist) verwendet. Der Opazitätskanal beinhaltet Informationen hinsichtlich der Durchlässigkeit jedes Elements. Der Opazitätskanal ist zusammen mit jeder Einheit einer Farbe gespeichert, sodass beispielsweise ein Bildelement basiertes Bild mit Opazität einen Opazitätswert als Teil der Darstellung jedes Bildelementes speichert. Ein Element ohne explizite Opazitätskanalinformationen wird typischerweise als vollständig undurchlässig innerhalb definierter Grenzen des Elements verstanden, und von ihm wird angenommen, dass es außerhalb dieser Grenzen vollständig transparent ist.

[0003] Ein Ausdrucksbaum bietet ein systematisches Mittel oder eine systematische Darstellung zum Rendern von Objekten oder Elementen eines Bildes. Ausdrucksbäume umfassen eine Vielzahl von Knoten einschließlich Blattknoten, inneren Knoten und eines Wurzelknotens. Ein Blattknoten, der einen äußersten Knoten eines Baums darstellt, hat keine Nachkommenknoten und besteht aus einem oder mehreren Graphikelementen. Ein innerer Knoten verzweigt typischerweise in linke und rechte Unterbäume, wobei jeder Unterbaum selbst ein Ausdrucksbaum mit zumindest einem Blattknoten ist. Die inneren Knoten eines Ausdruckbaums sind Zusammensetzungsoperatoren, die den linken und rechten Unterbaum als Operanden des Operators behandeln. Der erste Knoten des Ausdruckbaums wird allgemein als Wurzelknoten bezeichnet. Der Wurzelknoten eines Ausdruckbaums stellt das endgültige Bild dar, und jeder Knoten des Baums stellt einen Teil des endgültigen Bildes dar.

[0004] Obwohl ein Graphikelement selbst eine bestimmte Größe haben kann, muss es im endgültigen Bild nicht unbedingt vollständig sichtbar sein, oder es kann sich lediglich ein Teil des Elements auf das endgültige Bild auswirken. Beispielsweise wird angenommen, dass ein Bild einer bestimmten Größe auf einer Anzeigeeinrichtung anzuzeigen ist. Ist aber das Bild derart positioniert, dass lediglich die obere linke Ecke des Bildes von der Anzeigeeinrichtung angezeigt wird, wird der Rest des Bildes nicht angezeigt. Das endgültige Bild wie auf der Anzeigeeinrichtung angezeigt umfasst somit den sichtbaren Abschnitt des Bildes, und der unsichtbare Abschnitt muss in diesem Fall nicht gerendert werden.

[0005] Ein anderer Fall, in dem lediglich ein Abschnitt eines Elements eine Auswirkung haben kann, ist dann gegeben, wenn der Abschnitt durch ein anderes Element verdeckt wird. Beispielsweise kann ein anzuzeigendes (oder zu Renderndes) endgültiges Bild ein oder mehrere undurchlässige Graphikelemente umfassen, von denen einige andere Graphikelemente verdecken. Daher haben die verdeckten Elemente keine Auswirkung auf das endgültige Bild.

[0006] Kann ein Element, oder Teile eines Elements, die keine Auswirkung auf das endgültige Bild haben, identifiziert werden, müssen diese Elemente (oder Teile) nicht gerendert werden, wodurch erheblich Zeit und möglicherweise Speicherplatz eingespart wird.

[0007] Die am 31. Januar 1996 veröffentlichte Europäische Patentanmeldung mit der Veröffentlichungsnummer EP 0 694 881 A2 (Europäische Patentanmeldung Nr. 95305144.8) von Canon Kabushiki Kaisha und Canon Information Systems Research Australia Pty Ltd) beschreibt ein System, ein Verfahren und eine Sprache zur Zusammensetzung oder Erzeugung von Bildern. Die Bilder umfassen typischerweise eine Vielzahl von Graphikelementen, die jeweils Farb- und Opazitätsinformationen enthalten. Das System verwendet Operatoren, die die Graphikelemente aufweisen, und Operanden, wobei die Operatoren die Operanden entsprechend einer durch die Operatoren definierten Funktion, den Farbinformationen und Opazitätsinformationen zur Erzeugung neuer Graphikelemente kombinieren. Ein Teil des Systems beinhaltet die Interpretation der Parsingsprache und die Ausführung einer Folge von Aussagen und die Bildung eines Ausdrucksbaums, dessen Knoten ein Graphikelement umfassen. Anweisungen werden dann anhand des Baums erhalten. Ein anderer Teil

erlaubt die Zusammensetzung undurchlässiger Graphikelemente und zugehöriger Abschneidoperationen. Bounding-Box-Verfahren werden zum Lokalisieren aktiver Bereiche der Graphikelemente aus den Knoten verwendet.

[0008] Die EP 0 694 881 A2 offenbart ein Verfahren und eine Vorrichtung zum Optimieren eines Ausdrucksbaums. Der Ausdrucksbaum gemäß der EP 0 694 881 A2 dient der Zusammensetzung eines Bildes und umfasst zumindest drei Knoten. Jeder Knoten des Baums ist entweder ein Graphikelement oder ein Graphikoperator und weist eine durch den Knoten dargestellte Region des Bildes auf. Das Verfahren der EP 0 694 881 A2 umfasst für zumindest einen Graphikoperatorknoten im Baum die Schritte des Vergleichs einer ersten Region, die aus Operanden des Operatorknotens hergeleitet wird, mit einer Region, die von einem anderen Operanden des Operatorknotens hergeleitet wird, die Bestimmung, ob die erste Region die zweite Region total oder teilweise verdeckt, und die Modifikation des Ausdrucksbaums, wenn die erste Region die zweite Region ganz oder teilweise verdeckt.

[0009] Aus dem Verfahren des Standes der Technik ergeben sich zumindest für Bilder Probleme, in denen eine Überlappung auftritt, weil diese Verfahren mit transparenten Graphikobjekten nicht leicht fertig werden, und auch nicht den vollständigen Bereich von Zusammensetzungsoperatoren behandeln. Daher soll zumindest eines dieser Probleme vermindert werden.

Kurzzusammenfassung

[0010] Gemäß einer Ausgestaltung der Erfindung ist ein Verfahren zur Optimierung eines Ausdrucksbaums ausgebildet, der zum Zusammensetzen eines Bildes dient und zumindest drei Knoten umfasst, wobei jeder Knoten des Baums zumindest entweder ein eine Bildregion definierendes Graphikelement oder ein Graphikoperator ist, wobei das Verfahren für zumindest einen Knoten im Baum die Schritte umfasst
Vergleichen einer ersten Region des Knotens mit einer zweiten Region, die von zumindest einem anderen Knoten irgendwo im Ausdrucksbaum erhalten wird,
Bestimmen, ob die erste Region durch die zweite Region ganz oder teilweise verborgen ist, und
Modifizieren des Ausdrucksbaums, wenn die erste Region zumindest teilweise oder ganz durch die zweite Region verborgen ist, um einen optimierten Ausdrucksbaum zum Zusammensetzen des gleichen Bildes auszubilden, wobei ein optimierter Teil des Ausdrucksbaums im wesentlichen nicht-verborgene Abschnitte der ersten Region darstellt.

[0011] Gemäß einer anderen Ausgestaltung der Erfindung ist eine Vorrichtung zur Optimierung eines Ausdrucksbaums ausgestaltet, der zum Zusammensetzen eines Bildes dient und zumindest drei Knoten umfasst, wobei jeder Knoten des Baums zumindest entweder ein eine Bildregion definierendes Graphikelement oder ein Graphikoperator ist, mit
einer Einrichtung zum Vergleichen einer ersten Region des Knotens mit einer zweiten Region, die von zumindest einem anderen Knoten irgendwo im Ausdrucksbaums erhalten wird,
einer Einrichtung zum Bestimmen, ob die erste Region durch die zweite Region ganz oder teilweise verborgen ist, und
einer Einrichtung zum Modifizieren des Ausdrucksbaums, wenn die erste Region zumindest teilweise oder ganz durch die zweite Region verborgen ist, um den optimierten Ausdrucksbaum zum Zusammensetzen des gleichen Bildes auszubilden, wobei ein optimierter Teil des Ausdrucksbaums im wesentlichen nicht-verborgene Abschnitte der ersten Region darstellt.

Kurzbeschreibung der Zeichnung

[0012] Nachstehend wird ein bevorzugtes Ausführungsbeispiel der Erfindung unter Bezugnahme auf die beiliegende Zeichnung und den Anhang näher beschrieben. Es zeigen:

[0013] [Fig. 1](#) eine schematische Darstellung verschiedener Zusammensetzungsoperationen,

[0014] [Fig. 2A](#) bis [Fig. 2D](#) ein Beispiel der Anwendung eines Abschneideoperators gemäß einem Ausführungsbeispiel der Erfindung,

[0015] [Fig. 3](#) ein Beispiel eines aus einfachen Graphikobjekten bestehenden gerenderten Bildes,

[0016] [Fig. 4](#) einen Bildausdrucksbaum, der die Zusammensetzung der einfachen Graphikobjekte zum Zusammensetzen oder Rendern des Bildes in [Fig. 3](#) darstellt,

- [0017] [Fig. 5](#) einen vereinfachten Bildausdrucksbaum des Bildausdrucksbaums in [Fig. 4](#) gemäß einem Ausführungsbeispiel der Erfindung,
- [0018] [Fig. 6](#) ein weiteres Beispiel eines aus einfachen Graphikobjekten bestehenden gerenderten Bildes,
- [0019] [Fig. 7](#) einen Bildausdrucksbaum, der die Zusammensetzung der Graphikobjekte zum Zusammensetzen oder Rendern des Bildes in [Fig. 6](#) darstellt,
- [0020] [Fig. 8](#) einen vereinfachten Ausdrucksbaum zum Zusammensetzen des Bildes in [Fig. 6](#) gemäß dem Ausführungsbeispiel der Erfindung,
- [0021] [Fig. 9](#) ein abstraktes Ablaufdiagramm, das einen Überblick über den erfindungsgemäßen Vorgang der Optimierung eines Ausdrucksbaums liefert, der zum Zusammensetzen eines Bildes verwendet wird,
- [0022] [Fig. 10](#) ein ausführliches Ablaufdiagramm der Modifikation des Ausdrucksbaums gemäß Schritt 912 in [Fig. 9](#),
- [0023] [Fig. 11](#) ein weiteres ausführliches Ablaufdiagramm zur Optimierung des Ausdrucksbaums, und
- [0024] [Fig. 12](#) ein Blockschaltbild eines herkömmlichen Universalcomputers, der zum Implementieren der Ausführungsbeispiele der Erfindung verwendet werden kann.
- [0025] Der Anhang umfasst Pseudo-Code-Routinen, die für eine Computerimplementierung des bevorzugten Ausführungsbeispiels geeignet sind.

Ausführliche Beschreibung

- [0026] In der folgenden Beschreibung des bevorzugten Ausführungsbeispiels wird angenommen, dass ein Bildzusammensetzungsausdrucksbaum, wie er hier beschrieben wird, für ein zu renderndes Bild bestimmt wurde.
- [0027] Vorzugsweise sind Bildregionsdarstellungen hierarchische Datenstrukturen, die zur Darstellung einer Region oder eines Abschnitts eines Bildes geeignet sind, und typischerweise bei der Bildverarbeitung verwendet werden. Eine derartige Bildregionsdarstellung ist dem Fachmann als „Quadtree“ bekannt. Andere Formen der Bildregionsdarstellungen können auch zweckmäßig sein. Der Einfachheit halber wird eine Bildregionsdarstellung nachstehend als Quadtree bezeichnet.
- [0028] Typischerweise erfordert die Erzeugung eines Quadtree, der eine Region eines Bildes darstellt, die Unterteilung der Region in eine Vielzahl von Zellen, wobei jede Zelle ein Teil der Region ist, und jede Zelle durch einen Knoten des Quadtree dargestellt wird. Daher erhöht die Erhöhung der Anzahl von Unterteilungen einer Region eines Bildes entsprechend die Anzahl von Knoten des Quadtree, wodurch sich die Tiefe des Quadtree und die Auflösung der durch den Quadtree dargestellten Region erhöht.

ZUSAMMENSETZUNGSOPERATIONEN

- [0029] Zusammensetzungsoperationen enthalten 13 Hauptzusammensetzungsoperationen zum Zusammensetzen von zwei Abschnitten eines einzelnen Bildes. Die Funktion jeder dieser Zusammensetzungsoperationen ist in Tabelle 1 aufgeführt, wobei D_c ein vormultipliziertes Ziel oder eine resultierende Farbe ist, D_o ein Ziel oder ein resultierender Alpha- (α) Kanalwert ist, A_c eine vormultiplizierte Elementfarbe eines ersten Abschnitts einer ersten Quelle A ist, A_o ein α -Wert ist, der dem Bildelement mit der Farbe A_c entspricht, B_c ein vormultiplizierter Bildelementfarbwert eines Abschnitts eines Bildes einer zweiten Quelle B ist, und B_o der α -Kanalwert des Bildelements ist, das B_c der Quelle B entspricht.

Tabelle 1: Zusammensetzungsoperationen

OPERATION	GLEICHUNG
LEER	$D_c = 0$ $D_o = 0$
A	$D_c = A_c$ $D_o = A_o$
B	$D_c = B_c$ $D_o = B_o$
A over B	$D_c = A_c + B_c(1-A_o)$ $D_o = A_o + B_o(1-A_o)$
A rover B	$D_c = A_c(1-B_o) + B_c$ (Umgekehrter Fall von A over B) $D_o = A_o(1-B_o) + B_o$
A in B	$D_c = A_c B_o$ $D_o = A_o B_o$
A rin B	$D_c = A_o B_c$ (Umgekehrter Fall von A in B) $D_o = A_o B_c$
A out B	$D_c = A_c(1-B_o)$ $D_o = A_o(1-B_o)$
A rout B	$D_c = B_c(1-A_o)$ (Umgekehrter Fall von A out B) $D_o = B_o(1-A_o)$
A atop B	$D_c = A_c B_o + B_c(1-A_o)$ $D_o = A_o B_o + B_o(1-A_o)$
A ratop B	$D_c = A_c(1-B_o) + B_c A_o$ $D_o = A_o(1-B_o) + B_o A_o$
A Xor B	$D_c = A_c(1-B_o) + B_c(1-A_o)$ $D_o = A_o(1-B_o) + B_o(1-A_o)$
A plusW B	$D_c = A_c + B_c$ (mit D_c "wrap around") $D_o = A_o + B_o$ (mit D_o "wrap around")
A plusC B	$D_c = A_c + B_c$ (mit D_c "clamped") $D_o = A_o + B_o$ (mit D_o "clamped")

[0030] Tabelle 1 zeigt insbesondere verschiedene Zusammensetzungsverfahren zum Zusammensetzen von zwei verschiedenen Bildern unter Verwendung verschiedener Operatoren. Es sind auch zusätzliche Operatoren möglich. Die zusätzlichen Operatoren können zum Implementieren von Spezialeffekten verwendet werden.

[0031] Die „wrap around“-Eigenschaft des „plusW“-Operators bedeutet, dass, wenn beispielsweise die Addition von $A_c + B_c$ größer als ein maximaler Wert einer Farbkomponente ist, der Wert „herumgewickelt“ wird, um erneut unter Bezugnahme auf den minimalen Wert im Farbraum zu beginnen. Alternativ dazu beinhaltet der Vorgang „clamping“, der von „plusC“ verwendet wird, eine abschneidende bzw. begrenzende Addition beispielsweise von $A_c + B_c$ auf einen maximalen Wert einer Farbkomponente, wenn die Addition größer als diese Komponente ist.

[0032] Fig. 1 zeigt verschiedene Beispiele des endgültigen Bildes, das erzeugt wird, wenn verschiedene Operationen (wie sie in Tabelle 1 aufgeführt sind) bei der Zusammensetzung von zwei vollständig undurchlässigen Kreisen A und B verwendet werden. Die Operatoren „rover“, „rin“, „rout“ und „ratop“ sind dem Vertauschen der Operanden gemäß dem „r“ (Umkehr-)Operator und Anwenden des entsprechenden Operators „over“ („über“), „in“, „out“ („außerhalb“) und „atop“ („zu Oberst“) jeweils äquivalent.

[0033] Bei dem bevorzugten Ausführungsbeispiel kann ein Ausdrucksbaum eine Vielfalt von Knotentypen enthalten, die binäre Zusammensetzungsoperatoren, unäre Operatoren und Primitive beinhalten. Unäre Operatoren enthalten typischerweise Farbtransformationen, Bildfaltungen, affine Transformationen und eine Bildverbindung. Primitive beinhalten typischerweise Graphikelemente, wie Bildelement-basierte Bilder, splinebasierte Wege, Text, „all“ („all“ ist ein Graphikelement, das die Größe des gesamten erzeugten Bildes überspannt), Kantenmischungen, Boxen oder dergleichen.

BINÄRE ZUSAMMENSETZUNGSOPERATOREN

[0034] Tabelle 2 listet einen Satz binärer Zusammensetzungsoperatoren und die durchzuführende Aktion auf, wenn diese Operatoren bei der Vereinfachung eines Ausdrucksbaums behandelt werden.

TABELLE 2

Operator	linkem Operanden übergeben	rechtem Operanden übergeben	Rückgabe	wenn linker Operand verschwindet	wenn rechter Operand verschwindet
over	q0	q0 ∪ qL	qL ∪ qR	R	L
in	q0	q0	qL ∩ qR	V	V
out (bei rechtem Operanden zuerst anwenden)	q0 qR	q0	qL - B (rechts)	V	L
Out (bei linkem Operanden zuerst anwenden)	q0	q0	qL- B (rechts)	V	L
plusC	q0	q0	qL ∪ qR	R	L
plusW, Xor	q0	q0	(qL- B (rechts)) ∪ (qR- B (links))	R	L

[0035] Bei einem Knoten eines Ausdrucksbaums, der durch einen Operator dargestellt wird, wird typischerweise während des Vorgangs der Vereinfachung des Ausdrucksbaums jedem Operanden eine Regionsdarstellung, wie ein Quadtree, übergeben. An dem den Operator vergleichenden Knoten muss ein Vorgang dahingehend ausgeführt werden, ob ein vom Knoten verzweigender Unterbaum verschwindet (das heißt, Zweige müssen nicht beschnitten werden), oder ein den verdeckten Abschnitten der Graphikelemente entsprechender Quadtree von diesem Knoten für eine mögliche weitere Verarbeitung an andere Knoten zurückzugeben ist.

[0036] In Tabelle 2 wird folgende Notation verwendet:
 q0: der dem Knoten übergebene Quadtree,

q_L, q_R : der von dem linken und rechten Unterbaum entsprechend dem linken und rechten Operanden eines Operators aus Tabelle 2 zurückgegebene Quadtree,

$$\left. \begin{array}{l} q_1 \cap q_2 \\ q_1 \cup q_2 \\ q_1 - q_2 \end{array} \right\} :$$

eingestellte Quadtree-Operation und

$B(\text{Knoten})$: ein Quadtree, der vollständig die Bounding-Box des Knotens enthält.

[0037] In den letzten zwei Spalten in Tabelle 2 sind typische Ersetzungsregeln spezifiziert, wobei „L“ bedeutet, dass ein aktueller Knoten durch den linken Unterbaum zu ersetzen ist, der vom aktuellen Knoten verzweigt, „R“ bedeutet, dass ein aktueller Knoten durch den rechten Unterbaum zu ersetzen ist, der vom aktuellen Knoten verzweigt, und „V“ bedeutet, dass der aktuelle Knoten verschwindet. Ein Knoten, der „verschwindet“, lässt darauf schließen, dass die durch den Knoten dargestellte Region des Bildes durch andere Graphikelemente verborgen wird. Der Knoten hat daher keine Auswirkung auf das endgültige Bild. Verschwinden beide Operanden, verschwindet auch der aktuelle Knoten.

[0038] Für die in Tabelle 2 beschriebenen Operatoren können Umkehroperatoren eingesetzt werden. Der als „A over B“ beschriebene „Over“-Operator gibt an, dass das Graphikelement „A“ über dem Graphikelement „B“ liegt. Dies kann durch einen Umkehroperator des „Over“-Operators ersetzt werden, der typischerweise mit „rover“ (umgekehrtes über) bezeichnet wird, sodass „B rover A“ in einer Zusammensetzung der Graphikelemente „A“ und „B“ äquivalent zu „A over B“ resultiert.

BEISPIELE BINÄRER ZUSAMMENSETZUNGSOPERATOREN

[0039] Als Beispiel wird der (erste) Operator in der ersten Reihe der „Operator“-Spalte in Tabelle 2 (das heißt, der „Over“-Operator) betrachtet. An einem aktuellen Knoten eines Ausdrucksbaums, der durch einen „Over“-Operator dargestellt wird, übergibt ein Elternknoten dem aktuellen Knoten einen Quadtree q_0 . Den Vorgang unter der Überschrift „dem linken Operanden übergeben“ (Spalte 2 in Tabelle 2) folgend wird der Quadtree q_0 dem linken Operanden übergeben, der der linke Unterbaum oder Zweig am aktuellen Knoten ist.

[0040] Der Quadtree q_0 wird zur Verarbeitung des linken Operanden verwendet, und ein Quadtree q_L wird als verbergender Bereich des linken Operanden zurückgegeben. Wegen „dem rechten Operanden übergeben“ (Spalte 3 in Tabelle 2) ist die am aktuellen Knoten auszuführende Aktion die Übergabe einer Vereinigung des Elternknotens, des Quadtree q_0 , und des nun zurückgegebenen Quadtree q_L des linken Operanden als rechten Operanden. Der sich aus dieser Vereinigung ($q_0 \cup q_L$) ergebene Quadtree wird zur Verarbeitung des rechten Operanden verwendet. Ein Quadtree q_R wird dem aktuellen Knoten als verbergender Bereich des rechten Operanden zurückgegeben. Der aktuelle Knoten gibt dann die Vereinigung ($q_L \cup q_R$) des linken Operanden q_L und des rechten Operanden q_R dem Elternknoten zurück (4 „Rückgabe“ in Spalte 4 der Tabelle 2).

[0041] Geht hervor, dass die durch den linken Operanden dargestellte Region vom Quadtree q_0 vollständig verborgen ist, der dem linken Operanden übergeben wird, besteht der Vorgang „wenn linker Operand verschwindet“ in Spalte 5 in Tabelle 2 in dem Ersetzen des aktuellen Knotens durch den rechten („R“) Unterbaum beziehungsweise den rechten Operanden. Dies ist erwünscht, da die Änderung des Baums durch Ersetzen des aktuellen Knotens durch seinen rechten Operanden das gerenderte Bild nicht ändert, jedoch die zum Rendern des Bildes erforderliche Zeit verbessert. Ergibt sich, dass die durch den rechten Operanden dargestellte Region vollständig vom Quadtree ($q_0 \cup q_L$) verborgen ist, der dem rechten Operanden übergeben wird, besteht der Vorgang „wenn rechter Operand verschwindet“ in Spalte 6 in Tabelle 2 im Ersetzen des aktuellen Knotens durch den linken („L“) Unterbaum.

UNÄRE OPERATOREN

[0042] Die Behandlung unärer Operatoren bei der Vereinfachung des Ausdrucksbaums hängt vom Typ der Operation ab:

- (a) Bei einer Farbtransformation wird der Quadtree q_0 dem Operanden des Farbtransformationsoperators übergeben. Bewahrt die Transformation die Undurchsichtigkeit (das heißt, undurchsichtige Bildelemente bleiben nach der Transformation undurchsichtig), wird der vom Operanden zurückgegebene Quadtree vom unären Operator zurückgegeben. Das heißt, der Operand verbirgt das, was das Ergebnis der Farbtransfor-

mation verbirgt. Bewahrt die Transformation die Undurchsichtigkeit nicht, gibt der unäre Operator einen leeren Quadtree zurück, da die Region, die die unäre Operation verbirgt, nicht bestimmt werden kann. Verschwindet der Operand, verschwindet der unäre Operator, wenn die Unsichtbarkeit (Null-Opazität) nicht bewahrt wird. Wird die Unsichtbarkeit nicht bewahrt, wird der Unterbaum mit dem unären Operator als Wurzel durch ein geeignetes „All“-Graphikelement ersetzt.

(b) Affine Transformationen und Verbindungen bewahren die Geometrie zwischen dem Quadtree und den Primitiven nicht. Wird der unäre Operator durch den Quadtree q_0 verborgen, verschwindet er. Ansonsten wird der Durchlauf am Operanden der affinen Transformation oder des Bildverbindungsoperators neu gestartet, wobei ein leerer Quadtree als verbergende Region durchlaufen wird. Ein leerer Quadtree wird zum Operator zurückgegeben, wenn der durch einen Operanden zurückgegebene Quadtree nicht leicht transformiert werden kann.

(c) Bildfaltung: Wird der unäre Operator durch den Quadtree q_0 verborgen, verschwindet er. Ansonsten wird der Durchlauf am Operanden der Bildfaltung neu gestartet, wobei ein leerer Quadtree als verbergender Bereich durchlaufen wird. Ein leerer Quadtree wird durch einen solchen Operator zurückgegeben, da das vom Operator bewirkte Verwischen die Verwendung eines durch seinen Operanden zurückgegebenen Quadtree schwierig macht. Ändert der Bildfaltungsoperator allerdings die Opazität nicht, kann der vom Operanden des Operators zurückgegebene Quadtree wiederum vom Operator der Bildfaltung zu seinem Elternknoten zurückgegeben werden.

OPTIMIERUNG DES AUSDRUCKSBAUMS

[0043] Bei dem bevorzugten Ausführungsbeispiel wird ein Bildzusammensetzungsausdrucksbaum (der nachstehend als „Ausdrucksbaum“ bezeichnet wird) eines zu rendernden Bildes durchlaufen, vorzugsweise zuerst in einer Tiefenrichtung. Jeder Knoten des Ausdrucksbaums erhält von seinem Elternknoten eine Regionsdarstellung eines oder mehrerer Bereiche jedes Bildes. Die Regionsdarstellung wird mit der am Knoten dargestellten Region zur Bestimmung verglichen, ob die von diesem Knoten dargestellte Region verborgen ist.

[0044] Ein Knoten, bei dem die durch den Knoten dargestellte Region vollständig verborgen ist, wird aus dem Ausdrucksbaum mit einer geeigneten Vereinfachung des Ausdrucksbaums wie nachstehend beschrieben entfernt. Ist die durch den Knoten dargestellte Region lediglich teilweise verborgen, wird ein Abschneideoperator bei der durch den Knoten dargestellten Region zum Abschneiden der am Knoten dargestellte Region des Bildes zum Verwerfen der verborgenen Abschnitte des Bildes angewendet. Ist beispielsweise die durch einen Knoten dargestellte Region durch eine oder mehrere durch andere Knoten des Ausdrucksbaums dargestellte Regionen vollständig verborgen, wird der Knoten aus dem Ausdrucksbaum derart entfernt, dass eine Graphikoperation oder ein Graphikelement an dem Knoten nicht ausgeführt oder gerendert werden muss, was auch immer der Fall ist.

[0045] Ist ein Knoten teilweise durch eine oder mehrere durch andere Knoten im Ausdrucksbaum dargestellte Regionen verborgen, wird ein Abschneideoperator bei dem Knoten derart angewendet, dass bei der Ausführung eines Zusammensetzungsoperators im wesentlichen nicht-verborgene Regionen des Bildes, die am Knoten dargestellt werden, sich in der resultierenden Zusammensetzung der Region des Knotens befinden. Wird ein Bild aus einem Ausdrucksbaum zusammengesetzt und danach gerendert, der durch einen Abschneideoperator abgeschnittene Knoten enthält, werden im wesentlichen solche Abschnitte der Graphikelemente wiedergegeben oder gerendert, die von anderen Graphikelementen des Bildes nicht verborgen sind.

[0046] Die Anwendung eines Abschneideoperators bei einem Knoten kann in der einfachsten Form im Abschneiden der an den Nachkommenknoten dargestellten Graphikelemente auf im wesentlichen solche Abschnitte der Graphikelemente resultieren, die nicht verborgen sind. Allerdings ist die Anwendung eines Abschneideoperators bei einem Knoten nicht darauf beschränkt. Die Anwendung eines Abschneideoperators bei einem Knoten eines Ausdrucksbaums mit einer Zusammensetzungsoperation an dem Knoten kann in einer unterschiedlichen Kombination von Zusammensetzungsoperatoren resultieren, was eine Auswirkung auf den Knoten haben kann, als ob die dargestellte Region bis auf ihren nicht-verborgenen Abschnitt abgeschnitten ist.

[0047] Der Vorgang der Zusammensetzung eines in [Fig. 2A](#) gezeigten Ausdrucksbaums **101** wird nun unter Bezugnahme auf die [Fig. 2B](#) bis [Fig. 2D](#) beschrieben. Wie es in [Fig. 2B](#) gezeigt ist, wird ein Pfeil **102** im Uhrzeigersinn um 30° gedreht, und der „In“-Operator wird in Verbindung mit einer undurchsichtigen Box **104** ausgeführt, damit ein Abschnitt des gedrehten Pfeils erhalten wird, der in der Box **104** liegt. Dies kann durch die Anwendung eines Abschneideoperators bei dem im Uhrzeigersinn um 30° gedrehten Pfeil zum Abschneiden des gedrehten Pfeils bis auf die Grenzen der Box **104** bewirkt werden.

[0048] Alternativ dazu kann wie in [Fig. 2C](#) gezeigt die Anwendung einer anderen Kombination von Operatoren im wesentlichen das gleiche Endbildergebnis **105** liefern. Die Box **104** wird gegen den Uhrzeigersinn um 30° gedreht, und der Pfeil **102** wird auf die Box **104** zugeschnitten. Das resultierende Bild **107** wird um 30° im Uhrzeigersinn gedreht, um das Bildergebnis **105** zu erhalten. Wie in [Fig. 2D](#) gezeigt, ergibt dies aber nicht das Gleiche wie das Zuschneiden des Pfeils **102** auf die Box **104** und dann Anwenden einer Drehung um 30° im Uhrzeigersinn, um ein endgültiges zusammengesetztes Bild **106** zu erhalten. Auf diese Weise kann die Anwendung eines Abschneideoperators aus einem Knoten eine unterschiedliche Kombination von Zusammensetzungsoperatoren ergeben.

[0049] Wurde eine durch einen Knoten dargestellte Region des Bildes als nicht-verborgen oder lediglich teilweise verborgen bestimmt, übergibt der Knoten die Regionsdarstellung, die der Knoten von einem Elternknoten erhalten hat, wiederum jedem seiner Nachkommenknoten. Der gleiche Vorgang geschieht an jedem Nachkommenknoten mit dem Effekt, dass jeder Nachkommenknoten seinem Elternknoten entweder eine Bilddarstellung der Bereiche des Bildes, die durch die am Nachkommenknoten dargestellte Region verborgen sind, oder eine Anzeige zurückgibt, dass der Nachkommenknoten vollständig verborgen ist.

[0050] Wurden die Nachkommen eines Knotens verarbeitet, werden die von dem Nachkommen zurückgegebenen Regionsdarstellungen zum Herleiten einer Regionsdarstellung der Regionen des Bildes verwendet, die von dem Knoten verborgen sind. Dieses Ergebnis wird dem Elternknoten des Knotens zurückgegeben.

[0051] Bei dem bevorzugten Ausführungsbeispiel wird der Durchlauf des Ausdrucksbaums zum Vereinfachen des Baums an der Wurzel des Baums immer in der Tiefenrichtung zuerst initiiert, was dem Fachmann bekannt ist. Vorzugsweise wird beim Durchlaufen eines Ausdruckbaums in der Tiefenrichtung zuerst dem zum linken Zweig führenden Weg an jedem Knoten Priorität gegeben, und dieser Weg den Baum hinunter zu einem Nachkommenknoten wird zuerst genommen. Sind keine weiteren linken Zweigwege an einem aktuellen Knoten vorhanden, kehrt die Verarbeitung zu dem vorhergehenden Knoten zurück, und ein Weg einen rechten Zweig dieses Knotens hinunter wird genommen. Ein Ausdrucksbaum wird auf diese Weise durchlaufen, bis alle Knoten des Ausdruckbaums besucht wurden.

ABLAUFDIAGRAMME ZUR OPTIMIERUNG EINES AUSDRUCKSBAUMS

[0052] [Fig. 9](#) zeigt ein abstraktes Ablaufdiagramm, das eine Übersicht über den Vorgang der Optimierung eines Ausdrucksbaums **902** gemäß dem bevorzugten Ausführungsbeispiel liefert, der zum Zusammensetzen eines Bildes verwendet wird. Der Ausdrucksbaums **902** enthält zumindest zwei Knoten, und jeder Knoten ist entweder ein Graphikelement oder Graphikoperator. Vorzugsweise sind die Graphikoperatoren Bildzusammensetzungsoperatoren. Des weiteren wird eine Region des Bildes durch den Knoten dargestellt. Der Ausdrucksbaum **902** kann von Knoten zu Knoten durchlaufen werden. Ein Steuerblock **904** hat vorzugsweise eine Schleifensteuerstruktur zur Verarbeitung jedes Knotens des Ausdrucksbaums **902**, der als Eingangssignal zugeführt wird. Wurde der gesamte Ausdrucksbaum **902** verarbeitet (was durch „fertig“ angegeben wird), endet die Verarbeitung in Schritt **914**. Ansonsten wird die Verarbeitung in Schritt **906** fortgesetzt.

[0053] In Schritt **906** wird einer der verbleibenden Knoten als aktueller Knoten ausgewählt. In Schritt **908** wird die durch den Knoten dargestellte Region mit einer Regionsdarstellungsdatenstruktur verglichen, die einer oder mehreren Regionen entspricht, die durch zumindest einen weiteren Knoten dargestellt werden. Die Regionsdarstellung hat vorzugsweise die Form einer hierarchischen Datenstruktur, und kann eine Quadtree-Darstellung sein. Im Entscheidungsblock **910** wird überprüft, ob die durch den Knoten dargestellte Region durch eine der Regionen entweder ganz oder teilweise verborgen wird. Gibt der Entscheidungsblock **910** falsch (nein) zurück, wird die Verarbeitung am Steuerschritt **904** fortgesetzt. Ansonsten wird die Verarbeitung in Schritt **912** fortgesetzt, wenn der Entscheidungsblock **910** wahr (ja) zurückgibt. In Schritt **912** wird der Ausdrucksbaum modifiziert. Die Modifizierung kann die Entfernung des aktuellen Knotens oder das Ersetzen des aktuellen Knotens durch einen anderen Knoten des Ausdrucksbaums beinhalten. Sie kann ferner ein Abschneiden oder ein späteres Abschneiden der durch den aktuellen Knoten dargestellten Region enthalten. Die Verarbeitung wird dann im Steuerschritt **904** fortgesetzt.

[0054] [Fig. 10](#) zeigt ein ausführlicheres Ablaufdiagramm der Schritte zum Modifizieren des Ausdrucksbaums gemäß Schritt **912** in [Fig. 9](#). Die Verarbeitung beginnt in Schritt **1002**, und in Schritt **1004** wird überprüft, ob die durch den aktuellen Knoten dargestellte Region vollständig oder teilweise verborgen ist. Bestimmt der Entscheidungsblock **1004**, dass die Region teilweise verborgen ist, wird die Verarbeitung in Schritt **1012** fortgesetzt. In Schritt **1012** wird ein Abschneideoperator bei dem Knoten angewendet, und dann kehrt der Prozess zu Schritt **1014** zurück. Ansonsten wird die Verarbeitung im Entscheidungsblock **1006** fortgesetzt, wenn der

Endscheidungsblock **1004** bestimmt, dass die Region total verborgen ist.

[0055] Im Entscheidungsblock **1006** wird überprüft, ob der aktuelle Knoten ein Graphikelement oder Graphikoperator ist. Bestimmt der Entscheidungsblock **1004**, dass der Knoten ein Graphikelement ist, wird die Verarbeitung in Schritt **1008** fortgesetzt. In Schritt **1008** wird der Knoten aus dem Ausdrucksbaum entfernt und die Verarbeitung springt zur Aufrufprozedur in Schritt **1014** zurück. Ansonsten wird die Verarbeitung in Schritt **1010** fortgesetzt, wenn der Entscheidungsblock **1006** bestimmt, dass der Knoten ein Graphikoperator ist. In Schritt **1010** wird ein vorbestimmter Satz von Knotenersetzungsregeln entsprechend dem Graphikoperator angewendet.

[0056] Der vorbestimmte Satz an Knotenersetzungsregeln (in den [Fig. 9](#) und [Fig. 10](#) nicht gezeigt) kann eine oder mehrere der folgenden Regeln beinhalten:

ist der Elternknoten ein „Over“-Graphikoperator, und ist der aktuelle Knoten auf dem linken Zweig des Elternknotens, ersetze den Elternknoten durch einen rechten Unterbaum des Elternknotens,
 ist der Elternknoten ein „Over“-Graphikoperator und ist der aktuelle Knoten auf dem rechten Zweig des Elternknotens, ersetze den Elternknoten durch einen linken Unterbaum des Elternknotens,
 ist der Elternknoten ein „In“-Graphikoperator, entferne den Elternknoten und jeden vom Elternbaum verzweigenden Unterbaum,
 ist der Elternknoten ein „Ratop“-Graphikoperator und ist der aktuelle Knoten auf einem linken Zweig des Elternknotens, entferne den Elternknoten und jeden vom Elternknoten verzweigenden Unterbaum,
 ist der Elternknoten ein „Ratop“-Graphikoperator und ist der aktuelle Knoten auf einem rechten Zweig des Elternknotens, ersetze den Elternknoten durch einen linken Unterbaum des Elternknotens,
 ist der Elternknoten ein „Out“-Graphikoperator und ist der aktuelle Knoten auf einem linken Zweig des Elternknotens, entferne den Elternknoten und jeden vom Elternknoten verzweigenden Unterbaum,
 ist der Elternknoten ein „Out“-Graphikoperator und ist der aktuelle Knoten auf einem rechten Zweig des Elternknotens, ersetze den Elternknoten durch einen linken Unterbaum des Elternknotens,
 ist der Elternknoten ein „PlusC“-Graphikoperator und ist der aktuelle Knoten auf einem linken Zweig des Elternknotens, ersetze den Elternknoten durch einen rechten Unterbaum des Elternknotens,
 ist der Elternknoten ein „PlusC“-Graphikoperator und ist der aktuelle Knoten auf einem rechten Zweig des Elternknotens, ersetze den Elternknoten durch einen linken Unterbaum des Elternknotens,
 ist der Elternknoten ein „PlusW“-Graphikoperator oder ein „Xor“-Graphikoperator und ist der aktuelle Knoten auf einem linken Zweig des Elternknotens, ersetze den Elternknoten durch einen rechten Unterbaum des Elternknotens und
 ist der Elternknoten ein „PlusW“-Graphikoperator oder ein „Xor“-Graphikoperator und ist der aktuelle Knoten auf einem rechten Zweig des Elternknotens, ersetze den Elternknoten durch einen linken Unterbaum des Elternknotens.

[0057] [Fig. 11](#) zeigt ein ausführliches Ablaufdiagramm eines Vorgangs der Optimierung eines Ausdrucksbaums **1102** gemäß einem anderen Ausführungsbeispiel. Der Ausdrucksbaum umfasst eine Anzahl von Knoten, von denen jeder entweder ein graphisches Element oder ein Graphikoperator sein kann und eine Region des Bildes darstellt. Ein Steuerblock **1104** hat vorzugsweise eine Schleifensteuerstruktur zur Verarbeitung jedes Knotens des Ausdrucksbaums **1102**, der als Eingangssignal zugeführt wird. Der Ausdrucksbaum wird Knoten um Knoten durchlaufen. An jedem aktuellen Knoten, der einen Graphikoperator aufweist, werden die Schritte **1108** bis **1120** angewendet, wie es nachstehend beschrieben ist. Wurde der gesamte Ausdrucksbaum **1102** verarbeitet (was durch „fertig“ angegeben ist), vorzugsweise in der Tiefenrichtung zuerst, ist die Verarbeitung in Schritt **1106** beendet. Ansonsten wird die Verarbeitung in Schritt **1108** fortgesetzt.

[0058] In Schritt **1108** wird einer der verbleibenden Knoten als aktueller Knoten ausgewählt. In Schritt **1110** wird eine erste Regionsdarstellung von einem Elternknoten erhalten. In Schritt **1112** durchläuft eine modifizierte erste Regionsdarstellung einen ersten Operanden des Graphikoperators entsprechend einer ersten vorbestimmten Modifikationsregel für diesen Operator. In Schritt **1114** wird eine zweite Regionsdarstellung von durch einen Unterbaum des ersten Operanden verborgenen Regionen bestimmt, und zum Graphikoperator zurückgegeben. In Schritt **1116** durchläuft eine modifizierte zweite Regionsdarstellung einen zweiten Operanden des Graphikoperators gemäß einer zweiten vorbestimmten Modifikationsregel für den Operator.

[0059] In Schritt **1118** wird eine dritte Regionsdarstellung von Regionen, die durch einen mit dem zweiten Operanden verbundenen Unterbaum verborgen werden, zum Graphikoperator zurückgegeben. In Schritt **1120** wird eine endgültige Regionsdarstellung entsprechen einer Einstellregel für den Graphikoperator bestimmt und zu einem Elternknoten des aktuellen Knotens zurückgegeben. Vorzugsweise wird die Regel aus der folgenden Gruppe ausgewählt:

- (A) Ist der Graphikoperator ein „Over“- oder ein „PlusC“-Operator, wird die endgültige Regionsdarstellung aus einer Vereinigung der zweiten Regionsdarstellung und der dritten Regionsdarstellung bestimmt,
- (B) ist der Graphikoperator ein „In“-Operator, wird die endgültige Regionsdarstellung aus einer Schnittmenge der zweiten Regionsdarstellung und der dritten Regionsdarstellung bestimmt,
- (C) ist der Graphikoperator ein „Ratop“-Operator, ist die endgültige Regionsdarstellung die zweite Regionsdarstellung,
- (D) ist der Graphikoperator ein „Out“-Operator, wird die endgültige Regionsdarstellung aus einer Differenz der zweiten Regionsdarstellung und einer Regionsdarstellung mit zumindest einer durch eine Bounding-Box eines Knotens im rechten Unterbaum des aktuellen Knotens dargestellten Region bestimmt, und
- (E) ist der Graphikoperator ein „Xor“- oder ein „PlusW“-Operator, wird die endgültige Regionsdarstellung aus einer Vereinigung der zweiten Regionsdarstellung weniger einer Regionsdarstellung mit zumindest einer durch eine Bounding-Box eines Knotens im rechten Unterbaum des aktuellen Knotens dargestellten Region und der dritten Regionsdarstellung weniger einer Regionsdarstellung mit einer Bounding-Box eines Knotens im rechten Unterbaum des aktuellen Knotens bestimmt.

[0060] Die erste vorbestimmte Modifikationsregel besteht vorzugsweise im Weitergeben im wesentlichen der ersten Regionsdarstellung als modifizierte erste Regionsdarstellung, wenn der Graphikoperator ein „Over“- „In“- „Ratop“- „PlusC“- „PlusW“- „Xor“- „Out“- (linken Operanden zuerst vornehmen)-Operator oder dergleichen ist. Ist der Graphikoperator ein „Out“(rechten Operanden zuerst vornehmen)“-Vorgang, beinhaltet er das Weitergeben einer Vereinigung der ersten Regionsdarstellung mit der zweiten Regionsdarstellung als modifizierte erste Regionsdarstellung.

[0061] Des weiteren besteht die zweite vorbestimmte Modifikationsregel im wesentlichen im Weitergeben der ersten Regionsdarstellung als modifizierte zweite Regionsdarstellung, wenn der Graphikoperator ein „In“- „Ratop“- „Out“- „Plus“- „PlusW“- „Xor“-Operator oder dergleichen ist. Ist der Graphikoperator ein „Over“-Operator, beinhaltet er das Weitergeben einer Vereinigung der ersten Regionsdarstellung mit der zweiten Regionsdarstellung als modifizierte zweite Regionsdarstellung.

[0062] Vorzugsweise wird die Bilddarstellung nicht an einem Knoten erzeugt oder zu einem Elternknoten des Knotens zurückgegeben, wenn die Bilddarstellung nicht danach verwendet wird, oder wenn der Knoten der rechte Operand eines „Over“-Operators ist, und der „Over“-Operatorknoten keine Bilddarstellung zu seinem Elternknoten zurück zu geben braucht. Gleichermaßen wird die Bilddarstellung nicht am Knoten erzeugt oder zurückgegeben, wenn der Knoten der linke oder rechte Operand eines „In“- „PlusC“- „PlusW“- oder „Xor“-Operators ist, und der Operatorknoten keine Bilddarstellung zu seinem Elternknoten zurück zu geben braucht. Des weiteren wird die Bilddarstellung an dem Knoten eventuell nicht erzeugt oder zu dem Elternknoten zurückgegeben, wenn der Knoten der linke Operand eines „Out“- oder eines „Ratop“-Operators ist, und zu seinem Elternknoten keine Bilddarstellung zurück zu geben braucht. Die Bilddarstellung wird eventuell auch am Knoten nicht erzeugt oder zurückgegeben, wenn der Knoten der rechte Operand eines „Ratop“-Operators ist, die Wurzel des Ausdrucksbaums ist, der Operand einer Bildverbindung, affinen Transformation oder eines Faltungsoperators ist, der Operand einer Farbtransformation die Undurchsichtigkeit nicht bewahrt, oder wenn der Transformationsknoten keine Bilddarstellung zu seinem Elternknoten zurück zu geben braucht.

[0063] Weitere Ausgestaltungen des bevorzugten Ausführungsbeispiels sind in ihren Einzelheiten im Anhang dargestellt, der einen Teil der Beschreibung bildet. Insbesondere enthält der Anhang Pseudocodeprotokolle zum Implementieren des Verfahrens gemäß dem bevorzugten Ausführungsbeispiel. In diesem Zusammenhang ist das bevorzugte Ausführungsbeispiel vorzugsweise als Computersoftware implementiert, die auf Aufzeichnungsträgern gespeichert werden kann, die als auf einer Computereinrichtung ausgeführter Prozess ausgeführt werden kann, wie auf einem Universalcomputer.

[0064] Die Ausführungsbeispiele der Erfindung können vorzugsweise unter Verwendung eines herkömmlichen Universalcomputers ausgeübt werden, wie unter Verwendung eines in [Fig. 12](#) gezeigten, um Prozesse einschließlich der in den [Fig. 9](#) bis [Fig. 11](#) gezeigten auszuführen, sowie den im Anhang enthaltenen Pseudocode. Insbesondere werden die Schritte des Verfahrens zum Optimieren der Ausdrucksbäume durch Software-Befehle bewirkt, die durch den Computer ausgeführt werden. Das Computersystem **1200** umfasst den Computer **1202**, eine Videoanzeige **1216** und Eingabeeinrichtungen **1218**, **1220**. Außerdem kann das Computersystem **1200** eine beliebige Anzahl weiterer Ausgabeeinrichtungen einschließlich Zeilendrucker, Laserdrucker, Plotter und andere mit dem Computer **1202** verbundene Wiedergabeeinrichtungen umfassen. Das Computersystem **1200** kann mit einem oder mehreren Computern unter Verwendung eines geeigneten Kommunikationskanals, wie eines Modemkommunikationsweges, eines Computernetzes oder dergleichen verbunden sein.

[0065] Der Computer **1202** selbst umfasst eine Zentralverarbeitungseinheit beziehungsweise -Einheiten (die nachstehend einfach als Prozessor bezeichnet werden) **1204**, einen Speicher **1206**, der einen Speicher mit wahlfreiem Zugriff (RAM) und einen Nur-Lese-Speicher (ROM) enthalten kann, eine Eingangs-/Ausgangs-(IO)-Schnittstelle **1208**, eine Videoschnittstelle **1210** und eine oder mehrere Speichereinrichtungen, die allgemein durch einen Block **1212** in [Fig. 12](#) dargestellt sind. Die Speichereinrichtung(en) **1212** kann ein oder mehrere Elemente aus der folgenden Gruppe umfassen: Diskettenlaufwerk, Festplattenlaufwerk, magnetoptisches Plattenlaufwerk, CD-ROM oder eine beliebige weitere einer Anzahl nicht flüchtiger Speichereinrichtungen, die dem Fachmann bekannt sind. Jede der Komponenten **1204** bis **1212** ist typischerweise mit einer oder mehreren der anderen Einrichtungen über einen Bus **1214** verbunden, der wiederum aus Daten-, Adress- und Steuerbussen bestehen kann.

[0066] Die Videoschnittstelle **1210** ist mit der Videoanzeige **1216** verbunden und stellt Videosignale vom Computer **1202** zur Anzeige auf der Videoanzeige **1216** bereit. Benutzereingaben zum Bedienen des Computers **1202** können durch eine oder mehrere Eingabeeinrichtungen zugeführt werden. Beispielsweise kann der Bediener die Tastatur **1218** und/oder eine Zeigeeinrichtung wie eine Maus **1220** zur Bereitstellung einer Eingabe in den Computer **1202** verwenden. Beispielcomputer, auf denen das Ausführungsbeispiel praktiziert werden kann, beinhalten IBM-PC/ATs und kompatible Einrichtungen, sowie Sun Sparc Stations.

ERSTES BEISPIEL

[0067] [Fig. 3](#) zeigt ein Bild **10** aus einer Anzahl von Graphikelementen. Die Graphikelemente enthalten ein undurchsichtiges Bild A, das als Unterbild **11** bezeichnet wird, einen Kreis **12**, der als Kreis B bezeichnet wird, der durch das Unterbild **11** verborgen ist, und den Text „hello“ **13**, der optional als Text „C“ bezeichnet wird. Eine gestrichelte Linie **14** zeigt den Umfang des Bildes **10** und stellt eine leere Vordergrundregion dar, die nichts enthält, was das Bild **10** verdecken könnte.

[0068] [Fig. 4](#) zeigt einen Ausdrucksbaum **20**, der die Zusammensetzung des Bildes in [Fig. 3](#) darstellt. Nachstehend wird ein Beispiel der Vereinfachung des Ausdrucksbaums in [Fig. 4](#) beschrieben. Am Wurzelknoten (ersten Knoten) **21** des Ausdrucksbaums **20** übergibt ein computerimplementierter Vorgang dem ersten Knoten **21** eine leere Quadtree-Darstellung der leeren Region **14**, die das Bild **10** in [Fig. 3](#) nicht verbirgt, beziehungsweise äquivalent dazu einen weiteren Knoten über dem ersten Knoten **21** des Ausdrucksbaums **20** hat, um diesen zu verbergen.

[0069] Der erste Knoten **21** ist ein Zusammensetzungsoperator (das heißt, ein „Over“-Operator), der einen linken und rechten Operanden braucht. Der linke Operand ist ein Blattknoten **22**, der das Unterbild **11** in [Fig. 3](#) darstellt, und der rechte Operand wird durch einen zweiten Knoten **23** des Ausdrucksbaums zurückgegeben, der auch ein „Over“-Zusammensetzungsoperator ist.

[0070] Nach dem Empfang des leeren Quadtree im ersten Knoten **21** übergibt der Prozess den leeren Quadtree dem Blattknoten **22**. Am Blattknoten **22** wird der Quadtree typischerweise mit dem Unterbild **11** zur Bestimmung verglichen, ob das Unterbild **11** verborgen wird. Da der Quadtree bei diesem Beispiel aber ein leerer Quadtree ist, ist kein direkter Vergleich erforderlich, um das Ergebnis zu erhalten, dass das Unterbild **11** nicht durch den leeren Quadtree verborgen wird (verborgen werden kann).

[0071] Der Vergleich eines Quadtree mit einem Graphikelement (beispielsweise dem Unterbild **11**) bringt einen Vergleich mit sich, in dem durch den Quadtree dargestellte Regionen eines Bildes mit Regionen des Bildes verglichen werden, die durch das Graphikelement gedeckt werden, um zu bestimmen, ob eine Region eine andere Region des Bildes verbirgt. Der Vergleich einer Quadtree-Darstellung einer Region eines Bildes mit anderen Regionen des Bildes beinhaltet das Vergleichen der Region des Bildes mit den anderen Regionen entweder durch direkte Vergleiche ihrer jeweiligen Bereiche oder durch Vergleichen äquivalenter Darstellungen oder dergleichen.

[0072] Das durch den Blattknoten **22** dargestellte Unterbild **11** ist undurchsichtig und kann daher potenziell andere Graphikelemente des Bildes **10** verbergen. Eine erste Quadtree-Darstellung des Unterbildes **11** wird daher aufgebaut, die die Grenzen des Unterbildes **11** enthält, und wird zum ersten Knoten **21** zurückgegeben, da kein weiterer linker oder rechter Zweig am Blattknoten **22** des Ausdrucksbaums **10** verfügbar ist. Am ersten Knoten **21** führt der „Over“-Operator eine Vereinigung des ursprünglich zu diesem Knoten gegebenen Quadtree, der ein leerer Quadtree ist, und der vom linken Knoten zurückgegebenen Quadtree-Darstellung entsprechend den in Tabelle 2 aufgestellten Regeln für die Verarbeitung binärer Zusammensetzungsoperatoren durch.

[0073] Die Vereinigung eines leeren Quadtree mit der ersten Quadtreearstellung des Unterbildes **11** ergibt einen Quadtree, der der Quadtreearstellung äquivalent (oder im wesentlichen identisch) ist, und der nachstehend als der erste linke Quadtree bezeichnet wird.

[0074] Der erste linke Quadtree wird dem zweiten Knoten **23** des Ausdrucksbaums **10** übergeben, und wird wie in Verbindung mit dem Knoten **21** beschrieben dem linken Zweig des zweiten Knotens, d. h. einem Blattknoten **24** übergeben, der vom zweiten Knoten **23** abzweigt. Der Kreis **12** ist am Blattknoten **24** dargestellt. Beim Weitergeben des ersten linken Quadtree zum Blattknoten **24** vergleicht der Prozess den ersten linken Quadtree (das heißt, eine durch den ersten linken Quadtree dargestellte Bildregion) mit der Region des Bildes, die vom Kreis **12** belegt wird, wodurch zumindest bei diesem Beispiel herausgefunden wird, dass die Region des Kreises **12** in [Fig. 3](#) vollständig von der durch den ersten linken Quadtree dargestellten Region verborgen wird. Das Ergebnis, dass die Region des Kreises **12** vollständig verborgen ist, wird dem zweiten Knoten **23** zurückgegeben.

[0075] Der zweite Knoten **23** erhält typischerweise vom linken Knoten **24** eine Quadtreearstellung des Abschnitts des Bildes **10**, der vom Unterbild **11** und dem Kreis **12** verborgen ist (eine durch die Vereinigung des Unterbildes **11** und des Kreises erhaltene Region). Da bei diesem Beispiel aber der Kreis **12** vollständig vom Unterbild **11** verborgen wird, braucht keine Vereinigung der Quadtrees für das Unterbild **11** und den Kreis **12** durchgeführt werden.

[0076] Ein der ersten linken Quadtreearstellung des Unterbildes **11** im wesentlichen äquivalenter Quadtree wird dem zweiten Knoten **23** zurückgegeben, und dieser Quadtree wird einem rechten Blattknoten **25** übergeben, der vom zweiten Knoten **23** abzweigt. Der linke Blattknoten **25** des Ausdrucksbaums stellt eine Region des Bildes mit dem Text („hello“) **13** dar.

[0077] Der Text wird von dem vom zweiten Knoten **23** übergebenen Quadtree (der durch den Quadtree dargestellten Bildregion) nicht verborgen. Typischerweise wird eine Quadtreearstellung der Region des Bildes zum zweiten Knoten **23** zurückgegeben, die vom Graphikelement am rechten Blattknoten **25** verborgen wird. Da der Text aber in diesem Fall keine wesentliche Region (keinen wesentlichen Bereich) verbirgt, wird ein leerer Quadtree zum zweiten Knoten **23** zurückgegeben. Eine wesentliche Region ist vorzugsweise durch einen Leistungsgrad des Prozesses wie nachstehend beschrieben definiert.

[0078] Der zweite Knoten **23** empfängt den leeren Quadtree vom rechten Blattknoten **25**. Nach der Aktion (siehe Tabelle 2) eines „Over“-Operators am Knoten, wenn der linke Operand verborgen ist, ersetzt der zweite Knoten **23** sich selbst durch den rechten Blattknoten **25** und schneidet den linken „Zweig“ aus, was in diesem Beispiel der linke Blattknoten **24** ist. Der dem zweiten Knoten **23** zurückgegebene Quadtree (wenn auch der leere Quadtree) wird dem ersten Knoten **21** zurückgegeben.

[0079] Am ersten Knoten **21** wird keiner seiner Abkömmlinge abgeschnitten, und die Aktion eines „Over“-Operators besteht in der Bildung einer Vereinigung der Quadtrees, die zu dem „Over“-Operator der linken und rechten Zweige zurückgegeben wurden. Typischerweise wird das Ergebnis dieser Vereinigung dem Elternknoten des Knotens **21** zurückgegeben. Allerdings kann dieser Schritt bei diesem Beispiel wegoptimiert werden, da der erste Knoten **21** der oberste Knoten des Ausdrucksbaums (der Wurzelknoten) ist. Daher wird das Ergebnis der Vereinigung bei der Optimierung des Ausdrucksbaums nicht verwendet, und der vereinfachte Ausdrucksbaum ist in [Fig. 5](#) dargestellt, in dem der zweite Knoten **23** und der linke Blattknoten **24** aus dem Ausdrucksbaum in [Fig. 4](#) beseitigt wurden. Der vereinfachte Ausdrucksbaum in [Fig. 5](#) kann dann zum Rendern des Bildes in [Fig. 3](#) verwendet werden, ohne dass das Graphikelement gerendert werden muss, da der Kreis **12** als Graphikelement vom Unterbild **11** verborgen wird.

ZWEITES BEISPIEL

[0080] Nachstehend wird ein weiteres Beispiel der Vereinfachung (Optimierung) eines Ausdrucksbaums unter Bezugnahme auf die [Fig. 6](#) bis [Fig. 8](#) beschrieben. [Fig. 6](#) zeigt ein Bild **40** mit mehreren Graphikelementen: eine Seite „D“ **41**, ein undurchsichtiges Unterbild **42**, einen Text **43** und einen Kreis **44**. Ein entsprechender Ausdrucksbaum zum Zusammensetzen oder Rendern des Bildes **40** in [Fig. 6](#) ist in [Fig. 7](#) gezeigt. In [Fig. 7](#) stellen S0 bis S14 die bei diesem Beispiel vorgenommenen Schritte des bevorzugten Ausführungsbeispiels zum Vereinfachen oder Optimieren des Ausdrucksbaums **50** dar.

[0081] Die folgenden Schritte S0 bis S14 entsprechen der von einem Computer-implementierten Prozess an jedem Knoten bei der Vereinfachung des Ausdrucksbaums **50** in [Fig. 7](#) vorgenommenen Aktion.

[0082] S0: Ein leerer Quadtree q_0 wird erzeugt, der die leere Vordergrundregion **39** darstellt, die das gesamte Bild nicht verbirgt. Dieser leere Quadtree q_0 wird einem ersten Knoten **51** (oder Wurzelknoten) des Ausdrucksbaums **50** übergeben.

[0083] S1: Der erste Knoten **51** des Ausdrucksbaums **50** ist ein „Over“-Operator. Der Prozess erhält den leeren Quadtree q_0 , der dem ersten Knoten **51** vom vorhergehenden Schritt S0 zugeführt wurde, und vergleicht die durch den Quadtree q_0 dargestellte Region des Bildes mit einer durch den ersten Knoten **51** dargestellten Region, um zu bestimmen, ob die durch den ersten Knoten **51** dargestellte Region verborgen ist. Da q_0 ein leerer Quadtree ist und die durch den ersten Knoten **51** dargestellte Region nicht verbergen kann, geht der Prozess zu den Nachkommenknoten über. Zuerst wird der Quadtree q_0 einem zweiten Knoten **52** auf dem linken Zweig des Knotens **51** übergeben.

[0084] S2: Der zweite Knoten **52**, der hier ein „In“-Operator ist, erhält den leeren Quadtree q_0 . Der Quadtree q_0 wird mit einer durch den zweiten Knoten dargestellten Region zur Bestimmung verglichen, ob diese Region durch den Quadtree q_0 verborgen wird. Die Region des zweiten Knotens **52** wird durch den Quadtree q_0 nicht verborgen, da der Quadtree q_0 leer ist. Der Prozess wird in der Tiefenrichtung fortgesetzt und übergibt den Quadtree q_0 dem linken Zweig des zweiten Knotens **52**.

[0085] S3: Ein dritter Knoten **53** ist ein Blattknoten, der das Unterbild **42** darstellt. Dieser dritte Knoten **53** erhält den durch den Schritt S2 übergebenen Quadtree q_0 und vergleicht die Region dieses Knotens **53** mit der durch den Quadtree q_0 dargestellten Region, um zu bestimmen, ob die durch den Knoten **53** dargestellte Region vom Quadtree q_0 verborgen wird. In diesem Beispiel ist der Quadtree q_0 leer, und daher wird der Knoten **53** nicht verborgen. Allerdings ist das Bild „A“ ein Graphikelement, das andere Graphikelemente potenziell verbergen kann. Daher wird ein Quadtree q_0 erzeugt, der die durch das Bild verborgene Region darstellt, und zum zweiten Knoten **52** zurückgegeben, da keine weiteren linken Zweige am dritten Knoten **53** verfügbar sind.

[0086] S4: Der zweite Knoten **52** erhält vom dritten Knoten **53** den Quadtree q_1 , und da der zweite Knoten **52** ein „In“-Operator ist, wird der Quadtree q_1 im Speicher als verbergende Region des linken Operanden des „In“-Operators gespeichert. Die verbergende Region des linken Operanden eines Operators wird hier als q_L bezeichnet. So ist hier $q_L = q_1$. Die Aktion des Prozesses besteht gemäß Tabelle 2 darin, den am zweiten Knoten **52** erhaltenen, von dessen Elternknoten **51** übergebenen Quadtree dem rechten Nachkommenknoten **54** des Knotens **52** zu übergeben. In diesem Beispiel wird der dem zweiten Knoten **52** von dem ersten Knoten **51** (Elternknoten) übergebene Quadtree q_0 den rechten Zweig zum rechten Nachkommenknoten **54** hinuntergeschickt.

[0087] S5: Der rechte Nachkommenknoten **54** (fünfter Knoten) ist wiederum ein Blattknoten und stellt die durch den Kreis **54** angegebene Region dar. Der Quadtree q_0 wurde vom zweiten Knoten **52** nach Schritt S4 übergeben, und wird mit der mit dem Kreis **54** belegten Region des Bildes verglichen, um zu bestimmen, ob die durch den Quadtree q_0 dargestellte Region durch den Kreis **44** verborgen wird. Wiederum ist der Quadtree q_0 leer, und der Knoten **44** wird daher nicht verborgen. Allerdings ist der Kreis **44** ein Graphikelement (ein Objekt) mit der Möglichkeit des Verbergens von Graphikelementen (Objekten), die darunter liegen können. Daher wird ein die Region des Bildes darstellender Quadtree q_2 erzeugt, die durch den Kreis belegt wird. Der Quadtree q_2 wird dem zweiten Knoten **52** zurückgegeben, da keine weiteren Zweige an diesem Knoten **54** verfügbar sind.

[0088] S6: Der zweite Knoten **52** erhält den von seinem rechten Nachkommenknoten **54** zurückgegebenen Quadtree q_2 , und der Quadtree q_2 wird als verbergende Region des rechten Operanden des „In“-Operators gespeichert (das heißt, $q_R = q_2$). Der Prozess wird entsprechen der in Tabelle 2 festgelegten Aktion für den „In“-Operator fortgesetzt. Die Schnittmenge der durch die zwei Operanden des Prozesses dargestellten Regionen werden zum Elternknoten (das heißt, zum ersten Knoten **51**) zurückgegeben (das heißt, das Unterbild **42** mit der Region des Kreises **44**). Das Zuschneiden resultiert in der durch den Abschnitt des Unterbildes **42** dargestellten Region, der mit dem Kreis **44** (das heißt, dem Quadtree q_2) übereinstimmt. In diesem Fall stellt die Schnittmenge $q_L \cap q_R = q_1 \cap q_2 = q_2$ die Region dar, in der der Knoten **52** andere Graphikelemente verbergen kann.

[0089] S7: Der erste Knoten **51** erhält den vom zweiten Knoten **52** zurückgegebenen Quadtree q_2 . Der Quadtree q_2 wird als verbergende Region des linken Operanden des „Over“-Operators gespeichert ($q_L = q_2$). Gemäß Tabelle 2 ist die durchzuführende Aktion, wenn ein Nachkomme eines rechten Zweiges eines Knotens einen „Over“-Operator umfasst, eine Vereinigung (das heißt, $q_0 \cup q_L = q_0 \cup q_2 = q_2$) des Quadtree q_0 und des Quadtree q_L , die vom zweiten Knoten **52** zurückgegeben wird, den rechten Zweig hinunter zu schicken. Das

Ergebnis dieser Vereinigung ($q_0 \cup q_L$) ist ein im wesentlichen mit q_2 identischer Quadtree. Daher wird das Ergebnis dieser Vereinigung (der Quadtree q_2) den rechten Zweig zu einem fünften Knoten **55** hinuntergeschickt, der auch einen „Over“-Operator darstellt.

[0090] S8: Die Region, die durch den dem fünften Knoten **55** zugeführten Quadtree q_2 dargestellt wird, wird mit der am fünften Knoten **55** dargestellten Region zur Bestimmung verglichen, ob die Region des Knotens **55** durch den Quadtree q_2 (dessen Region) verborgen wird. Die am fünften Knoten **55** dargestellte Region des Bildes wird nicht durch die Region des Quadtree q_2 verborgen. Der Quadtree q_2 wird den linken Zweig hinuntergeschickt, der vom fünften Knoten **55** abzweigt.

[0091] S9: Der linke Nachkomme des fünften Knotens **55** ist ein Blattknoten **56**, der die Region des Bildes in [Fig. 6](#) darstellt, die den Text **43** zeigt. Der Blattknoten **56** erhält den vom fünften Knoten **55** übergebenen Quadtree q_2 und wird mit der am Blattknoten **56** dargestellten Region (typischerweise ist die Region des Bildes **6**, die vom Text **43** belegt wird, eine Bounding-Box mit Text) zur Bestimmung verglichen, ob die durch den Quadtree q_2 dargestellte Region die am Blattknoten **56** dargestellte Region verbirgt. Die durch den Quadtree q_2 dargestellte Region (die durch den Kreis **44** belegte Region) verbirgt den Text **43** teilweise. Daher wird der Text **43** für ein späteres Abschneiden begrenzt oder gekennzeichnet. Der Text **43** wird durch die Anwendung eines Abschneideoperators abgeschnitten, wobei die Abschneideoperation einen „Schneide-Weg“ aus dem Quadtree q_2 konstruiert und den Text **43** bis zu diesem Weg wegschneidet oder abschneidet.

[0092] An diesem Punkt wird typischerweise ein neuer Quadtree erzeugt, der die durch den Text belegte Region des Bildes darstellt, und zu dem fünften Knoten **55** zurückgegeben. Ist aber bei diesem Ausführungsbeispiel ein Graphikelement zu klein, um andere Graphikelemente des Bildes wesentlich zu verbergen (beispielsweise verbirgt das graphische Textelement „hello“ **43** andere Graphikelemente nicht wesentlich, obwohl die Bounding-Box des Texts **43** eine wesentliche Region darstellt), wird vorzugsweise ein leerer Quadtree zurückgegeben, anstatt Verarbeitungszeit zu investieren, um eine Quadtreedarstellung der Region des Textes **43** zu erzielen. Daher wird die Erzeugung eines neuen Quadtree q_3 für Regionen des Bildes, die durch den Text **43** belegt sind, als leerer Quadtree bewirkt. Die Wahl der Erzeugung eines leeren Quadtree für die durch den Text **43** dargestellte Region ist ein Leistungsmerkmal des Prozesses, was nachstehend unter der Unterüberschrift „Leistungsmerkmale“ beschrieben ist. Obwohl eine Quadtreedarstellung für den Text **43** erzeugt werden kann, übersteigt die Arbeitsgeschwindigkeit des Prozesses die zum Rendern des Textes erforderliche Zeit. Daher wird der leere Quadtree q_3 erzeugt und dem fünften Knoten **55** zurückgegeben.

[0093] S10: Der fünfte Knoten **55** erhält den durch den vorhergehenden Schritt S9 zurückgegebenen leeren Quadtree q_3 . Dieser Quadtree q_3 wird als verbergende Region des linken Operanden des „Over“-Operators am fünften Knoten **55** gespeichert ($q_L = q_3$). Wiederum besteht gemäß Tabelle 2 die durchzuführende Aktion beim Durchlaufen eines rechten Zweiges eines Knotens mit einem „Over“-Operator im Übergeben einer Vereinigung des dem Knoten **55** durch den Elternknoten **51** übergebenen Quadtree q_2 mit dem mit dem linken Operanden verbundenen Quadtree q_3 zu dem rechten Zweig ($q_L \cup q_2 = q_3 \cup q_2 = q_2$). Die Vereinigung des Quadtree q_3 mit dem Quadtree q_2 resultiert in einem zu dem Quadtree q_2 äquivalenten Quadtree, da der Quadtree q_3 der in Schritt S9 beschriebene leere Quadtree ist. Daher wird der Quadtree q_2 dem rechten Zweig des Ausdrucksbaums für einen rechten Blattknoten **57** des Elternknotens (fünfter Knoten) **55** übergeben.

[0094] Der Blattknoten **57** ist durch ein Graphikelement Seite „D“ **41** dargestellt, die die Hintergrundseite in [Fig. 6](#) darstellt. Der dem rechten Blattknoten **57** durch den fünften Knoten **55** übergebene Quadtree q_2 wird mit der Region der Seite „D“ **41** zur Bestimmung verglichen, ob die durch den Quadtree q_2 dargestellte Region die durch die Seite „D“ **41** dargestellte Region verbirgt. Das Ergebnis dieses Vergleichs ist, dass die durch den Quadtree q_2 (Kreis **49**) dargestellte Region die Seite „D“ **41** teilweise oder ganz verbirgt.

[0095] Das Graphikelement Seite „D“ wird daher entweder markiert, um später auf die Grenze des Kreises **44** zugeschnitten zu werden (einen aus dem Quadtree q_2 hergeleiteten Abschneideweg), was typischerweise vor einem Render-Vorgang geschieht, oder es wird ein Abschneideoperator angewendet, und die Seite „D“ **41** wird zugeschnitten, sodass die durch den Kreis **44** beschriebene Region aus der Seite „D“ **41** ausgeschnitten wird. Zur Darstellung der Seite „D“ **41** kann ein Quadtree erzeugt werden, sodass er zurück zu einem Elternknoten gegeben werden kann. Allerdings ist bei diesem Beispiel die Erzeugung eines derartigen Quadtree nicht erforderlich, da entnommen werden kann, dass keine weiteren Graphikelemente verborgen sein können.

[0096] S12: Der Prozess kehrt zum fünften Knoten **55** zurück, an dem keine weiteren Quadtrees zu erzeugen sind.

[0097] S13: Der Prozess kehrt zum ersten Knoten **51** zurück, an dem keine weiteren Quadrees zu erzeugen sind.

[0098] S14: Der Prozess endet mit der Optimierung des Ausdrucksbaums **50** in [Fig. 7](#), womit der Ausdrucksbaum **60** in [Fig. 8](#) erhalten wird. Die Symbole **58** und **59** in Diamantform in [Fig. 8](#) zeigen, dass der Text **43** und die Seite „D“ **41** jeweils zuzuschneiden sind (oder auch zugeschnitten wurden).

LEISTUNGSMERKMALE

[0099] Die vorstehenden Beispiele der Quadreedarstellungen, die unter Bezugnahme auf die [Fig. 1](#) bis [Fig. 8](#) beschrieben wurden, umfassen die Darstellung einer Region eines Bildes, die von einem Graphikelement (Objekt) belegt wird, ungeachtet der relativen Größe des Graphikelements verglichen mit dem gesamten Bild. Allerdings wird der bei dem Ausführungsbeispiel durchgeführte Prozess vorzugsweise von den folgenden Prinzipien und Folgerungen gesteuert:

- (a) Es wird bevorzugt, so wenig als möglich zur Abdeckung der meisten Fälle zu tun, anstelle zu versuchen, perfekte Ergebnisse zu erzielen, und
- (b) an einem Knoten ist zumindest zu Beginn nicht bekannt, ob tatsächlich eine Verbergung in einem Bild geschieht oder nicht, weshalb es zu bevorzugt ist, kostspielige Tests mit unsicherem Ausgang zu vermeiden. Diese Prinzipien gelten im Folgenden.

[0100] Zum einen erhöht die Erhöhung der Tiefe (das heißt, die Anzahl von Knoten und Zweigen) eines Quadtree die Quadtreeauflösung und die Fähigkeit zur Erfassung einer Verbergung. Allerdings erhöhen sich ab einer bestimmten Auflösung die Berechnungskosten zur Erzeugung und Kombination von Quadrees exponentiell, was die Verarbeitungseinsparungen durch das Versuchen der Beseitigung verschwindender Bereiche aus einem Ausdrucksbaum, die durch die erhöhte Quadreetiefe dargestellt werden, überschreitet.

[0101] Außerdem ist es bezüglich der Berechnung kostspielig, jedes undurchsichtige Primitiv als potenziellen Verberger zu behandeln (als Graphikelement, das wahrscheinlich ein anderes Graphikelement eines Bildes verbirgt). Je kleiner das Primitiv ist, desto weniger wahrscheinlich verbirgt es ein anderes Primitiv. Daher ist die Erzeugung von Quadrees vorzugsweise auf mögliche Verberger begrenzt, die eine vorbestimmte Größe haben oder größer sind. Typischerweise werden Primitive, deren Umwandlung in einen Quadtree zu kostspielig ist, nicht betrachtet, da sie kein gutes Investitionsergebnis garantieren können. Ein „guter Verberger“ hat daher vorzugsweise die folgenden Merkmale:

- (a) Vollständig undurchsichtig,
- (b) Größer als eine vorbestimmte Größe (und daher sehr wahrscheinlich ein Verberger anderer Primitive eines Bildes),
- (c) Einfach in einen Quadtree sehr schnell umzuwandeln (beispielsweise werden lediglich Graphikelemente mit einer einzigen einfachen konvexen Umrisslinie gewählt).

[0102] Zum Dritten kann das Testen auf eine Verbergung (das heißt, die Bestimmung, ob ein erstes Graphikelement eines oder mehrerer Graphikelemente eines Bildes verbirgt) durch die Darstellung der durch das erste Graphikelement abgedeckten Region als Quadtree und Testen durchgeführt werden, ob eine oder mehrere Zellen der an den Knoten des Quadtree dargestellten Region durch das eine oder die mehreren Graphikelemente des Bildes abgedeckte Regionen verbergen. Typischerweise sind die eine oder mehreren Regionen auch durch Quadrees dargestellt, und die Zellen der Quadrees werden verglichen. Allerdings kann sich die Darstellung einer beliebigen Region eines Bildes als Quadreedarstellung ab einer bestimmten Auflösung als sehr rechenintensiv, obwohl insgesamt möglich, erweisen. Daher wird vorzugsweise eine Bounding-Box einer an einem Knoten eines Ausdrucksbaums dargestellten Region konstruiert. Ob der Knoten ein Graphikelement oder ein Operator ist, die am Ausdrucksbaumknoten dargestellte Region ist gut definiert.

[0103] Obwohl die Bounding-Box an einem Knoten eines Ausdrucksbaums die durch den Knoten abgedeckte Region eventuell nicht exakt darstellt, macht die Verbesserung der Rechenleistung typischerweise den Nachteil der Verarbeitung dahingehend, dass Verborgenenheiten nicht erfasst werden, wett. Das Testen auf verborgene Graphikelemente durch das Vergleichen ihrer jeweiligen Bounding-Box wird gegenüber dem Vergleichen einer Quadreedarstellung der Regionen des Bildes bevorzugt, die durch die Graphikelemente belegt werden. Dadurch können manche verborgenen Graphikelemente unterhalb einer vorbestimmten Größe übersehen und als nicht verborgen betrachtet werden. Allerdings ist die Auswahl eines einfachen Tests die Bestimmung, ob Graphikelemente durch andere Graphikelemente des Bildes verborgen werden, gegenüber rechenintensiver Tests zu bevorzugen, die in den meisten Fällen die Investition nicht rechtfertigen.

[0104] Im Folgenden wird ein Beispiel eines Pseudo-Code-Aufrufs einer Routine „test“ beschrieben, die die Bounding-Box an einem Knoten mit einer Quadtreezelle (die nachstehend „Zelle“ genannt wird) vergleicht.

```
function test (bounding_box, Zelle)
begin
  if Zelle ist voll then
    return true (Verbergung)
  else if Zelle ist leer then
    return false (keine Verbergung)
  else begin
    Zelle wird unterteilt.
    if bounding_box und obere rechte Unterzelle keine
    leere Schnittmenge haben then
      if not test (bounding_box, obere rechte
      Unterzelle) then
        return false
      if bounding_box und obere linke Unterzelle keine
      leere Schnittmenge haben then
        if not test (bounding_box, obere linke
        Unterzelle) then
          return false
    if bounding_box und untere rechte Unterzelle keine
    leere Schnittmenge haben then
      if not test (bounding_box, untere rechte
      Unterzelle) then
        return false
    if bounding_box und untere linke Unterzelle keine
    leere Schnittmenge haben then
      if not Text (bounding_box, untere linke
      Unterzelle) then
        return false
    return true
  end
end
```

[0105] Diese Funktion (Routine) wird aufgerufen durch:
 if bounding_box keine leere Schnittmenge mit dem durch die Quadtree-Wurzel dargestellten Rechteck hat then
 call test (bounding_box, Quadtree-Wurzel)

[0106] Quadrees werden kontinuierlich erzeugt und verworfen. Ein sehr einfaches und schnelles Schema zur Verwaltung des Computerspeichers mit geringer Speicherzuweisungsaktivität wird bevorzugt (beispielsweise die Zuweisung ziemlich großer Speicherblöcke, das heißt, 1000 Quadtree-Zellen auf einmal). Zellen werden diesen Blöcken zugewiesen, als nur einmal beschreibbar und nur lesbar behandelt, und bis zum Ende des ge-

samten Ausdrucksbaumdurchlaufs nicht freigegeben. Dieser Ansatz ermöglicht das gemeinsame Nutzen von Zellen unter Quadrees und verringert erheblich ein Kopieren der Durchführung von Quadreeinstelloperationen.

[0107] Muss keine Regionsdarstellung (kein Quadtree) erzeugt werden, wird vorzugsweise als Leistungsmerkmal an einem Knoten keine Regionsdarstellung erzeugt. Beispielsweise kann ein Elternknoten von einem Nachkommenknoten einen Quadtree der Region anfordern, die der Nachkommenknoten und seine Nachkommenknoten verbergen können. Wird eine Regionsdarstellung bei einer nachfolgenden Berechnung nie verwendet, muss die Regionsdarstellung vorzugsweise nicht erzeugt werden.

[0108] Der vorstehende Prozess zur Optimierung eines Ausdrucksbaums ist für eine angemessene Darstellung als Rekursion beschrieben. Die Implementierung des Prozesses ist auch unter Verwendung eines nicht rekursiven Prozesses durch Rück-Zeiger möglich. Dies dient sowohl der Verringerung eines Funktionsaufrufsoverheads und auch der Handhabung sehr großer Bäume, die in der Praxis kaum harmonisch sind.

[0109] Vorstehend wurde lediglich eine kleine Anzahl von Ausführungsbeispielen der Erfindung beschrieben, und es können Modifikationen anhand der vorstehenden Beschreibung vom Fachmann durchgeführt werden, ohne vom Schutzbereich der Erfindung abzuweichen.

ANLAGE

[0110] Die folgende Funktion testet einen Knoten auf eine Verbergung gegenüber einem Quadtree q0. Sie gibt zurück, ob alle sichtbaren Teile des Knotens verborgen sind oder nicht. Bei Ergebnis_erforderlich gibt sie auch eine Quadtreedarstellung zurück, welche Bereiche der Knoten verbirgt.

[0111] Sie wird mit dem Aufruf:

obscure (Wurzelknoten des Baums, false, leerer Quadtree)
aufgerufen.

function obscure (Knoten, Ergebnis_erforderlich, q0)

begin

case Knotentyp begin

Primitiv →

if q0 verbirgt die Knotenboundingbox then

return verborgen

else if q0 verbirgt teilweise die Knotenboundingbox and es wird ein Vorteil beim Beschneiden des Primitivs erzielt (beispielsweise handelt es sich um ein Bild-, Kantenmisch-Box-, all- oder Wegprimitiv) then

begin

Zuschneiden, wenn der Zuschneideoverhead die Einsparung der Nichterzeugung und Zusammensetzung der abgeschnittenen Bildelemente es wert ist.

Einen Abschneideweg aus q0 erhalten. Dieser Abschneideweg bleibt mit q0 verbunden, während q0 vorhanden ist, sodass er lediglich einmal erzeugt wird.

Den Knoten markieren, dass er ein Zuschneiden auf diesen Weg erfordert.

end

if Ergebnis_erforderlich then

begin

if das Primitiv ein guter Verberger ist (ein großes undurchsichtiges Bild, eine Box oder all; ein großer undurchsichtiger Weg mit einer einzelnen, einfachen, konvexen Kante) then

einen Quadtree aus der Primitivgrenze konstruieren.

return diesen Quadtree.

else

return leeren Quadtree.

end

Farbtransformation →

if q0 verbirgt die Knotenboundingbox then

return verborgen.

else if q0 verbirgt teilweise die Knotenboundingbox then

begin

Zuschneiden, da der Abschneideoverhead die Einsparung der Nichttransformation der nicht abgeschnittenen Bildelemente es wert ist.

Abschneideweg aus q0 erhalten. Dieser Abschneideweg bleibt mit q0 verbunden, während q0 vorhanden ist,

sodass er lediglich einmal erzeugt wird.

Knoten markieren, dass er ein Zuschneiden auf diesen Weg erfordert.

end

Bestimmen, ob die Transformation die Undurchsichtigkeit bewahrt

(Opazität 1 bildet auf Opazität 1 ab), und ob sie die Unsichtbarkeit bewahrt (Opazität 0 bildet auf Opazität 0 ab).

call obscure (Knotenoperand, Transformation bewahrt Undurchsichtigkeit and Ergebnis_erforderlich q0), Quadtree q1 auf Anforderung erhalten. Wird die Undurchsichtigkeit nicht bewahrt, ist nicht bekannt, welche Bereiche nach der Anwendung der Transformation verborgen sein werden, so dass es keinen Anlass zur Anforderung eines Quadtree gibt.

if Operand ist verborgen then

begin

soll der Operand verborgen sein, dann sind es lediglich die sichtbaren Abschnitte (Opazität #0), die garantiert verborgen sein werden.

if Transformation bewahrt Unsichtbarkeit then

return verborgen.

else

begin

Bestimmen, was die Transformation unsichtbar transformieren wird (Opazität = 0). Diesen Knoten durch ein „All“-Primitiv dieser Farbe (Opazität) ersetzen.

if Ergebnis_erforderlich then

return einen aus den All-Grenzen konstruierten Quadtree.

return

end

end

if Ergebnis_erforderlich then

if Transformation bewahrt Undurchsichtigkeit then

return Quadtree q1.

else

return leeren Quadtree.

Affine Transformation, Bildverwindung →

if q0 verbirgt die Knotenboundingbox then

return verborgen.

else if q0 verbirgt teilweise die Knotenboundingbox then

begin

Zuschneiden, da der Abschneideoverhead die Einsparung der Nichterzeugung und Zusammensetzung der zugeschnittenen Bildelemente es wert ist.

Abschneideweg aus q0 erhalten. Dieser Abschneideweg bleibt mit q0 verbunden, wenn q0 vorhanden ist, sodass er lediglich einmal erzeugt wird.

Den Knoten markieren, dass er ein Zuschneiden auf diesen Weg erfordert.

end

call obscure (Knotenoperand, false, leerer Quadtree). q0 kann im Baum nicht weitergereicht werden oder es kann kein Ergebnis akzeptiert werden, wenn die Quadtrees nicht durch die Transformation invertiert/transformatiert werden.

Bildfaltung →

if q0 verbirgt die Knotenboundingbox then

return verborgen

call obscure (Knotenoperand, false, leerer Quadtree).

Binäroperator →

if q0 verbirgt die Knotenboundingbox then

return verborgen

case Knotenoperator begin

over →

call obscure (linker Operand des Knotens, true, q0), durch linken Operanden verborgenen Bereich qL erhalten.

call obscure (rechter Operand des Knotens, Ergebnis_erforderlich, ist linker Operand verborgen then q0 else q0 ∪ qL), durch rechten Operanden verborgenen Bereich qR bei Ergebnis_erforderlich erhalten.

if linker Operand ist verborgen und rechter Operand ist verborgen then

return verborgen.

else if linker Operand ist verborgen then

begin

Ersetze diesen Knoten durch seinen rechten Operanden.

```

if Ergebnis_erforderlich then
return qR.
end
else if rechter Operand ist verborgen then
begin
Diesen Knoten durch seinen linken Operanden ersetzen.
if Ergebnis_erforderlich then
return qL.
end
else
if Ergebnis_erforderlich then
return qL u qR.
end
end
in →
call obscure (linker Operand des Knotens, Ergebnis_erforderlich, q0), durch linken Operanden verborgenen
Bereich qL bei Ergebnis_erforderlich erhalten.
if linker Operand ist verborgen then
return verborgen.
call obscure (rechter Operand des Knotens, Ergebnis_erforderlich, q0), durch rechten Operanden verborgenen
Bereich qR bei Ergebnis_erforderlich erhalten.
if rechter Operand ist verborgen then
return verborgen.
if Ergebnis_erforderlich then
return qL u qR.
out
call obscure (rechter Operand des Knotens, true, q0), durch rechten Operanden verborgenen Bereich qR er-
halten.
call obscure (linker Operand des Knotens, Ergebnis_erforderlich, ist rechter Operand verborgen then q0 else
q0 n qR), durch linken Operanden verborgenen Bereich qL bei Ergebnis_erforderlich erhalten.
if linker Operand ist verborgen then
return verborgen.
else if rechter Operand ist verborgen then
begin
Diesen Knoten durch seinen linken Operanden ersetzen.
if Ergebnis_erforderlich then
return qL.
end
else
if Ergebnis_erforderlich then
return qL-B(rechter Operand).
end
ratop →
call obscure (linker Operand des Knotens, Ergebnis_erforderlich, q0), durch linken Operanden verborgenen
Bereich qL bei Ergebnis_erforderlich erhalten.
if linker Operand ist verborgen then
return verborgen.
call obscure (rechter Operand des Knotens, false, q0).
if rechter Operand ist verborgen then
Diesen Knoten durch seinen linken Operanden ersetzen.
if Ergebnis_erforderlich then
return qL.
plusC →
call obscure (linker Operand des Knotens, Ergebnis_erforderlich, q0), durch linken Operanden verborgenen
Bereiche qL bei Ergebnis_erforderlich erhalten.
call obscure (rechter Operand des Knotens, Ergebnis_erforderlich, q0), durch rechten Operanden verborgenen
Bereich qR bei Ergebnis_erforderlich erhalten.
if linker Operand ist verborgen und rechter Operand ist verborgen then
return verborgen.
else if linker Operand ist verborgen then
begin

```

```

Diesen Knoten durch seinen rechten Operanden ersetzen.
if Ergebnis_erforderlich then
return qR.
end
else if rechter Operand ist verborgen then
begin
Diesen Knoten durch seinen linken Operanden ersetzen.
if Ergebnis_erforderlich then
return qL.
end
else
if Ergebnis_erforderlich then
return qL u qR.
end
plusW, Xor →
call obscure (linker Operand des Knotens, Ergebnis_erforderlich, q0), durch linken Operanden verborgenen
Bereich qL bei Ergebnis_erforderlich erhalten.
call obscure (rechter Operand des Knotens, Ergebnis_erforderlich, q0), durch rechten Operanden verborgenen
Bereich qR bei Ergebnis_erforderlich erhalten.
if linker Operand ist verborgen and rechter Operand ist verborgen then
return verborgen.
else if linker Operand ist verborgen then
begin
Diesen Knoten durch seinen rechten Operanden ersetzen.
if Ergebnis_erforderlich then
return qR.
end
else if rechter Operand ist verborgen then
begin
Diesen Knoten durch seinen linken Operanden ersetzen.
if Ergebnis_erforderlich then
return qL.
end
else
begin
if Ergebnis_erforderlich then
return (qL-B(rechter Operand)) u (qR-B(linker Operand)).
end
end case Binäroperator
end case Knotentyp
end

```

Patentansprüche

1. Verfahren zur Optimierung eines Ausdrucksbaums (**101**, **902**, **1102**), der zum Zusammensetzen eines Bildes dient und zumindest drei Knoten umfasst, wobei jeder Knoten des Baums zumindest entweder ein eine Bildregion definierendes Grafikelement oder ein Grafikoperator ist, wobei das Verfahren für zumindest einen Knoten im Baum die Schritte umfasst

Vergleichen (**908**) einer ersten Region des Knotens mit einer zweiten Region, die von zumindest einem anderen Knoten irgendwo im Ausdrucksbaum erhalten wird,

Bestimmen (**910**), ob die erste Region durch die zweite Region ganz oder teilweise verborgen ist, und

Modifizieren (**912**) des Ausdrucksbaums, wenn die erste Region zumindest teilweise oder ganz durch die zweite Region verborgen ist, um einen optimierten Ausdrucksbaum zum Zusammensetzen des gleichen Bildes auszubilden, wobei ein optimierter Teil des Ausdrucksbaums im Wesentlichen nicht-verborgene Abschnitte der ersten Region darstellt.

2. Verfahren nach Anspruch 1, wobei der Schritt des Modifizierens (**912**) des Ausdrucksbaums die Anwendung (**1012**) eines Abschneideoperators bei dem Knoten beinhaltet, wenn die erste Region teilweise verborgen ist.

3. Verfahren nach Anspruch 1, wobei der Schritt des Modifizierens (**912**) des Ausdrucksbaums die Schritte enthält, wenn der Knoten ganz verborgen ist:
Entfernen (**1008**) des Knotens und,
wenn der Knoten einen Elternknoten mit einem Grafikoperator hat, Auswählen (**1010**) einer Knotenersetzungsregel aus einem vorbestimmten Satz von Knotenersetzungsregeln entsprechend dem Grafikoperator und Anwenden (**1010**) der Regel.
4. Verfahren nach Anspruch 3, wobei der vorbestimmte Satz von Knotenersetzungsregeln zumindest einen aus der folgenden Gruppe ausgewählten Schritt umfasst:
wenn der Elternknoten einen Grafikoperator „over“ darstellt, und der aktuelle Knoten sich in einem linken Zweig des Elternknotens befindet, Ersetzen des Elternknotens durch einen rechten Unterbaum des Elternknotens,
wenn der Elternknoten einen Grafikoperator „over“ darstellt, und der aktuelle Knoten sich in einem rechten Zweig des Elternknotens befindet, Ersetzen des Elternknotens durch einen linken Unterbaum des Elternknotens,
wenn der Elternknoten einen Grafikoperator „in“ darstellt, Entfernen des Elternknotens und jedes Unterbaums, der vom Elternknoten verzweigt,
wenn der Elternknoten einen Grafikoperator „ratop“ darstellt, und der aktuelle Knoten sich in einem linken Zweig des Elternknotens befindet, Entfernen des Elternknotens und jedes Unterbaums, der vom Elternknoten verzweigt,
wenn der Elternknoten einen Grafikoperator „ratop“ darstellt, und der aktuelle Knoten sich in einem rechten Zweig des Elternknotens befindet, Ersetzen des Elternknotens durch einen linken Unterbaum des Elternknotens,
wenn der Elternknoten einen Grafikoperator „out“ darstellt, und der aktuelle Knoten sich in einem linken Zweig des zweiten Knotens befindet, Entfernen des Elternknotens und jedes Unterbaums, der vom Elternknoten verzweigt,
wenn der Elternknoten einen Grafikoperator „out“ darstellt, und der aktuelle Knoten sich in einem rechten Zweig des Elternknotens befindet, Ersetzen des Elternknotens durch einen linken Unterbaum des Elternknotens,
wenn der Elternknoten einen Grafikoperator „plusC“ darstellt, und der aktuelle Knoten sich in einem linken Zweig des Elternknotens befindet, Ersetzen des Elternknotens durch einen rechten Unterbaum des Elternknotens,
wenn der Elternknoten einen Grafikoperator „plusC“ darstellt, und sich der aktuelle Knoten in einem rechten Zweig des Elternknotens befindet, Ersetzen des Elternknotens durch einen linken Unterbaum des Elternknotens,
wenn der Elternknoten einen Grafikoperator „plusW“ oder „Xor“ darstellt, und sich der aktuelle Knoten in einem linken Zweig des Elternknotens befindet, Ersetzen des Elternknotens durch einen rechten Unterbaum des Elternknotens, und
wenn der Elternknoten einen Grafikoperator „plusW“ oder „Xor“ darstellt, und sich der aktuelle Knoten in einem rechten Zweig des Elternknotens befindet, Ersetzen des Elternknotens durch einen linken Unterbaum des Elternknotens.
5. Verfahren nach einem der Ansprüche 1 bis 4, wobei die Grafikoperatoren Bildzusammensetzungsoperatoren sind.
6. Verfahren nach Anspruch 1, wobei die zweite Region durch eine Regiondarstellung in der Form einer hierarchischen Datenstruktur dargestellt wird.
7. Verfahren nach Anspruch 6, wobei die hierarchische Datenstruktur eine Quadtree-Darstellung ist.
8. Verfahren nach Anspruch 1, ferner mit dem Schritt des Durchlaufens (**904**) des Ausdrucksbaums Knoten für Knoten und der Durchführung der Vergleichs-, Bestimmungs- und Modifizierungsschritte bei zumindest einem durchlaufenen Knoten.
9. Verfahren nach Anspruch 8, wobei der Modifizierungsschritt (**912**) die Entfernung (**1008**) des aktuellen Knotens oder das Ersetzen des aktuellen Knotens durch einen anderen Knoten des Ausdrucksbaums enthält.
10. Verfahren nach Anspruch 8, wobei der Modifizierungsschritt (**912**) ferner das Abschneiden (**1012**) der durch den aktuellen Knoten dargestellten ersten Region oder das Markieren für ein Abschneiden zu einem späteren Zeitpunkt enthält.

11. Verfahren nach Anspruch 8, wobei das Verfahren bei zumindest einem einen Grafikoperator darstellenden Knoten die Schritte umfasst

Erhalten (1110) einer ersten Regionsdarstellung von einem Elternknoten,
 Führen (1112) einer modifizierten ersten Regionsdarstellung zu einem ersten Operanden des Grafikoperators entsprechend einer ersten vorbestimmten Modifikationsregel für den Operator,
 Zurückgeben (1114) einer zweiten Regionsdarstellung von durch einen Unterbaum verborgenen Regionen, der mit dem ersten Operanden verbunden ist, zu dem Grafikoperator,
 Führen (1116) einer modifizierten zweiten Regionsdarstellung zu einem zweiten Operanden des Grafikoperators entsprechend einer zweiten vorbestimmten Modifikationsregel für den Operator,
 Zurückgeben (1118) einer dritten Regionsdarstellung von durch einen Unterbaum verborgenen Regionen, der mit dem zweiten Operanden verbunden ist, zu dem Grafikoperator, und
 Bestimmen (1120) einer zu dem Elternknoten zurückzugebenden endgültigen Regionsdarstellung entsprechend einer eingestellten Regel für den Grafikoperator, wobei die endgültige Regionsdarstellung im Wesentlichen eine Region darstellt, in der der aktuelle Knoten andere Knoten im Ausdrucksbaum verbergen kann.

12. Verfahren nach Anspruch 11, wobei die eingestellte Regel aus der folgenden Gruppe ausgewählt wird:

(a) ist der Grafikoperator ein Operator „over“ oder „plusC“, wird die zu dem Elternknoten zurückzugebende endgültige Regionsdarstellung aus einer Vereinigung der zweiten Regionsdarstellung und der dritten Regionsdarstellung bestimmt,
 (b) ist der Grafikoperator ein Operator „in“, wird die zum Elternknoten zurückzugebende endgültige Regionsdarstellung aus einer Schnittmenge der zweiten Regionsdarstellung und der dritten Regionsdarstellung bestimmt,
 (c) ist der Grafikoperator ein Operator „ratop“, ist die zum Elternknoten zurückzugebende endgültige Regionsdarstellung die zweite Regionsdarstellung,
 (d) ist der Grafikoperator ein Operator „out“, wird die zum Elternknoten zurückzugebende endgültige Regionsdarstellung aus einer Differenz der zweiten Regionsdarstellung und einer Regionsdarstellung mit zumindest einer Region bestimmt, die durch eine Bounding-Box eines Knotens in einem rechten Unterbaum des aktuellen Knotens dargestellt wird, und
 (e) ist der Grafikoperator ein Operator „Xor“ oder „plusW“, wird die zum Elternknoten zurückzugebende endgültige Regionsdarstellung aus einer Vereinigung der zweiten Regionsdarstellung weniger einer Regionsdarstellung mit zumindest einer Region, die durch eine Bounding-Box eines Knotens in einem rechten Unterbaum des aktuellen Knotens dargestellt wird, und der dritten Regionsdarstellung weniger einer Regionsdarstellung mit einer Bounding-Box eines Knotens in einem rechten Unterbaum des aktuellen Knotens bestimmt.

13. Verfahren nach Anspruch 11, wobei die erste vorbestimmte Modifikationsregel umfasst

Übergeben im Wesentlichen der ersten Regionsdarstellung als modifizierte erste Regionsdarstellung, wenn der Grafikoperator ein Operator „over“, „in“, „ratop“, „plusC“, „plusW“, „Xor“, „out (linken Operanden zuerst betrachten)“ oder dergleichen ist, und
 wenn der Grafikoperator ein Operator „out (rechten Operanden zuerst betrachten)“ ist, Übergeben einer Vereinigung der ersten Regionsdarstellung mit der zweiten Regionsdarstellung als modifizierte erste Regionsdarstellung.

14. Verfahren nach Anspruch 11, wobei die zweite vorbestimmte Modifikationsregel umfasst

Übergeben im Wesentlichen der ersten Regionsdarstellung als modifizierte zweite Regionsdarstellung, wenn der Grafikoperator ein Operator „in“, „ratop“, „out“, „plusC“, „plusW“, „Xor“ oder dergleichen ist, und
 wenn der Grafikoperator ein Operator „over“ ist, Übergeben einer Vereinigung der ersten Regionsdarstellung mit der zweiten Regionsdarstellung als modifizierte zweite Regionsdarstellung.

15. Verfahren nach einem der Ansprüche 11 bis 14, wobei die Bilddarstellung nicht an einem Knoten erzeugt oder zu einem Elternknoten des Knotens zurückgegeben wird, wenn die Bilddarstellung nicht nachfolgend verwendet wird.

16. Verfahren nach Anspruch 15, wobei die Bilddarstellung an einem Knoten nicht erzeugt oder zu dem Elternknoten nicht zurückgegeben wird, wenn der Knoten aus der folgenden Gruppe ausgewählt wird:

rechter Operand eines Operators „over“, und der „over“-Operatorknoten muss keine Bilddarstellung zu seinem Elternknoten zurückgeben,
 linker Operand eines Operators „in“, „plusC“, „plusW“ oder „Xor“, und der Operatorknoten muss keine Bilddarstellung zu seinem Elternknoten zurückgeben,
 rechter Operand eines Operators „in“, „plusC“, „plusW“ oder „Xor“, und der Operatorknoten muss keine Bilddarstellung zu seinem Elternknoten zurückgeben,
 linker Operand eines Operators „out“ oder „ratop“, und der Knoten muss keine Bilddarstellung zu seinem El-

ternknoten zurückgeben,
 rechter Operand eines Operators „ratop“,
 Wurzel des Ausdrucksbaums,
 Operand eines Bildwickel-, Affintransformations- oder Faltungsoperators,
 Operand einer Farbtransformation, die die Undurchsichtigkeit nicht bewahrt, oder wenn der Transformationsknoten keine Bilddarstellung zu seinem Elternknoten zurückgeben muss.

17. Vorrichtung zur Optimierung eines Ausdrucksbaums (**101**, **902**, **1102**), der zum Zusammensetzen eines Bildes dient und zumindest drei Knoten umfasst, wobei jeder Knoten des Baums zumindest entweder ein Bildregion definierendes Grafikelement oder ein Grafikoperator ist, mit einer Einrichtung zum Vergleichen (**908**) einer ersten Region des Knotens mit einer zweiten Region, die von zumindest einem anderen Knoten irgendwo im Ausdrucksbaum erhalten wird, einer Einrichtung zum Bestimmen (**910**), ob die erste Region durch die zweite Region ganz oder teilweise verborgen ist, und einer Einrichtung zum Modifizieren (**912**) des Ausdrucksbaums, wenn die erste Region zumindest teilweise oder ganz durch die zweite Region verborgen ist, um einen optimierten Ausdrucksbaum zum Zusammensetzen des gleichen Bildes auszubilden, wobei ein optimierter Teil des Ausdrucksbaums im Wesentlichen nicht-verborgene Abschnitte der ersten Region darstellt.

18. Vorrichtung nach Anspruch 17, wobei die Modifizierungseinrichtung eine Einrichtung zur Anwendung (**1012**) eines Abschneideoperators bei dem Knoten beinhaltet, wenn die erste Region teilweise verborgen ist.

19. Vorrichtung nach Anspruch 17, wobei die Modifizierungseinrichtung umfasst:
 eine Einrichtung zum Entfernen (**1008**) des Knotens und,
 eine Einrichtung zum Auswählen (**1010**) einer Knotenersetzungsregel aus einem vorbestimmten Satz von Knotenersetzungsregeln entsprechend dem Grafikoperator und Anwenden (**1010**) der Regel, wenn der Knoten einen Elternknoten mit einem Grafikoperator hat und der Knoten ganz verborgen ist.

20. Vorrichtung nach Anspruch 19, wobei der vorbestimmte Satz von Knotenersetzungsregeln zumindest einen aus der folgenden Gruppe ausgewählten Schritt umfasst:
 wenn der Elternknoten einen Grafikoperator „over“ darstellt, und der aktuelle Knoten sich in einem linken Zweig des Elternknotens befindet, Ersetzen des Elternknotens durch einen rechten Unterbaum des Elternknotens,
 wenn der Elternknoten einen Grafikoperator „over“ darstellt, und der aktuelle Knoten sich in einem rechten Zweig des Elternknotens befindet, Ersetzen des Elternknotens durch einen linken Unterbaum des Elternknotens,
 wenn der Elternknoten einen Grafikoperator „in“ darstellt, Entfernen des Elternknotens und jedes Unterbaums, der vom Elternknoten verzweigt,
 wenn der Elternknoten einen Grafikoperator „ratop“ darstellt, und der aktuelle Knoten sich in einem linken Zweig des Elternknotens befindet, Entfernen des Elternknotens und jedes Unterbaums, der vom Elternknoten verzweigt,
 wenn der Elternknoten einen Grafikoperator „ratop“ darstellt, und der aktuelle Knoten sich in einem rechten Zweig des Elternknotens befindet, Ersetzen des Elternknotens durch einen linken Unterbaum des Elternknotens,
 wenn der Elternknoten einen Grafikoperator „out“ darstellt, und der aktuelle Knoten sich in einem linken Zweig des zweiten Knotens befindet, Entfernen des Elternknotens und jedes Unterbaums, der vom Elternknoten verzweigt,
 wenn der Elternknoten einen Grafikoperator „out“ darstellt, und der aktuelle Knoten sich in einem rechten Zweig des Elternknotens befindet, Ersetzen des Elternknotens durch einen linken Unterbaum des Elternknotens,
 wenn der Elternknoten einen Grafikoperator „plusC“ darstellt, und der aktuelle Knoten sich in einem linken Zweig des Elternknotens befindet, Ersetzen des Elternknotens durch einen rechten Unterbaum des Elternknotens,
 wenn der Elternknoten einen Grafikoperator „plusC“ darstellt, und sich der aktuelle Knoten in einem rechten Zweig des Elternknotens befindet, Ersetzen des Elternknotens durch einen linken Unterbaum des Elternknotens,
 wenn der Elternknoten einen Grafikoperator „plusW“ oder „Xor“ darstellt, und sich der aktuelle Knoten in einem linken Zweig des Elternknotens befindet, Ersetzen des Elternknotens durch einen rechten Unterbaum des Elternknotens, und
 wenn der Elternknoten einen Grafikoperator „plusW“ oder „Xor“ darstellt, und sich der aktuelle Knoten in einem rechten Zweig des Elternknotens befindet, Ersetzen des Elternknotens durch einen linken Unterbaum des El-

ternknotens.

21. Vorrichtung nach einem der Ansprüche 17 bis 20, wobei die Grafikoperatoren Bildzusammensetzungsoperatoren sind.

22. Vorrichtung nach Anspruch 17, wobei die zweite Region durch eine Regiondarstellung in der Form einer hierarchischen Datenstruktur dargestellt ist.

23. Vorrichtung nach Anspruch 22, wobei die hierarchische Datenstruktur eine Quadtree-Darstellung ist.

24. Vorrichtung nach Anspruch 17, ferner mit einer Einrichtung zum Durchlaufen (**904**) des Ausdrucksbaums Knoten für Knoten, wobei die Vergleichs-, Bestimmungs- und Modifizierungseinrichtungen zum Arbeiten bei zumindest einem durchlaufenen Knoten eingerichtet sind.

25. Vorrichtung nach Anspruch 24, wobei die Modifizierungseinrichtung eine Einrichtung zur Entfernung (**1008**) des aktuellen Knotens oder zum Ersetzen des aktuellen Knotens durch einen anderen Knoten des Ausdrucksbaums enthält.

26. Vorrichtung nach Anspruch 24, wobei die Modifizierungseinrichtung ferner eine Einrichtung zum Abschneiden (**1012**) der durch den aktuellen Knoten dargestellten Region oder Markieren für ein Abschneiden zu einem späteren Zeitpunkt enthält.

27. Vorrichtung nach Anspruch 24, wobei die Vorrichtung für zumindest einen einen Grafikoperator darstellenden Knoten umfasst

eine Einrichtung zum Erhalten (**1110**) einer ersten Regionsdarstellung von einem Elternknoten, eine Einrichtung zum Führen (**1112**) einer modifizierten ersten Regionsdarstellung zu einem ersten Operanden des Grafikoperators entsprechend einer ersten vorbestimmten Modifikationsregel für den Operator, eine Einrichtung zum Zurückgeben (**1114**) einer zweiten Regionsdarstellung von durch einen Unterbaum verborgenen Regionen, der mit dem ersten Operanden verbunden ist, zu dem Grafikoperator, eine Einrichtung zum Führen (**1116**) einer modifizierten zweiten Regionsdarstellung zu einem zweiten Operanden des Grafikoperators entsprechend einer zweiten vorbestimmten Modifikationsregel für den Operator, eine Einrichtung zum Zurückgeben (**1118**) einer dritten Regionsdarstellung von durch einen Unterbaum verborgenen Regionen, der mit dem zweiten Operanden verbunden ist, zu dem Grafikoperator, und eine Einrichtung zum Bestimmen (**1120**) einer zu dem Elternknoten zurückzugebenden endgültigen Regionsdarstellung entsprechend einer eingestellten Regel für den Grafikoperator, wobei die endgültige Regionsdarstellung im Wesentlichen eine Region darstellt, in der der aktuelle Knoten andere Knoten im Ausdrucksbaum verbergen kann.

28. Vorrichtung nach Anspruch 27, wobei die eingestellte Regel aus der folgenden Gruppe ausgewählt wird:

(a) ist der Grafikoperator ein Operator „over“ oder „plusC“, wird die zu dem Elternknoten zurückzugebende endgültige Regionsdarstellung aus einer Vereinigung der zweiten Regionsdarstellung und der dritten Regionsdarstellung bestimmt,

(b) ist der Grafikoperator ein Operator „in“, wird die zum Elternknoten zurückzugebende endgültige Regionsdarstellung aus einer Schnittmenge der zweiten Regionsdarstellung und der dritten Regionsdarstellung bestimmt,

(c) ist der Grafikoperator ein Operator „ratop“, ist die zum Elternknoten zurückzugebende endgültige Regionsdarstellung die zweite Regionsdarstellung,

(d) ist der Grafikoperator ein Operator „out“, wird die zum Elternknoten zurückzugebende endgültige Regionsdarstellung aus einer Differenz der zweiten Regionsdarstellung und einer Regionsdarstellung mit zumindest einer Region bestimmt, die durch eine Bounding-Box eines Knotens in einem rechten Unterbaum des aktuellen Knotens dargestellt wird, und

(e) ist der Grafikoperator ein Operator „Xor“ oder „plusW“, wird die zum Elternknoten zurückzugebende endgültige Regionsdarstellung aus einer Vereinigung der zweiten Regionsdarstellung weniger einer Regionsdarstellung mit zumindest einer Region, die durch eine Bounding-Box eines Knotens in einem rechten Unterbaum des aktuellen Knotens dargestellt wird, und der dritten Regionsdarstellung weniger einer Regionsdarstellung mit einer Bounding-Box eines Knotens in einem rechten Unterbaum des aktuellen Knotens bestimmt.

29. Vorrichtung nach Anspruch 27, wobei die erste vorbestimmte Modifikationsregel umfasst Übergeben im Wesentlichen der ersten Regionsdarstellung als modifizierte erste Regionsdarstellung, wenn der Grafikoperator ein Operator „over“, „in“, „ratop“, „plusC“, „plusW“, „Xor“, „out“ (linken Operanden zuerst be-

trachten)" oder dergleichen ist, und wenn der Grafikoperator ein Operator „out (rechten Operanden zuerst betrachten)" ist, Übergeben einer Vereinigung der ersten Regionsdarstellung mit der zweiten Regionsdarstellung als modifizierte erste Regionsdarstellung.

30. Vorrichtung nach Anspruch 27, wobei die zweite vorbestimmte Modifikationsregel umfasst Übergeben im Wesentlichen der ersten Regionsdarstellung als modifizierte zweite Regionsdarstellung, wenn der Grafikoperator ein Operator „in", „ratop", „out", „plusC", „plusW", „Xor" oder dergleichen ist, und wenn der Grafikoperator ein Operator „over" ist, Übergeben einer Vereinigung der ersten Regionsdarstellung mit der zweiten Regionsdarstellung als modifizierte zweite Regionsdarstellung.

31. Vorrichtung nach einem der Ansprüche 27 bis 30, wobei die Bilddarstellung nicht an einem Knoten erzeugt oder zu einem Elternknoten des Knotens zurückgegeben wird, wenn die Bilddarstellung nicht nachfolgend verwendet wird.

32. Vorrichtung nach Anspruch 31, wobei die Bilddarstellung an einem Knoten nicht erzeugt oder zu dem Elternknoten nicht zurückgegeben wird, wenn der Knoten aus der folgenden Gruppe ausgewählt wird: rechter Operand eines Operators „over", und der „over" -Operatorknoten muss keine Bilddarstellung zu seinem Elternknoten zurückgeben, linker Operand eines Operators „in", „plusC", „plusW" oder „Xor", und der Operatorknoten muss keine Bilddarstellung zu seinem Elternknoten zurückgeben, rechter Operand eines Operators „in", „plusC", „plusW" oder „Xor", und der Operatorknoten muss keine Bilddarstellung zu seinem Elternknoten zurückgeben, linker Operand eines Operators „out" oder „ratop", und der Knoten muss keine Bilddarstellung zu seinem Elternknoten zurückgeben, rechter Operand eines Operators „ratop", Wurzel des Ausdrucksbaums, Operand eines Bildwickel-, Affintransformations- oder Faltungsoperators, Operand einer Farbtransformation, die die Undurchsichtigkeit nicht bewahrt, oder wenn der Transformationsknoten keine Bilddarstellung zu seinem Elternknoten zurückgeben muss.

33. Bildverarbeitungsverfahren mit den Schritten Erzeugen eines Bildausdrucksbaums (**101, 902, 1102**) für ein Bild (**40**), Optimieren des Ausdruckbaums unter Verwendung eines Verfahrens nach einem der Ansprüche 1 bis 16, Zusammensetzen des Bildes unter Verwendung des optimierten Ausdruckbaums und Ausgeben des zusammengesetzten Bildes.

34. Bildverarbeitungsvorrichtung mit einer Einrichtung zur Erzeugung eines Bildausdruckbaums (**101, 902, 1102**) für ein Bild (**40**); einer Vorrichtung zur Optimierung des Ausdruckbaums nach einem der Ansprüche 17 bis 32, einer Einrichtung zur Zusammensetzung des Bildes unter Verwendung des optimierten Ausdruckbaums und einer Ausgabereinrichtung zur Ausgabe des zusammengesetzten Bildes.

Es folgen 10 Blatt Zeichnungen

Anhängende Zeichnungen

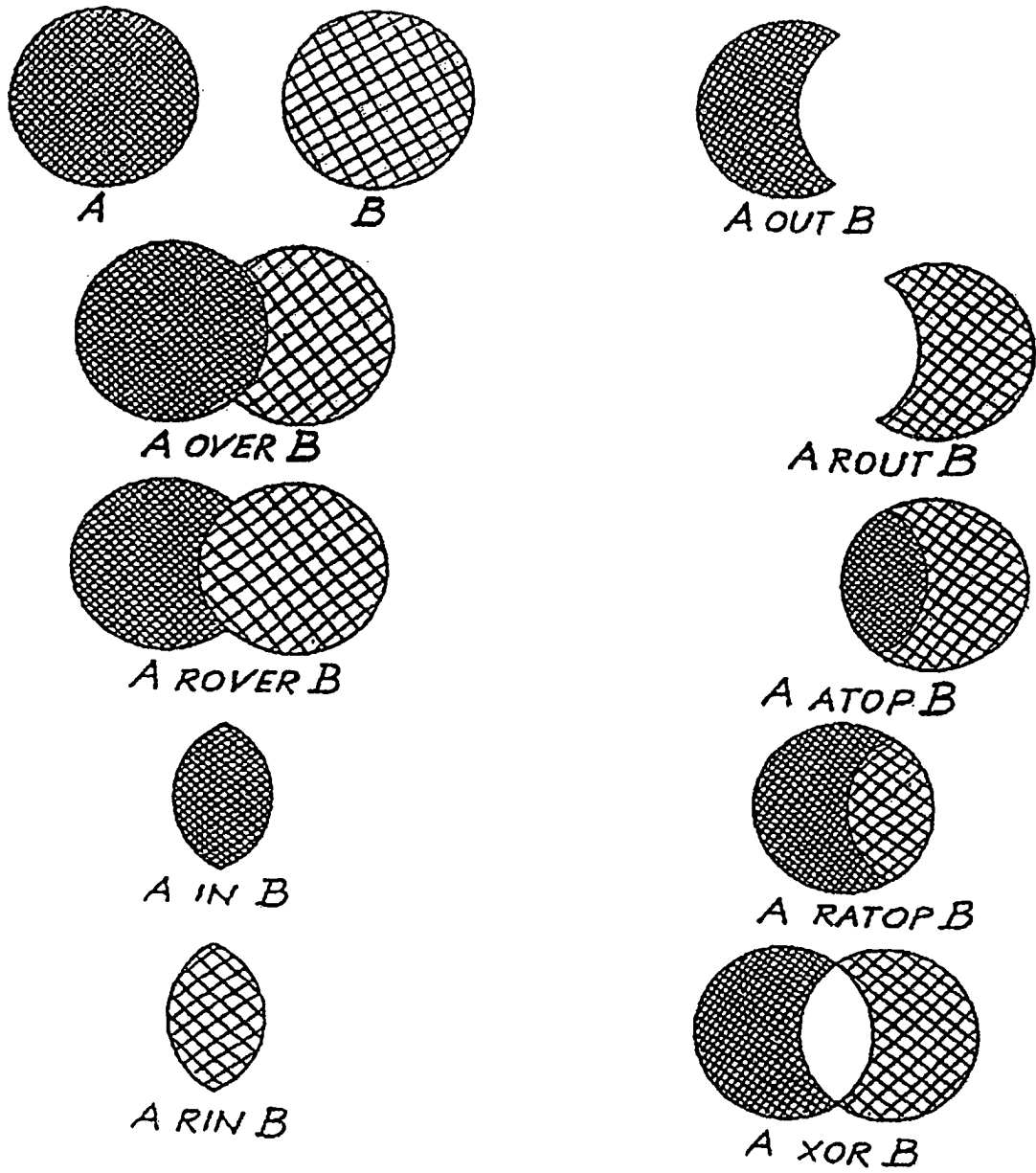


Fig. 1

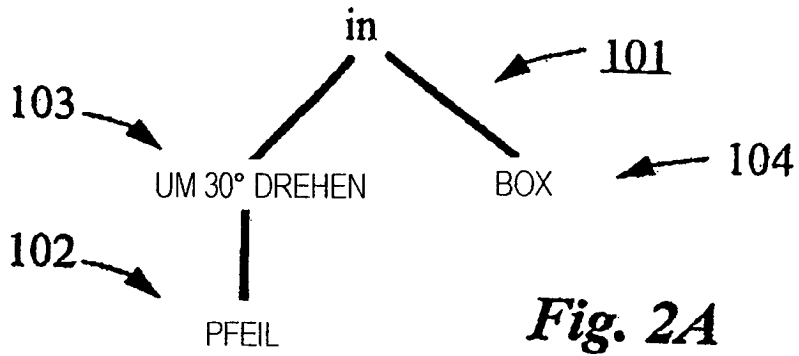


Fig. 2A

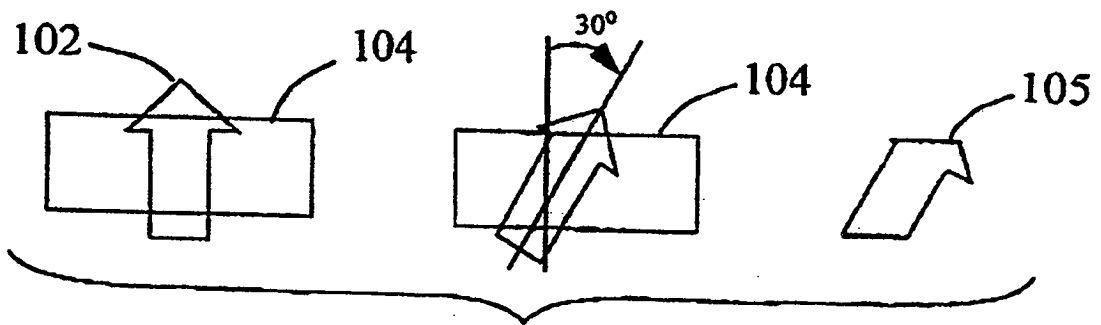


Fig. 2B

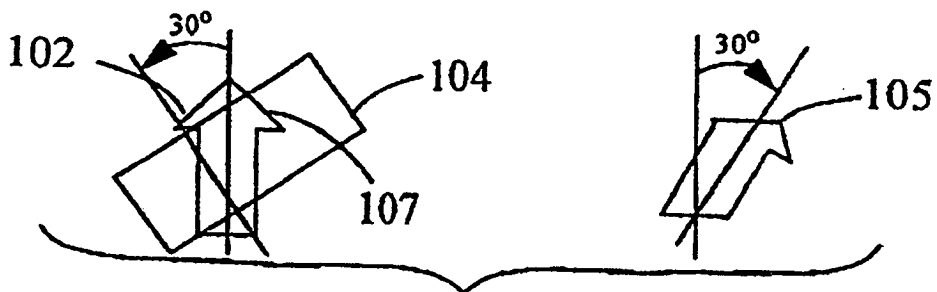


Fig. 2C

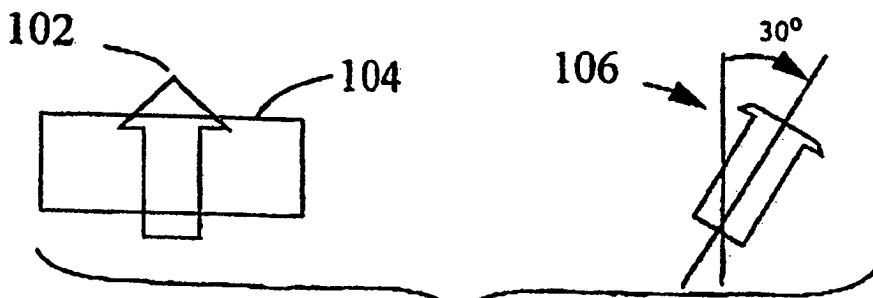


Fig. 2D

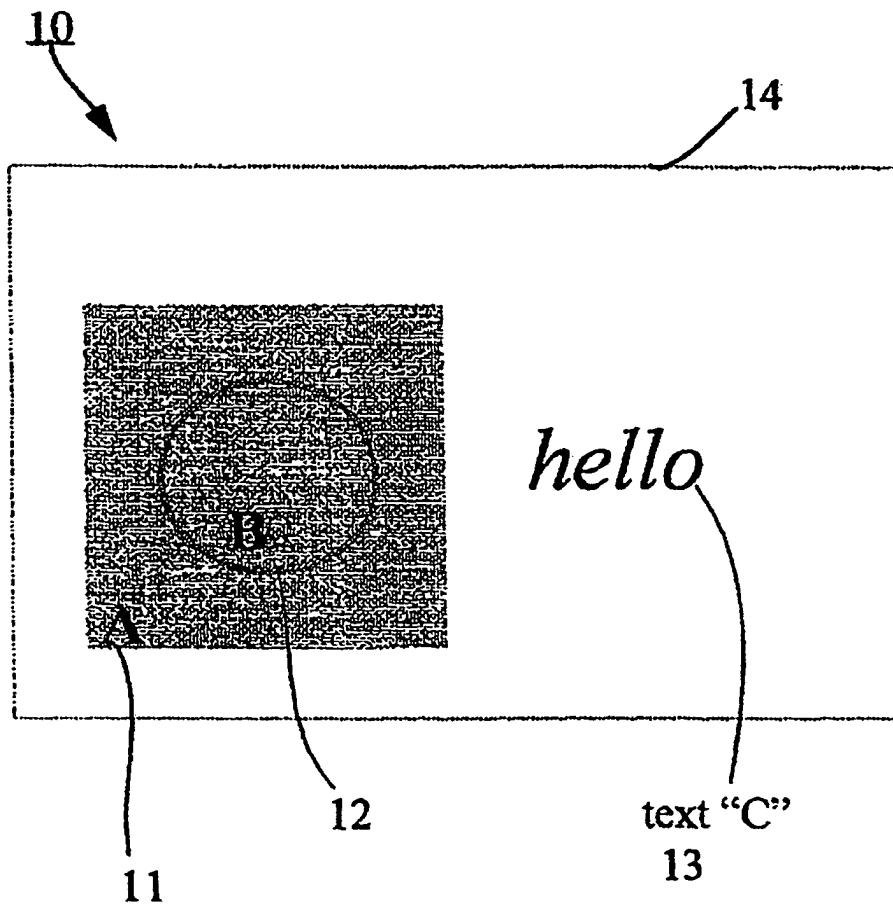


Fig. 3

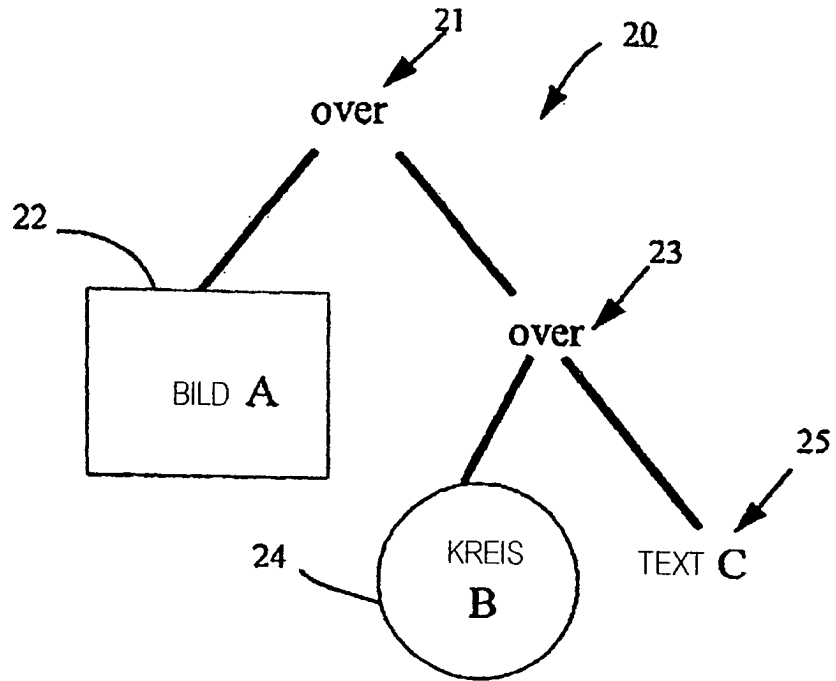


Fig. 4

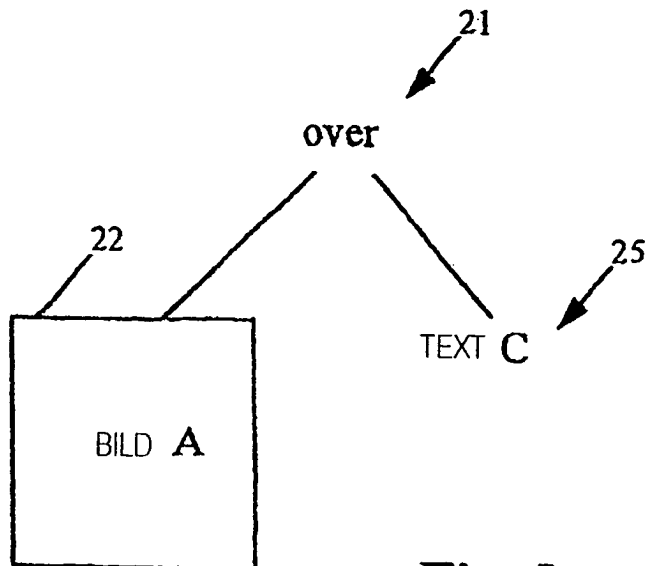


Fig. 5

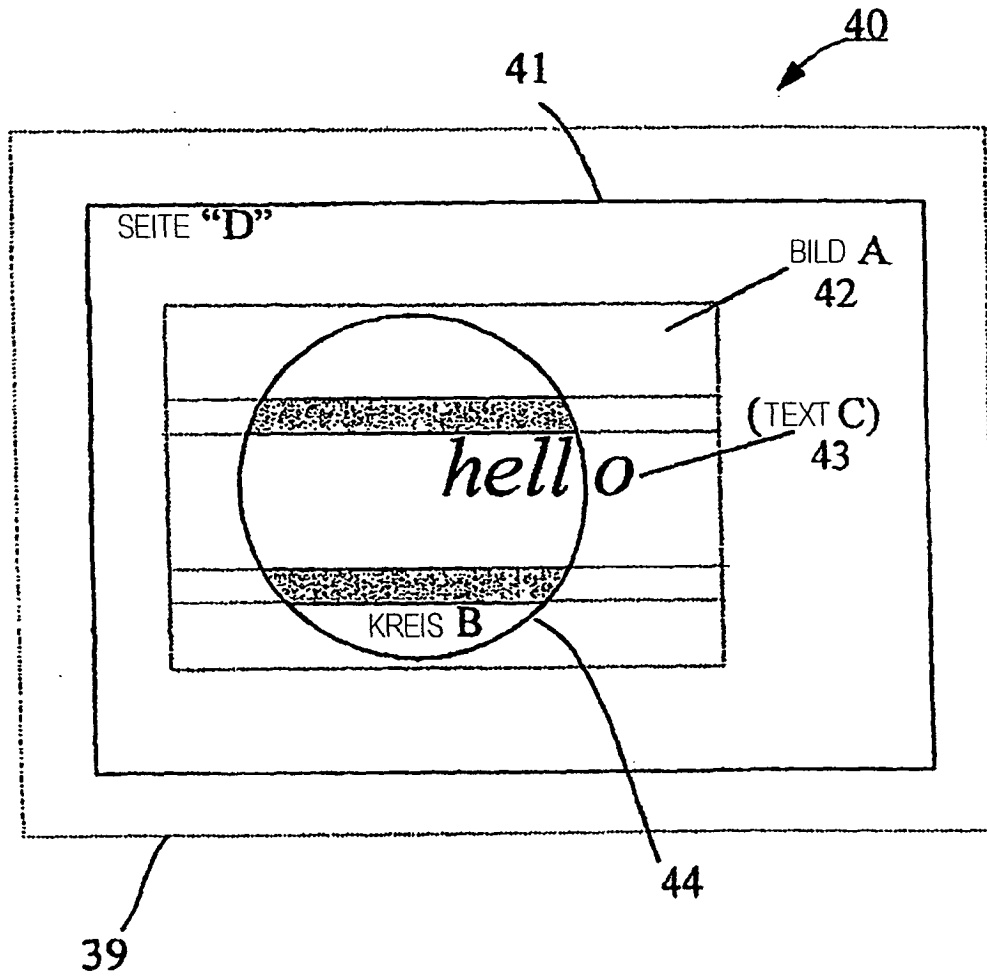


Fig. 6

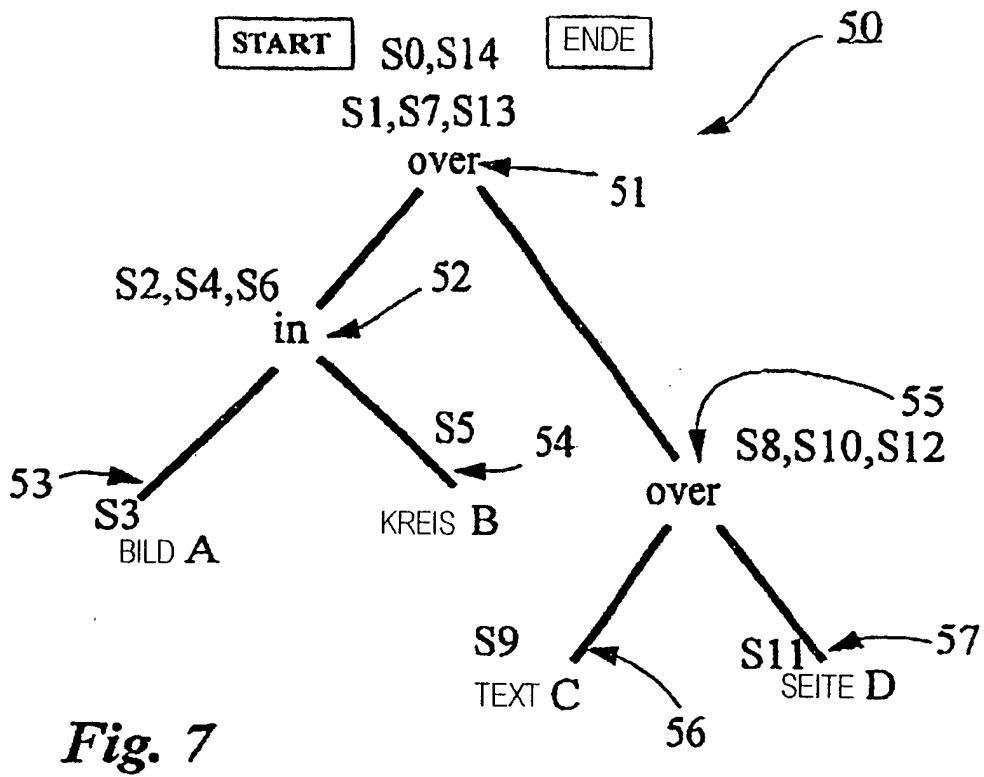


Fig. 7

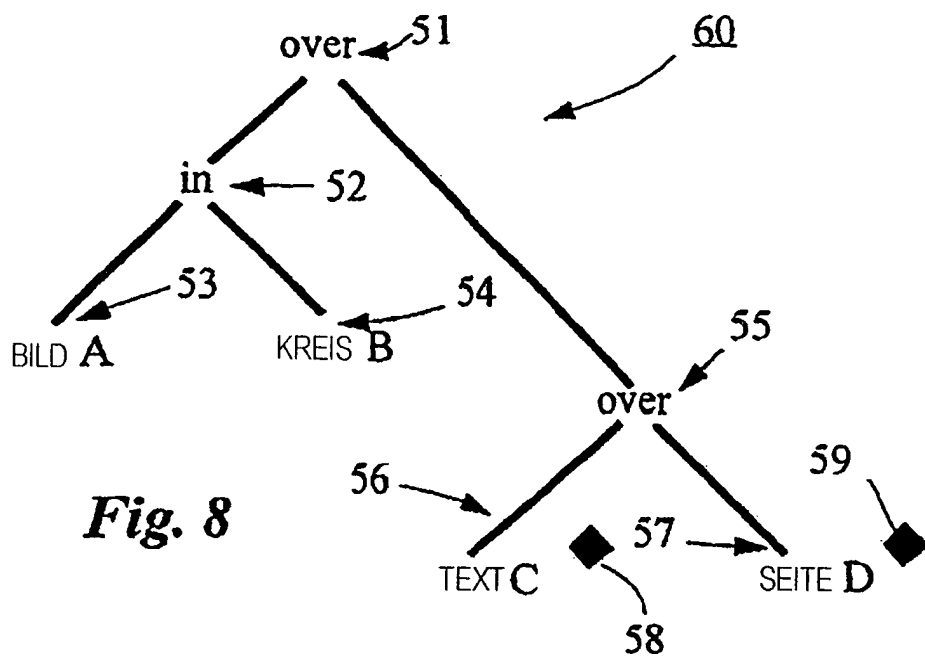


Fig. 8

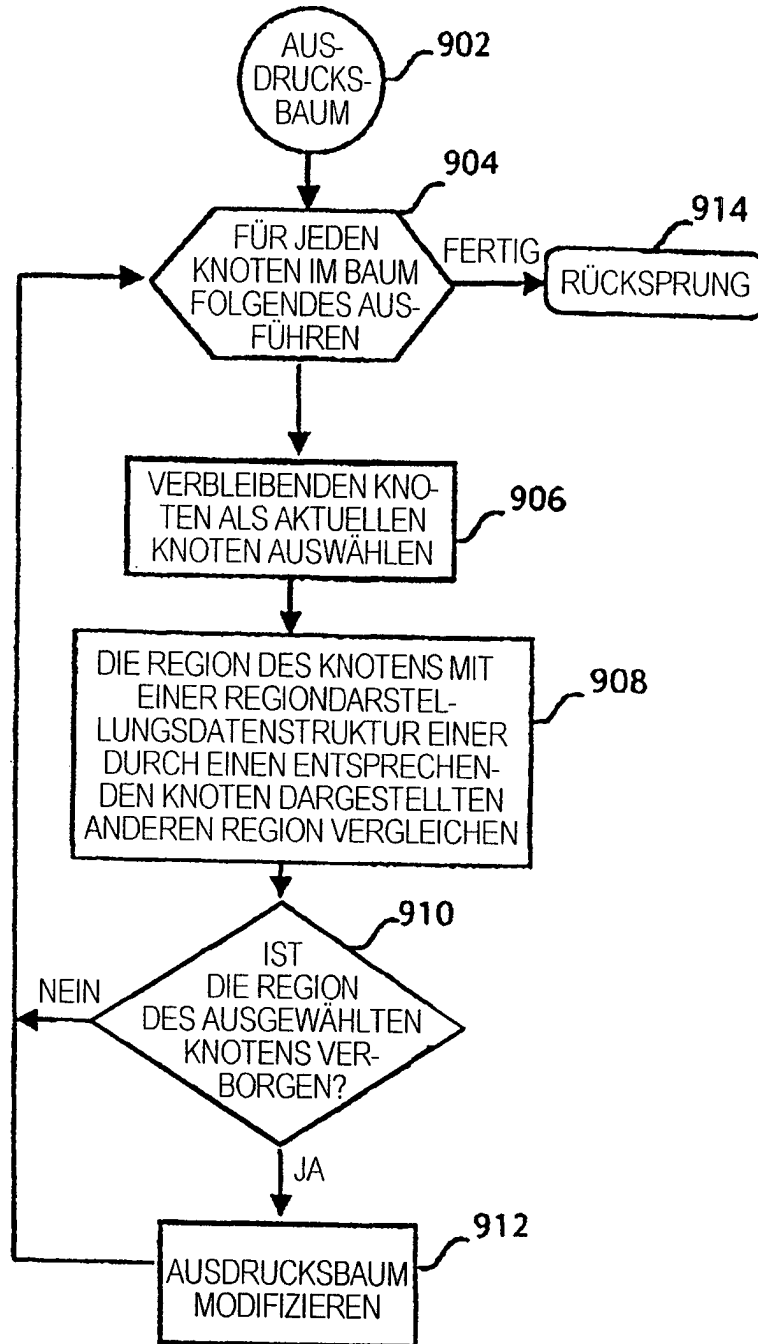


FIG. 9

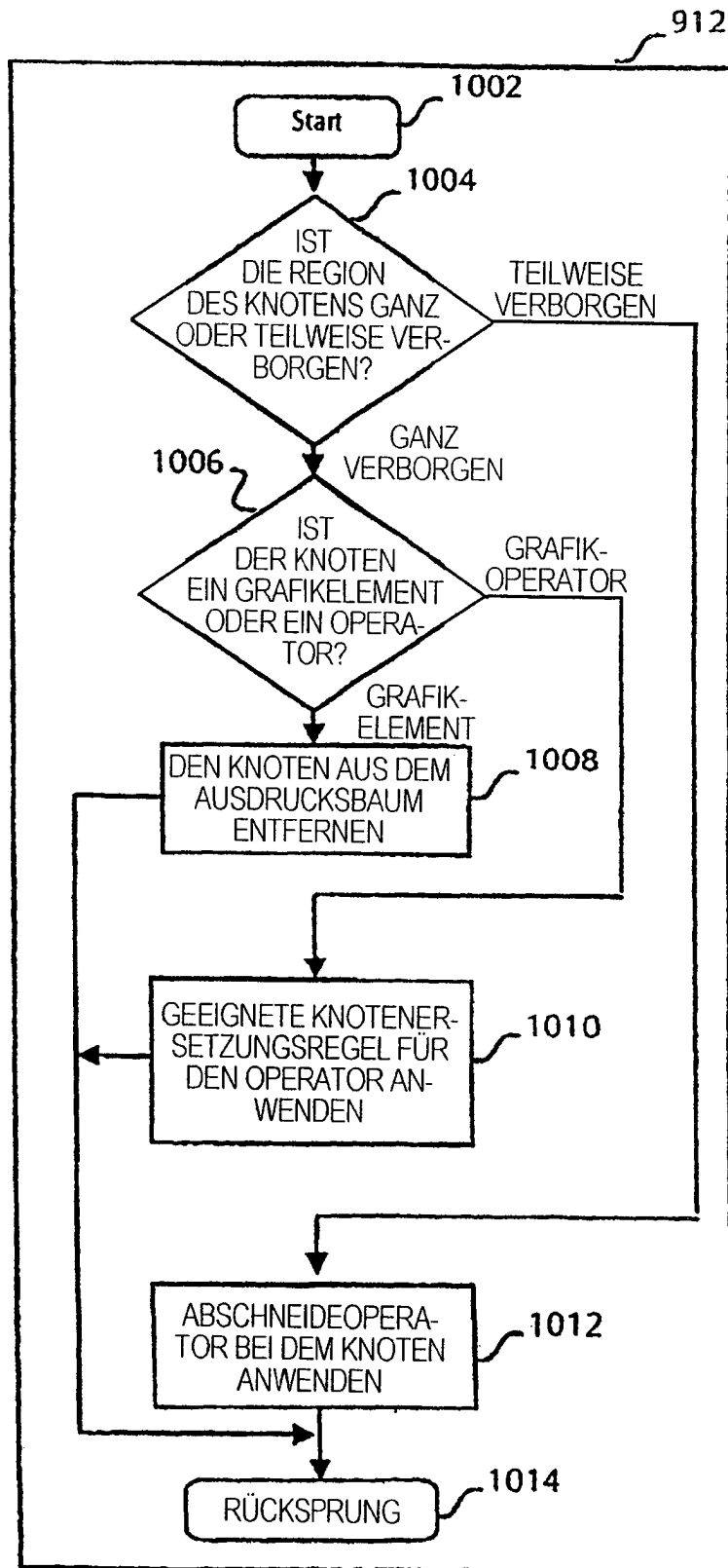


FIG. 10

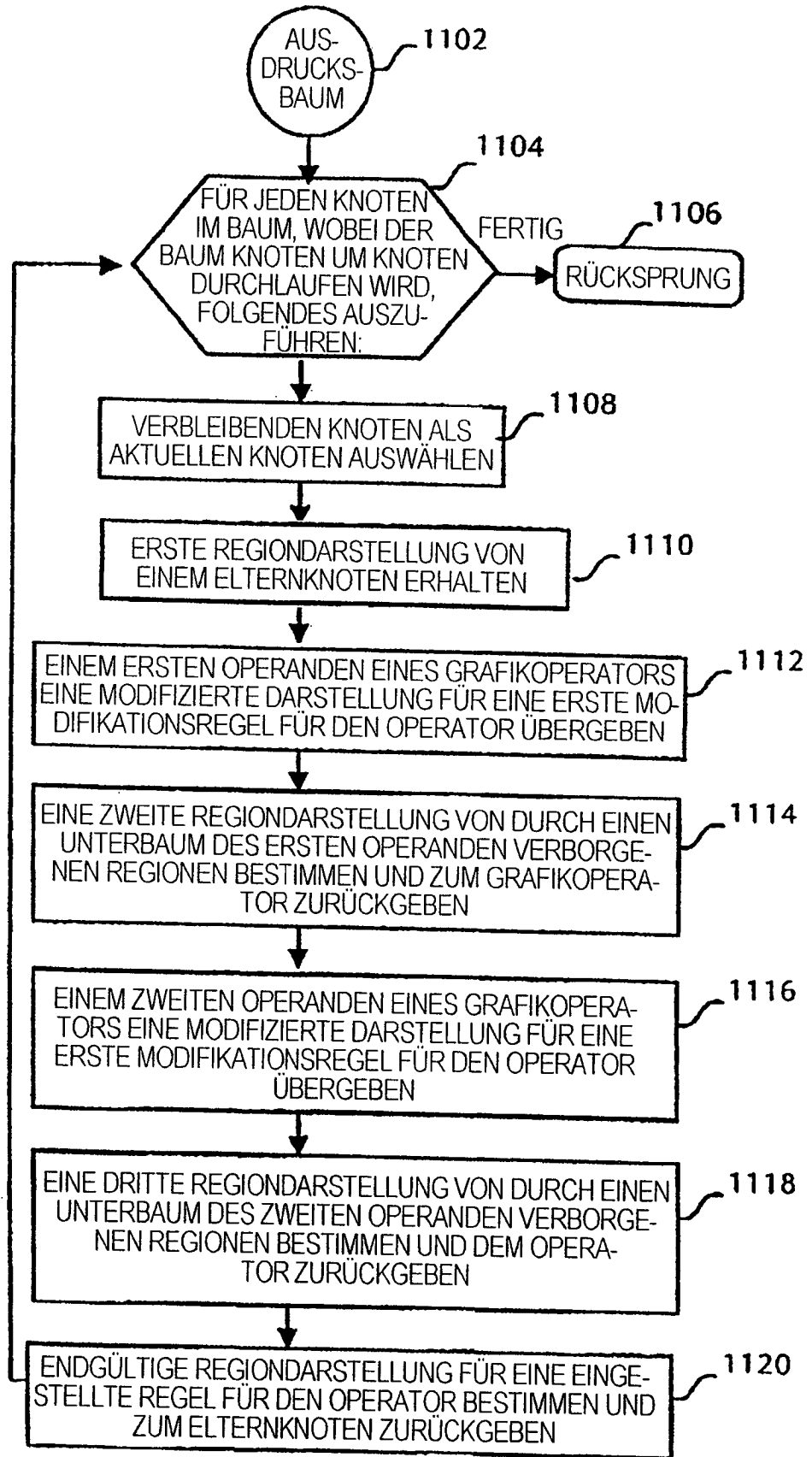


FIG. 11

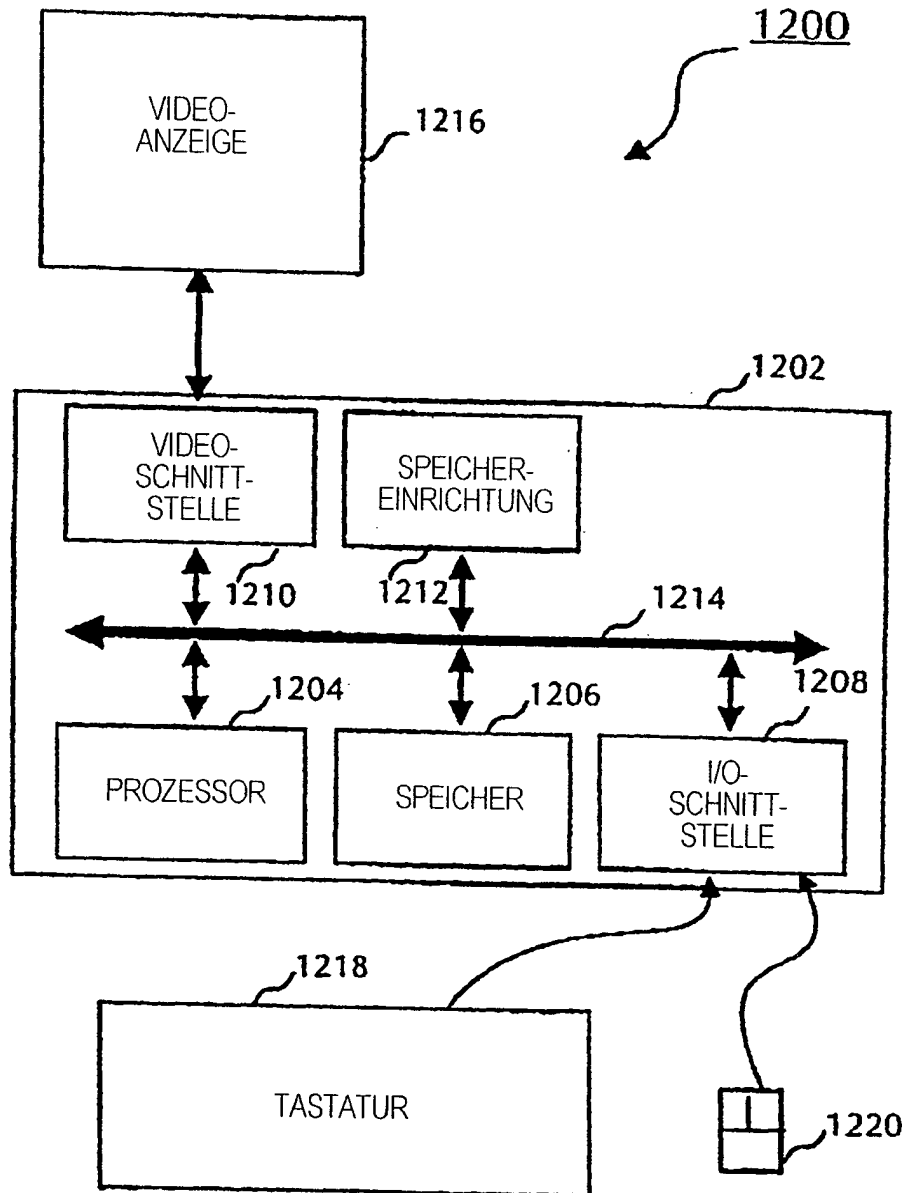


FIG. 12