



(19)  
Bundesrepublik Deutschland  
Deutsches Patent- und Markenamt

(10) **DE 698 28 994 T2** 2006.03.30

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 031 127 B1**

(21) Deutsches Aktenzeichen: **698 28 994.3**

(86) PCT-Aktenzeichen: **PCT/US98/24633**

(96) Europäisches Aktenzeichen: **98 959 501.2**

(87) PCT-Veröffentlichungs-Nr.: **WO 99/027517**

(86) PCT-Anmeldetag: **18.11.1998**

(87) Veröffentlichungstag

der PCT-Anmeldung: **03.06.1999**

(97) Erstveröffentlichung durch das EPA: **30.08.2000**

(97) Veröffentlichungstag

der Patenterteilung beim EPA: **09.02.2005**

(47) Veröffentlichungstag im Patentblatt: **30.03.2006**

(51) Int Cl.<sup>8</sup>: **G09G 1/16** (2006.01)  
**G06F 9/44** (2006.01)

(30) Unionspriorität:

**975268**            **21.11.1997**    **US**

**88478 P**           **05.06.1998**    **US**

**191322**           **13.11.1998**    **US**

(73) Patentinhaber:

**xSides Corporation, Bellevue, WA, US**

(74) Vertreter:

**Grünecker, Kinkeldey, Stockmair &  
Schwanhäusser, 80538 München**

(84) Benannte Vertragsstaaten:

**AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT,  
LI, LU, MC, NL, PT, SE**

(72) Erfinder:

**NASON, D., David, Seattle, US; O'ROURKE, C.,  
Thomas, Seattle, US; CAMPBELL, J., Scott,  
Seattle, US**

(54) Bezeichnung: **SEKUNDÄRBENUTZERSCHNITTSTELLE**

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**

## HINTERGRUND DER ERFINDUNG

## 1. Erfindungsgebiet

**[0001]** Die vorliegende Erfindung betrifft Computer-Benutzerschnittstellenanzeigen und insbesondere den Gebrauch einer Benutzerschnittstelle getrennt von der Standard-Benutzerschnittstellenanzeige.

## 2. Beschreibung des Standes der Technik

**[0002]** Es gab eine Zeit als das populärste Betriebssystem für Personal Computer (DOS) über keine grafische Benutzerschnittstelle verfügte. Jedes Unternehmen konnte ein "Menü" oder "Gerüst" schaffen, das das erste nach dem Anlaufen des Computers gestartete Programm sein würde und dem Benutzer Optionen zum Starten und Verwalten verschiedener Anwendungen vorlegen würde. Obwohl Grafik-Programmierung in der DOS-Umgebung schwierig war, erzeugten einige Firmen sogar grafische Benutzerschnittstellen, die dann andere Programme starten konnten.

**[0003]** Microsoft Corporation, Redmond, Washington, führte eine solche grafische Benutzerschnittstelle zum Starten von Anwendungen ein, die "Windows" genannt wird. Die ersten drei Versionen von Windows waren einfach nur Anwendungen, die unter DOS liefen und eines von zahlreichen Elementen sein konnten, das aus einem vorher laufenden Menü oder Gerüst auszuwählen ist und von einer anderen Firma als Microsoft angeboten werden konnte. Dies erlaubte anderen Firmen, Benutzern Primärbenutzer-Schnittstellenprogramme anzubieten, ohne dass der Benutzer durch eine Microsoft-kontrollierte Benutzerschnittstelle geht.

**[0004]** Mit der Einführung von Windows™ durch Microsoft präsentiert das anfängliche Laden des Betriebssystems von Anfang an eine von Microsoft entwickelte grafische Benutzerschnittstelle, die die ganze Bildschirmanzeige belegt. Wie bei ihren früheren Betriebssystem-Produkten arrangierte sich Microsoft mit Herstellern von Standard-Computerhardware, dieses Betriebssystem mit jedem verkauften Computer einzuschließen. Mit der Beherrschung dieses Marktes durch Microsoft war es für andere Software-Lieferanten unmöglich geworden, Benutzern eine Schnittstelle anders als ein Microsoft-Symbol in dem Microsoft-"Desktop", der aus der ganzen Bildschirmanzeige besteht, zu präsentieren. Dies führte zu einer Notwendigkeit nach Zugang zu einer Benutzerschnittstelle, die außerhalb der Standard-Computer-Bildschirmanzeige und daher unabhängig von den Diktaten von Microsoft für Elemente in ihrem "Desktop" präsentiert werden konnte.

**[0005]** Standard Personal Computer verwenden VGA oder Super VGA oder XGA Videoanzeigesysteme. Diese Anzeigesysteme arbeiten in standardisierten Grafik-Modi wie 640 × 480 Pixel, 800 × 600 Pixel, 1024 × 768 Pixel und 1280 × 1024 Pixel. Wenn einer dieser Anzeigemodi ausgewählt wird, ist dies die ganze zur Anzeige verfügbare Fläche. In der Microsoft Windows-Umgebung weist der Benutzer das Windows-Betriebssystem an, einen dieser Standard-Anzeigemodi auszuwählen, und das Windows-Betriebssystem präsentiert dann alle Anwendungen und ihre Icons in dem ausgewählten Anzeigebereich. Zur Zeit gibt es keine Möglichkeit, den Windows-"Desktop" zu veranlassen, weniger als den ganzen Anzeigebereich zu verwenden und dennoch wie gedacht zu funktionieren und einem anderen Programm von einem anderen Lieferanten zu erlauben, den Rest zu kontrollieren. Was benötigt wird, ist die Fähigkeit, behindernden Videospeicher aus dem Weg zu räumen und sicherzustellen, dass nicht anderes, was behindernd sein könnte, später diesem Raum zugeteilt wird.

## ZUSAMMENFASSUNG DER ERFINDUNG

**[0006]** Die Erfindung ist ein Verfahren, das bereitgestellt wird, um der grafischen Standard-Benutzeranzeigeschnittstelle eine neue Benutzerschnittstelle hinzuzufügen und diese zu verwenden, z.B. in dem Rand jenseits der Standard-Schirmanzeige-Fläche. Herkömmliche Videosysteme wie VGA, SVGA und XGA enthalten einen definierten Rand, der die Anzeigefläche umgibt. Der ursprüngliche Zweck dieses Randes war, dem horizontalen und vertikalen Rücklauf der Elektronenkanone in einer Kathodenstrahlröhrenanzeige eine angemessene Zeit einzuräumen. Mit dem Erscheinen von LCD-Displays und da die Recklaufgeschwindigkeiten in modernen Monitoren zugenommen haben, ist es aber jetzt möglich, eine Benutzer-Schnittstellenanzeige in diesem Rand zu präsentieren. Der Rand, der als eine Benutzerschnittstelle gesteuert werden kann, ist ein Abschnitt von dem, was als "Overscan" bekannt ist. Diese Erfindung ist ein Verfahren zum Präsentieren von einer oder mehreren zusätzlichen Sekundär-Benutzerschnittstellen, z.B. in dem Overscan-Bereich, der die herkömmliche Benutzer-Schnittstellenanzeige, oft Desktop genannt, umgibt.

**[0007]** Wenn die Elektronenkanone in einer CRT zum linken oder oberen Rand des Bildschirms zurückspringt, benötigt sie eine bedeutende Menge an Zeit im Vergleich zu der Präsentation einer abgetasteten Datenzeile. Während des Rücklaufs wird die Elektronenkanone abgeschaltet ("ausgetastet"). Wenn die für den Rücklauf benötigte Austastzeit gleich dem Betrag an verfügbarer Zeit ist, gibt es keinen brauchbaren Overscan. Die Rücklaufgeschwindigkeiten moderner Monitore sind jedoch viel schneller geworden, sodass eine bedeutende Menge an Zeit übrig ist, wenn die Elektronenkanone nicht ausgetastet werden muss, was einen anzeigefähigen Rand gestattet. Obwohl beim Stand der Technik der Rand gewöhnlich "schwarz" ist (die Kanone ist ausgeschaltet), ist wohl bekannt, zu spezifizieren, dass dem Rand irgendeine von sechs Farben gegeben werden soll. Standard BIOS erlaubt eine Spezifikation dieser Farbe. Die gewünschte Farbe wird einfach in einem der Register für den Video-Controller spezifiziert. Keine Daten für diese Farbe werden in dem Videospeicherpuffer für die Anzeige gespeichert. Diese Erfindung errichtet einen zusätzlichen Videopuffer für den Rand und erlaubt es, diesen Puffer mit Anzeigedaten wie der reguläre Anzeigepuffer zu beschreiben. Die Anzeigefläche wird dadurch an einer oder mehreren Kanten erweitert, um eine vorher unsichtbare, sichtbare Fläche bereitzustellen. Die Pixel in dieser neu sichtbaren Fläche der Anzeige werden für Programme durch eine Anwendungsprogrammierungs-Schnittstellen-(API)Komponente dieser Erfindung zugänglich gemacht. Ein Programm, das eine grafische Benutzerschnittstelle enthält, kann in der früher ausgetasteten Fläche der Anzeige angezeigt werden, was die zugängliche Fläche der Anzeige ohne Hardware-Modifikation funktionell vergrößert.

**[0008]** Die Erfindung umfasst ein Verfahren entsprechend dem in dem unabhängigen Anspruch 1 beanspruchten, einen Anzeige-Controller entsprechend dem in dem unabhängigen Anspruch 22 beanspruchten sowie ein computerlesbares Speichermedium entsprechend dem in dem unabhängigen Anspruch 44 beanspruchten.

**[0009]** In einer ersten Ausführung wird nur die vertikale Abmessung vergrößert, und die Overscan-Benutzerschnittstelle wird über oder unter der Primär-Anzeigefläche präsentiert. Alternativ kann die horizontale Abmessung vergrößert und die Overscan-Benutzerschnittstelle links oder rechts von der Primär-Anzeigefläche angezeigt werden. Desgleichen kann das Schnittstellenbild auf einer oder allen vier Seiten der Primär-Anzeigefläche angezeigt werden.

#### KURZBESCHREIBUNG DER ZEICHNUNGEN

**[0010]** [Fig. 1](#) zeigt eine Standardanzeige des Standes der Technik.

**[0011]** [Fig. 2](#) zeigt eine Standardanzeige mit einer Overscan-Benutzerschnittstelle in der unteren Overscan-Fläche.

**[0012]** [Fig. 3](#) zeigt eine Standardanzeige mit einer Overscan-Benutzerschnittstelle auf allen vier Rändern der Anzeige.

**[0013]** [Fig. 4](#) zeigt die Komponenten des Computersystems, die das Video-Anzeigesystem betreffen.

**[0014]** [Fig. 5](#) zeigt einen Cursor oder Zeiger in der Overscan-Benutzerschnittstelle und den Hotspot darüber in der Standardanzeige.

**[0015]** [Fig. 6](#) zeigt den nutzbaren Rand in dem vertikalen Overscan und dem horizontalen Overscan, der die Standardanzeige umgibt,

**[0016]** [Fig. 7](#) ist ein Übersichts-Flussdiagramm, das die Arbeitsweise einer bevorzugten Ausführung der vorliegenden Erfindung zeigt.

**[0017]** [Fig. 8](#) ist ein Flussdiagramm der Unterschritte im Display-Identifizieren-Schritt **102** von [Fig. 7](#).

**[0018]** [Fig. 9](#) ist ein Flussdiagramm der Unterschritte des Schrittes zum Ändern der Anzeigeauflösung **114** in [Fig. 7](#).

**[0019]** [Fig. 10](#) ist ein Flussdiagramm der Unterschritte des Anzeige-Malen-Schrittes **120** von [Fig. 7](#).

**[0020]** [Fig. 11](#) ist ein Flussdiagramm der Unterschritte des Schrittes zum Freigeben linearer Adressierung **112** von [Fig. 7](#).

**[0021]** [Fig. 12](#) ist ein Flussdiagramm der Unterschritte der Nachrichten-Verarbeitungsschleife von [Fig. 7](#).

[0022] [Fig. 13](#) ist ein Flussdiagramm der Unterschritte des Schrittes zum Prüfen von Maus- und Keyboard-Ereignissen [184](#) in [Fig. 12](#).

[0023] [Fig. 14](#) ist ein Flussdiagramm der Unterschritte des Schrittes zum Ändern der Emulationsauflösung [115](#) in [Fig. 7](#).

#### AUSFÜHRLICHE BESCHREIBUNG EINER BEVORZUGTEN AUSFÜHRUNG

[0024] Die vorliegende Erfindung umfasst Verfahren zum Bereitstellen und Verwenden einer Sekundär- oder Zusatz-Benutzerschnittstelle, vorzugsweise einer sekundären grafischen Benutzerschnittstelle oder Sekundär-GUI, die auf der Anzeige wenigstens scheinbar gleichzeitig mit der Primär-Benutzerschnittstelle, z.B. der herkömmlichen Desktop-GUI, vorhanden sein soll.

[0025] Bei einer bevorzugten Ausführung stellen Programmier-Mechanismen und Schnittstellen in einem Computersystem die Sekundär-GUI in einer bequemen und gegenwärtig unbenutzten potenziellen Anzeigefläche bereit, durch Bereitstellen von Zugang und Sichtbarkeit für einen Abschnitt der Monitoranzeige, der normal ignoriert wird und unzugänglich ist (im Folgenden "Overscan-Bereich"). [Fig. 1](#) zeigt einen Standard-Anzeige-Desktop des Standes der Technik, der Microsoft Windows™ fährt. In dem Desktop [31](#) befinden sich die Taskleiste [32](#) und Desktop-Icons [33](#).

[0026] Bei einer bevorzugten Ausführung der vorliegenden Erfindung wird ein grafisches Benutzer-Schnittstellenbild auf eine oder mehrere Seiten des Overscan-Bereichs gemalt, wie in [Fig. 2](#) und [Fig. 3](#) gezeigt. [Fig. 2](#) und [Fig. 3](#) zeigen Darstellungen einer Super VGA (SVGA) Anzeige, wobei zusätzlich eine grafische Balken-Benutzerschnittstelle in dem Overscan-Bereich angezeigt wird. Der Overscan-Benutzerschnittstellenbalken [30](#) ist definiert, außerhalb der Grenzen des "Desktop"-Anzeigebereichs [31](#) zu liegen. In [Fig. 2](#) ist die Anzeige modifiziert, um eine grafische Benutzerschnittstelle [30](#) in einem 20 Pixel hohen Balken unter der Unterkante zu enthalten. In [Fig. 3](#) ist die Anzeige modifiziert, um eine grafische Benutzerschnittstelle in vier je 20 Pixel hohen/breiten Balken außerhalb aller vier Anzeigekanten zu enthalten: ein unterer Balken [30](#), ein linker Balken [34](#), ein rechter Balken [36](#) und ein oberer Balken [38](#).

[0027] Die Overscan-Schnittstelle kann Schaltflächen, Menüs, Anwendungs-Ausgabesteuerungen (z.B. ein "Ticker-Fenster"), Animationen und Benutzer-Eingabesteuerungen (z.B. Editierfelder) enthalten, ist aber nicht darauf begrenzt. Weil die Overscan-Schnittstelle durch andere Anwendungen, die in dem Standani-Desktop laufen, nicht getrübt wird, kann die Overscan-Schnittstelle dauernd sichtbar sein oder sie kann zwischen einem sichtbaren und einem unsichtbaren Zustand basierend auf einer Anzahl von Programmierparametern umschalten (einschließlich, aber nicht darauf begrenzt, des Zustands des aktiven Fensters, des Zustands einer Umschall-Schaltfläche usw.).

[0028] [Fig. 4](#) zeigt die Hauptkomponenten des Computersystems, die das Video-Anzeigesystem betreffen. In der Software-Komponente [5](#) befinden sich das Betriebssystem [63](#) und Anwendungen [61](#). In den geschützten Modi moderner Systeme haben Anwendungen [61](#) keinen direkten Zugang zu den Video- oder Grafik-Treibern [64](#) oder Hardware-Komponenten, z.B. die Videokarte [66](#), die den Video-Chipsatz [66A](#), [66B](#) und [66C](#) enthält. Abstraktionsschichten, z.B. Anwendungsschnittstelle (AOI) [60](#) und/oder Direkt-API [62](#), stellen begrenzten Zugang bereit, oft durch das Betriebssystem [63](#).

[0029] Die Erfindung stellt ein Verfahren zum Malen und Zugreifen auf einen Bereich der Computeranzeige in Grafikmodi bereit, die normalerweise nicht zugänglich sind oder benutzt werden. In den Microsoft Windows-Umgebungen (einschließlich Windows 95 und Derivaten und Windows NT 4.0 und Derivaten) und anderen zeitgemäßen Betriebsumgebungen wird der primäre Anzeigebereich "Desktop" durch das Betriebssystem so zugewiesen, dass er ein Satz von vorbestimmten Video-"Modi" ist, wie z.B. in Tabellen 1 und 2 unten dargelegt, von denen jeder mit einer spezifischen Pixelauflösung vordefiniert ist. Der zugängliche Bereich der Computeranzeige kann daher nicht modifiziert werden, außer durch Auswählen eines anderen der verfügbaren vordefinierten Modi.

TABELLE 1: ROM BIOS Videomodi

Mode Number	Resolution	Mode Colors	Buffer Type	Segment
00H	42x25 chars (320x350 pixels)	16	Alpha	B800
00H	42x25 chars (320x350 pixels)	16	Alpha	B800
00H	42x25 chars (320x400 pixels)	16	Alpha	B800
00H	42x25 chars (320x400 pixels)	16	Alpha	B800
01H	42x25 chars (320x200 pixels)	16	Alpha	B800
01H	42x25 chars (320x350 pixels)	16	Alpha	B800
01H	42x25 chars (320x400 pixels)	16	Alpha	B800
01H	42x25 chars (320x400 pixels)	16	Alpha	B800
02H	80x25 chars (640x200 pixels)	16	Alpha	B800
02H	80x25 chars (640x350 pixels)	16	Alpha	B800
02H	80x25 chars (640x400 pixels)	16	Alpha	B800
02H	80x25 chars (640x400 pixels)	16	Alpha	B800
03H	80x25 chars (640x200 pixels)	16	Alpha	B800
03H	80x25 chars (640x350 pixels)	16	Alpha	B800
03H	80x25 chars (640x400 pixels)	16	Alpha	B800
03H	80x25 chars (720x400 pixels)	16	Alpha	B800
04H	320x200 pixels	4	Graphics	B800
05H	320x200 pixels	4	Graphics	B800
06H	840x200 pixels	2	Graphics	B800
07H	80x25 chars (720x350 pixels)	2	Alpha	B000
07H	80x25 chars (720x400 pixels)	2	Alpha	B000
0DH	320x200 pixels	16	Graphics	A000
0EH	640x200 pixels	16	Graphics	A000
0FH	640x350 pixels	4	Graphics	A000
10H	640x350 pixels	4	Graphics	A000
10H	640x350 pixels	16	Graphics	A000
11H	640x480 pixels	2	Graphics	A000
12H	640x480 pixels	16	Graphics	A000
13H	320x200 pixels	256	Graphics	A000

TABELLE 2: In der VESA BIOS Erweiterung definierte SVGA-Videomodi

<b>Mode Number</b>	<b>Resolution</b>	<b>Mode Colors</b>	<b>Buffer Type</b>
100H	640x480 pixels	256	Graphics
101H	640x480 pixels	256	Graphics
102H	800x600 pixels	16	Graphics
103H	800x600 pixels	256	Graphics
104H	1024x768 pixels	16	Graphics
105H	1024x768 pixels	256	Graphics
106H	1280x1024 pixels	16	Graphics
107H	1280x1024 pixels	256	Graphics
108H	80x60 chars	16	Alpha
109H	132x25 chars	16	Alpha
10AH	132x43 chars	16	Alpha
10BH	132x50 chars	16	Alpha
10CH	132x60 chars	16	Alpha
10DH	320x200 pixels	32,768	Graphics
10EH	320x200 pixels	65,536	Graphics

<b>10FH</b>	<b>320x200 pixels</b>	<b>16,777,216</b>	<b>Graphics</b>
<b>110H</b>	<b>640x480 pixels</b>	<b>32,768</b>	<b>Graphics</b>
<b>111H</b>	<b>640x480 pixels</b>	<b>65,536</b>	<b>Graphics</b>
<b>112H</b>	<b>640x480 pixels</b>	<b>16,777,216</b>	<b>Graphics</b>
<b>113H</b>	<b>800x600 pixels</b>	<b>32,768</b>	<b>Graphics</b>
<b>114H</b>	<b>800x600 pixels</b>	<b>65,536</b>	<b>Graphics</b>
<b>115H</b>	<b>800x600 pixels</b>	<b>16,777,216</b>	<b>Graphics</b>
<b>116H</b>	<b>1024x788 pixels</b>	<b>32,768</b>	<b>Graphics</b>
<b>117H</b>	<b>1024x768 pixels</b>	<b>65,536</b>	<b>Graphics</b>
<b>118H</b>	<b>1024x768 pixels</b>	<b>16,777,216</b>	<b>Graphics</b>
<b>119H</b>	<b>1280x1024 pixels</b>	<b>32,768</b>	<b>Graphics</b>
<b>11AH</b>	<b>1280x1024 pixels</b>	<b>65,536</b>	<b>Graphics</b>
<b>11BH</b>	<b>1280x1024 pixels</b>	<b>16,777,216</b>	<b>Graphics</b>

[0030] Wie in [Fig. 6](#) gezeigt, wird ein angezeigtes Bild "overscanned". Das heißt, die angezeigten Videopufferdaten belegen weniger als die ganze treibbare Schirmgröße. Die Breite des nutzbaren Overscan-Randes hängt von dem Betrag des Horizontal-Overscans **52** vermindert um die Horizontalaustastung **54** und dem Betrag des Vertikal-Overscans **53** vermindert um die Vertikalaustastung **55** ab.

[0031] In einer ersten bevorzugten Ausführung wird nur ein Rand am Boden des Standard-Anzeigebereichs benutzt. Folglich müssen nur die Vertikal-Steuerparameter für den Kathodenstrahlröhren-(CTR)Controller, gezeigt als Steuerregister **6H**, **16H**, **11H**, **10H**, **12H** und **15H** in [Fig. 4](#), justiert werden. Diese Parameter und andere werden in Tabelle 3 unten gezeigt.

TABELLE 3: Vertikal-Timing-Parameter für CR-Programmierung

Register	Name	Description
6H	Vertical Total	Value = (total number of scan lines per frame) - 2 The high-order bits of this value are stored in the overflow registers.
7H	Overflow	High-order bits from other CR registers.
10H	Vertical Retrace Start	Scan line at which vertical retrace starts. The high-order bits of this value are stored in the overflow registers.
11H	Vertical Retrace End	Only the low-order 4 bits of the actual Vertical Retrace End value are stored. (Bit 7 is set to 1 to write-protect registers 0 through 7.)
12H	Vertical Display End	Scan line at which display on the screen ends. The high-order bits of this value are stored in the overflow registers.
15H	Start Vertical Blank	Scan line at which vertical blanking starts. The high-order bits of this value are stored in the overflow registers.
16H	End Vertical Blank	Scan line at which vertical blanking ends. The high order bits of this value are stored in the overflow registers.
59H-5AH	Linear Address Window Position	Linear address window position in 32-bit CPU address space.

[0032] In den Standard 640 × 480 Grafikmodus beträgt die nominelle Horizontal-Abtastrate 31.5 kHz (31,500 Mal pro Sekunde) mit einer Vertikal-Abtastrate 60 Hz (60 Bilder pro Sekunde). Die Zahl von Zeilen in einem Bild ist somit 31,500/60 oder 525. Weil nur 480 Datenzeilen angezeigt werden müssen, gibt es 525–480 oder 45 Zeilen, die für den Vertikal-Overscan zur Verfügung stehen. Es bleibt mehr als genug Spielraum für den Rücklauf, der nur die Zeit von 2 Zeilen kostet. Die bevorzugte Ausführung verwendet 20 Zeilen für die erfundene Overscan-Anzeige.

[0033] Das offenbarte Verfahren der bevorzugten Ausführung der vorliegenden Erfindung wird durch Erfüllen von drei Forderungen zu Stande gebracht:

- (1) Adressieren und Modifizieren der sichtbaren Auflösung des Video-Anzeigesystems so, dass Abschnitte des Overscan-Bereichs sichtbar sind, wie in [Fig. 6](#) gezeigt,
- (2) Adressieren und Modifizieren des Video-Anzeigeinhalts für den sichtbaren Abschnitt des Overscan-Bereichs, und
- (3) Bereitstellen einer Anwendungsprogrammierungsschnittstelle (API) oder anderen Mechanismus, um Anwendungen zu erlauben, diese Funktionalität zu implementieren.

[0034] [Fig. 7](#) und die in [Fig. 8–Fig. 13](#) bereitgestellten weiteren Details und Unterschritte stellen ein Flussdiagramm einer Implementierung einer bevorzugten Ausführung der vorliegenden Erfindung bereit, die die oben beschriebenen Forderungen erfüllt. Die Umgebung dieser Implementierung ist eine Standard Microsoft Windows™ Betriebsumgebung, die Microsoft Visual C und Microsoft MASM für die Entwicklungsplattform verwendet. Dies bedeutet nicht, dass diese Erfindung im Umfang auf diese Umgebung oder Plattform beschränkt ist. Die Erfindung könnte in jeder grafischen Schnittstenumgebung, z.B. OSF Motif, Apple OS, Java OS und andere, implementiert werden, in der ähnliche Video-Standards (VGA, SVGA, XGA, 8514/A) praktiziert werden. Die Handbücher PC Video Systems von Richard Wilton, veröffentlicht von Microsoft Press und Programmer's Guide für die EGA-, VGA- und Super VGA-Karten von Richard F. Ferrano, veröffentlicht von Addison Wesley liefern mehr als angemessene Hintergrundinformation, um diese Ausführung zu implementieren.

[0035] Nun besonders auf [Fig. 7](#) verweisend versucht nach Initialisierung das Programm in Schritt 102 den Anzeigetyp und die von dem Anzeigetreiber benutzte gegenwärtige Stelle in Speicher zu bestimmen, um die Größe und Stellen aller vorzunehmenden Anzeigemodifikationen zu bestimmen, z.B. die Größe und Stelle des zu verwendenden Overscan-Bereichs.

[0036] Wie in [Fig. 8](#) weiter im Einzelnen beschrieben, fragt das Programm zuerst in Schritt 131 die Hardware-Registrierung ab, um zu versuchen, den registrierten Hardware-Typ zu bestimmen. Wenn erfolgreich, ermittelt das Programm dann Kompatibilitäts-Information, Schritt 135, um zu verifizieren, dass das Programm diesen Anzeigetyp unterstützt, und um Speicher-Zuteilungsinformation zu ermitteln.

**[0037]** Wenn die Hardware-Registrierinformation nicht verfügbar ist, wie in Schritt **131** bestimmt, oder der in Schritt **131** bestimmte Anzeigetyp nicht unterstützt wird, wie durch Schritt **104** bestimmt, kann das Programm einen alternativen Weg verwenden, gezeigt als Unterprogramm Hardware-Abfragen, Schritte **135** in [Fig. 8](#), um das BIOS in Schritt **134** und den Chipsatz in Schritt **136** nach ähnlicher Information, wie direkt unten beschrieben, abzufragen.

**[0038]** Wenn in Schritt **134** auf das BIOS zuzugreifen ist, wird zuerst in Schritt **132** physikalischer Speicher zugeteilt und darauf unter Verwendung von Microsoft DPML (DOS Protected Mode Interface) zugegriffen, um ihn in die lineare Speicheradresse abzubilden, in der das BIOS residiert, unter Verwendung von DPML, um lineare BIOS-Adresse physikalischem Speicher zuzuweisen, Schritt **133**.

**[0039]** Danach fragt das Programm das BIOS in Lesen-BIOS-Block Suchen nach VGA/XVA-Typ und Hersteller-ID ab, Schritt **134**. Wenn erfolgreich, werden der Treiber und Chipsatz weiter nach dem genauen Chipsatz abgefragt, um den Anzeigetyp und Speicherstelle in Treiber/Chipsatz Abfragen zu bestimmen, Schritt **136**.

**[0040]** Wenn die Kompatibilitäts-Information kein Standard VGA, SVGA, XGA oder 8514/A Signatur angibt, Schritt **134**, gibt diese Routine einen Fehler zurück. Wenn eine bekannte Chipsatz-Hersteller-ID gefunden wird, kann der Treiber und/oder Chipsatz mit herstellerspezifischen Routinen abgefragt werden, Schritt **136**, um den spezifischen Chipsatz zu identifizieren, und, wenn nötig, zu initialisieren.

**[0041]** Wenn in Schritt **104** das Programm außerstande war, den Anzeigetyp zu identifizieren, weil entweder die Registrierungsabfrage in Schritt **131** oder die Hardware-Abfrage in Schritt **135** erfolglos war, kann der Benutzer in Laufen im Windows-Modus, Schritt **116**, dahin gehend abgefragt werden, ob das Programm als eine Standard "Anwendungsleiste" oder "Werkzeugeleiste" weiterlaufen soll. Das Programm kann entweder abtreten oder als eine Werkzeugeleiste auf dem Desktop weiterlaufen.

**[0042]** Wenn zu [Fig. 8](#) zurückkehrend ein unterstützter Anzeigetyp erfasst wird, bestimmt das Programm in Ränder Identifizieren die Schirmränder, auf die zuzugreifen ist, um im Overscan anzuzeigen, Schritt **106**, basierend auf Benutzer-Präferenzen, und bestimmt, wenn nötig, ob genügend Videospeicher vorhanden ist, um die Anzeigeänderungen vorzunehmen. Wenn z.B. der Schirm gegenwärtig auf eine 1024 × 768 Auflösung bei 16 Bit pro Pixel eingestellt ist, und das Programm vier grafische Schnittstellenbalken, einen auf jeder Kante, wobei jeder Balken 20 Pixel tief ist, einschließen soll, muss das Programm prüfen, ob der Videospeicher größer als 1.7 MB ist (benötigte Zahl von Bytes = Pixelbreite × Bits pro Pixel × Pixelhöhe).

**[0043]** Auf die Controller-Register **6H**, **16H**, **11H**, **10H**, **12H** und **15H**, wie in [Fig. 4](#) gezeigt und in Tabelle 3 aufgeführt, kann durch Standard-Eingabe/Ausgabe-Ports unter Verwendung von Standard-Eingabe/Ausgabe-Funktionen zugegriffen werden. Die CR-Register **6H**, **16H**, **11H**, **10H**, **12H** und **15H** müssen zuerst entriegelt werden, wie in Unlock-CRTC-Registern angegeben, Schritt **108** in [Fig. 7](#), um sie beschreibbar zu machen. Sie werden durch Löschen von Bit 7 im Controller Register **11H** entriegelt.

**[0044]** Das Adressieren von Videospeicher, Schritt **112**, erfolgt durch eines von mehreren Mitteln. Eines ist, das Standard VGA 64 kB "Hardware-Fenster" zu benutzen und es entlang dem Videospeicherpuffer **67** ([Fig. 4](#)) in 64 kB Schritten, wenn nötig, zu bewegen. Das bevorzugte Verfahren ist, lineare Adressierung durch Abfragen des Video-Chipsatzes nach der linearen Fensterpositionsadresse zu ermöglichen, Schritt **138** von [Fig. 11](#). Dieser 32-Bit Offset im Speicher erlaubt dem Programm, den linearen Speicher in eine physikalische Adresse abzubilden, Schritte **140** und **142** von [Fig. 11](#), die programmatisch manipuliert werden kann.

**[0045]** An diesem Punkt kann das Programm die Anzeige modifizieren, Schritt **114** und [Fig. 9](#), um die Randbereiche zu inkrementieren. Diese Routine stellt zuerst fest, ob das System im "Werkzeugeleisten"-Modus arbeitet, Schritt **144**, und gibt, wenn ja, wahr zurück. Wenn nicht, stellt sie fest, ob alle Register und Werte auf ihren Originalzustand zurückzusetzen sind, um so die Anzeige effektiv in ihr ursprüngliches Aussehen zurückzubringen, Schritt **152**. Die Feststellung basiert auf einer Anzahl von Parametern, z.B., ob die gegenwärtige Auflösung, Schritt **146**, einen Standardwert oder eine frühere programmatische Manipulation widerspiegelt, Schritt **148**. Wenn bereits eine Standard-Auflösung eingestellt ist, werden die Variablen zurückgesetzt, um die spezifizierten Randbereiche zu enthalten (Schritt **150**). Die CR-Register werden erhöht, Schritt **154**, um die abgetasteten und ausgetasteten Bereiche der Anzeige zu modifizieren. Wenn der obere oder seitliche Bereiche modifiziert werden, wird vorhandener Videospeicher in Schritt **162** von [Fig. 10](#) entsprechend verschoben.

**[0046]** Wenn eine der vorangehenden Routinen einen Fehler zurückgibt, kann das Programm den Benutzer auffordern, festzustellen, ob "Emulations"-Modus, Schritt **13**, oder Fenstermodus, Schritt **116**, benutzt werden

soll, oder ob das Programm in Schritt **124** abtreten soll.

**[0047]** In ihrer einfachsten Form kann die Erfindung als ein Verfahren zum Hinzufügen einer Sekundär-GUI durch Rekonfigurieren des wirklichen Anzeigemodus, um einen modifizierten Nicht-Standard-GUI-Modus hinzuzufügen, in dem die Standard-Anzeigegröße oder Auflösung erhöht wurde, um zusätzlich zu der Standard-Anzeige eine sekundäre Anzeige zu enthalten, behandelt werden. Zum Beispiel wird eine Standard 640 × 480 Anzeige erfindungsgemäß modifiziert, um eine größere Anzeige zu werden, wobei ein Abschnitt davon der ursprünglichen 640 × 480 Anzeige entspricht, während ein anderer Abschnitt einer 640 × 25 Sekundär-GUI-Anzeige entspricht.

**[0048]** Es sind verschiedene Verfahren oder Mechanismen zum Modifizieren des Systems erforderlich, um die Sekundär-GUI einzuschließen, abhängig von den Anforderungen der Sekundär-GUI und den gegenwärtigen Gegebenheiten des unmodifizierten Systems.

**[0049]** In einer anderen Ausführung der vorliegenden Erfindung werden Systemressourcen für eine Sekundär-GUI zugeteilt, durch Täuschen des Videotreibers, zu größerer Auflösung zu gehen. Dieses Verfahren garantiert automatisch, dass genug Raum frei gehalten wird, da der Videotreiber Systemressourcen entsprechend der Auflösung zuteilt, von der der Videotreiber glaubt, dass er dann arbeiten wird. Um eine oder mehrere Sekundär-Benutzerschnittstellen in einem oder mehreren Bereichen des Schirmes zu betreiben, ist es erforderlich, einen Speicher zu haben, der im Videospeicher oder im Bildspeicher mit dieser Stelle verbunden war, zusammenhängend unter der Hauptoberfläche frei und verfügbar. Durch Schreiben einer Reihe von kleinen Applets spezifisch für Hardware, die bekannt ist, Systemressourcen-Zuteilungsprobleme zu haben, für eine Sekundär Benutzerschnittstelle kann die Sekundär-Benutzerschnittstellenanwendung ein solches Applet immer dann fahren, wenn Auflösungen gewechselt werden, und den zu diesem einzelnen Applet gehörenden Chipsatz initialisieren. Wenn die Anwendung ein zu dem gegenwärtigen bestimmten Chipsatz gehörendes Applet findet, wird es gestartet. Das Applet oder Minitreiber initialisiert sich selbst, führt notwendige Änderungen an den Videoauflösungstabellen des Treibers durch, erzwingt eine Neufreigabe, und ausreichender Raum ist anschließend für eine oder mehrere Sekundär-Benutzerschnittstellen verfügbar.

**[0050]** Wenn neu freigegeben, teilt der Treiber Videospeicher für die Primäranzeige entsprechend den Daten in den UCCO-Auflösungstabellen nach Bedarf zu. Die modifizierten Werte resultieren daher in einer größeren Zuteilung. Sobald der Treiber den für die Primär-Oberfläche benötigten Speicher zugeteilt hat, wird der Treiber keinen äußeren Zugriff auf den zugeteilten Speicher erlauben. Durch Täuschen des Treibers, zu glauben, dass er genügend Speicher für eine Auflösung genau x Bytes größer als die gegenwärtige Auflösung, wo x die Größe von einer oder mehreren Sekundär-Benutzerschnittstellen ist, zuteilen muss, kann die Anwendung sicher sein, dass kein interner oder externer Gebrauch der zugeteilten Speicherstelle mit der Sekundär-Benutzerschnittstelle in Konflikt gerät.

**[0051]** Dieses Verfahren stellt sicher, dass Systemressourcen für eine oder mehrere Sekundär-Benutzerschnittstellen durch Schreiben eines Applets zugeteilt werden, das den Videotreiber so adressieren würde, dass der Videotreiber bei seiner nächsten Neufreigabe gezwungen wird, Videospeicher ausreichend für eine Auflösung höher als die wirkliche Betriebssystem-Auflösung zuzuteilen. Dies kann auch durch Modifizieren jeder Version der Werbemodus-Tabellen getan werden, um so eine Schirmgröße größer als die Primär-Benutzerschnittstellen-Schirmgröße zu erzeugen.

**[0052]** Dieses Verfahren hat einen zusätzlichen Nutzen des Beseitigens der Notwendigkeit, zu verhindern, dass der Treiber wirklich zu der spezifizierten größeren Auflösung wechselt, die der Primär-Benutzerschnittstelle eine größere Anzeigeflächenauflösung übergibt. Die "Hardware-Modustabelle", eine Variante der vorerwähnten Videoauflösungstabellen, wird nicht bekannt gegeben und ist nicht zugänglich. Wenn der Treiber die neue Auflösung bestätigt, wird daher das Prüfen gegen die Hardware-Modustabelle immer fehlschlagen, und sich daher weigern, zu dieser Auflösung zu wechseln. Weil dieses Verfahren die bekannt gegebenen Videoauflösungstabellen früh genug in dem Prozess des Treibers modifizierte, wurde der zugeteilte Speicher modifiziert und Speicheradressen vor dem Fehler in einen gültigen Modus gesetzt. Danach, wenn die CRTCs in Schritt **114** modifiziert werden, reserviert der Treiber genügend Speicher für eine oder mehrere Sekundär-Benutzerschnittstellen und macht ihn für jeden anderen Prozess oder Zweck unverfügbar.

**[0053]** In noch einer anderen Ausführung der vorliegenden Erfindung wird ein einhüllender Treiber installiert, der über dem bestehenden Treiber sitzt und sich zwischen die Hardware-Abstraktionsschicht und den wirklichen Videotreiber legt, um in der Lage zu sein, alle Aufrufe an den Videotreiber zu behandeln und den Treiber und die Treibertabellen in einer viel generischeren Weise als in einer chipsatzspezifischen Weise zu modifizie-

ren. Der umhüllende Treiber legt sich in den Primär-Videotreiber, um Aufrufe an den Primär-Videotreiber hin und her transparent zu übergeben. Der umhüllende Treiber findet die Videoauflösungstabellen in dem Primär-Videotreiber, die an einer Anzahl von Stellen in dem Treiber sein können. Der umhüllende Treiber modifiziert die Tabellen (z.B. Erhöhen von 800 mal 600 auf 800 mal 620). Ein 1024 mal 768 Tabelleneintrag kann 1024 mal 800 werden.

**[0054]** Wie die vorher beschriebene Ausführung kann der Primärtreiber die neue Auflösung nicht bestätigen und kann daher die Anzeigeeinstellung nicht wirklich ändern. Als Folge teilte der Treiber Speicher zu, teilte Cach-Raum zu, bestimmt durch Speicheradresse, und den Cache und Offscreen-Puffer nach Bedarf. Der Primärtreiber benutzt somit niemals den ganzen zugeordneten Raum und wird niemals in diesem Raum zeichnen.

**[0055]** Wie oben erwähnt, enthält das Verfahren der vorliegenden Erfindung die Hauptschritte, Finden des Overscan-Bereichs, Vergrößern oder Erweitern des Overscan-Bereichs und Legen von Daten in der erweiterten Overscan-Bereich.

**[0056]** Der Schritt des Findens des Overscan-Bereichs erfordert eine Durchsicht des Inhalts der Controller-Register, die CR-Register, die von VGA-kompatiblen Chipsätzen oder Grafikkarten benutzt werden, um zu identifizieren, wohin der Overscan-Bereich, das Austasten, das vertikale und horizontale Gesamt und das Senken gesetzt werden sollen. Das CR definiert die Desktop-Anzeige, wie sie synchronisiert wird, wo ihr Layout links und rechts ist, wieviel Pufferbereich es auf jeder Seite geben würde, und wie sie in dem Videospeicherbereich gespeichert werden würde. Eine Durchsicht des Inhalts der CR-Datenregister definiert daher voll die Stelle und Größe des Overscan-Bereichs.

**[0057]** Um den Schritt des Erweiterns des Overscan-Bereichs zu vollbringen, können die CRs gegenwärtig direkt für Systeme mit Videoanzeigaufösungen bis zu und einschließlich 1024 Pixel in jeder Abmessung benutzt werden, das heißt, Auflösungen, die in den allgemein akzeptierten VGA-Standards durch 10 Bit pro Register definiert werden können. Um den Overscan-Bereich zu erweitern, werden neue Daten unter Verwendung von Standard-Verfahren wie die Input- und Output-Funktionen in das CR geschrieben. Ein Standard-Videoport und MMIO-Funktionen können ebenfalls benutzt werden, um die CRs zu modifizieren.

**[0058]** Bei größeren Auflösungen können 11 Bit nötig sein, um die Auflösung richtig zu definieren. Es gibt gegenwärtig kein Standardverfahren, in dem die 11-te Bitstelle definiert ist. Bei einer Auflösung von z.B. über  $1280 \times 1024$  ist daher gegenwärtig ein Verständnis der Videokarte selbst, besonders, wie die 11 Bits, die die Auflösung darstellen, gespeichert werden, erforderlich und wird unten ausführlicher beschrieben.

**[0059]** Wenn der Overscan-Bereich erweitert wird, ist es wichtig, sicherzustellen, dass ein vorheriger Overscan nicht bereits angezeigt wird, vielleicht von einem früheren Absturz oder einem anderen unerwarteten Problem. Entweder die Anzeige muss sofort auf die geeigneten Auflösungs-Vorgabewerte zurückgesetzt werden oder das CR muss abgefragt werden, um festzustellen, ob die Gesamtschirmauflösung, wie von der Videokarte und den Treibern verstanden, sich von der durch die Betriebssystem-Anzeigeschnittstelle bekannte Schirmauflösung unterscheidet. Ein Overscan-Balken kann bereits angezeigt werden, wenn die Gesamtschirmauflösung nicht gleich einer der Standard-VGA- oder SVGA-Auflösungen ist. Das heißt, wenn die Gesamtschirmauflösung gleich einer Standard-VGA/SVGA-Auflösung plus dem für den Overscan-Balken benötigten Bereich ist oder größer ist als die von der Betriebssystem-Anzeigeschnittstelle mitgeteilte Auflösung, wird die Anzeige zurückgesetzt.

**[0060]** Sobald der Anzeigebereich oder die Auflösung, wie in dem CR gespeichert, bestimmt ist, kann die Auflösung oder der Anzeigebereich auf mehreren verschiedenen Wegen erweitert werden. Der Overscan-Bereich kann dem Boden, der Spitze oder der rechten Seite des gegenwärtigen Anzeigebereichs hinzugefügt werden, und optional kann der Anzeigebereich umpositioniert werden, sodass der Overscan-Balken im Aussehen zentriert bleiben kann. Alternativ kann der Overscan-Bereich irgendwo hinzugefügt werden, und der Original- oder Desktop-Anzeige reich kann zentriert werden, um das Aussehen zu verbessern. In jedem Fall wird die Höhe/Breite des für den Overscan-Balken benötigten Bereichs zu der bereits in dem CR gespeicherten Größe des Anzeigebereichs addiert, und die Summe wird in das CR geschrieben, wobei die früheren Daten überschrieben werden.

**[0061]** Der Schirm zeigt typischerweise ein schnelles Aufblitzen, wenn er in einen anderen Modus gebracht wird, einschließlich des Original-Anzeigebereichs plus einem neuen Anzeigebalken in dem Overscan-Bereich. Sobald diese Änderung eintritt, kann eine schwarze Maske über den neuen Bereichen positioniert werden. Die neuen Menüdaten können dann sicher oben auf die Maske geschrieben werden, sodass der Benutzer niemals

Speicher-"Schrott" sieht.

**[0062]** Typischerweise gibt es ein paar Sekunden Ladezeit, während der eine einfache Nachricht, z.B. "Laden ...", angezeigt werden kann, um Verwirrung des Benutzers zu vermeiden.

**[0063]** Es gibt eine Anzahl von Mechanismen, mit denen dies getan werden kann. Ein Satz von Klassen-Objekten wird benutzt, die alle von einer Grundklasse abgeleitet werden, die der oben beschriebenen generischen VGA-Technik entspricht.

**[0064]** Der erste Mechanismus ist eine Implementierung der generischen VGA-Technik. Bei Verwendung dieses Mechanismus ist keine für eine Videokarte spezifische Information erforderlich, außer der, die VGA-Unterstützung sicherstellt. Unter Verwendung von Standard-Anwendungsprogrammierschnittstellen-(API)Routinen werden Primär- und Sekundär-Oberflächen zugeteilt. Die neuen Anzeigedaten in dem CR sind einfach die physikalische Adresse am Anfang der Primär-Oberfläche plus die durch die Schirmgröße definierten Zahl von Pixeln.

**[0065]** Die Zuteilung der Primär-Oberfläche wird immer auf der ganzen Schirmanzeige basieren. Die lineare Adresse der zugeteilten Primär-Oberfläche gegeben, von der eine physikalische Adresse abgeleitet werden kann, kann extrapoliert werden, dass die physikalische Adresse der Stelle im Videospeicher unmittelbar angrenzend an die Primär-Oberfläche durch die Summe der Zahl von Speicherbytes, die zum Unterhalten der Primär-Oberfläche im Speicher benutzt werden, plus dem Wert der physikalischen Adresse der Primär-Oberfläche dargestellt wird.

**[0066]** Sobald die physikalische Adresse der Primär-Oberfläche bekannt ist, kann die Größe der Primär-Oberfläche, wie im Videospeicher dargestellt, bestimmt werden.

**[0067]** Das System sucht z.B. in den CRs nach der Auflösung des Schirms, 800 mal 600, in Form der Zahl von Bits pro Pixel oder Bytes pro Pixel. Dann werden alle in dem CR gespeicherten Daten, die jeden horizontalen Synchronisationsraum darstellen, addiert. Dies ist die wahre Abtastzeilenlänge. Die Abtastzeilenlänge ist ein genaueres Maß der Breite einer gegebenen Auflösung.

**[0068]** Als Nächstes wird die physikalische Adresse der zugeteilten Sekundär-Oberfläche aus ihrer linearen Adresse gewonnen. In dem Fall, wo die zugeteilte Sekundär-Oberfläche tatsächlich in dem Speicherraum angrenzend an die Primär-Oberfläche zugeteilt wird (der Wert der physikalischen Adresse der Sekundär-Oberfläche ist gleich dem Wert der physikalischen Adresse der Primär-Oberfläche plus der Größe der Primär-Oberfläche), wird die Sekundär-Oberfläche als die Stelle im Speicher für die Overscan-Anzeige bestimmt.

**[0069]** Wenn jedoch das Obige nicht wahr ist und die Sekundär-Oberfläche nicht an die Primär-Oberfläche angrenzt, ist ein anderer Lösungsmechanismus erforderlich.

**[0070]** Zusammengefasst, der erste Mechanismus bestimmt, was der physikalische Bereich für den Desktop sein wird und fügt dann einen Sekundärraum darunter hinzu, um ihn in dem Overscan-Bereich anzuzeigen. Der neu zugeteilte Bereich wird der allererste verfügbare Speicherblock sein. Wenn dieser Block unmittelbar auf die Primär-Oberfläche folgt, wird die physikalische Adresse dem mit der physikalischen Adresse der Primär-Oberfläche verbundenen Wert plus der Größe der Primär-Oberfläche entsprechen. Wenn dies wahr ist, sind die Speicherblöcke zusammenhängend, und dieser VGA-generische Mechanismus kann verwendet werden.

**[0071]** Wenn dieser erste VGA-generische Mechanismus nicht verwendet werden kann, werden der Videokarten- und Treibername und Versionsinformation, die aus der Hardware-Registrierung und BIOS zurückgewonnen werden, in Verbindung mit einer Look-up-Tabelle benutzt, um die besten Alternativen unter den verbleibenden Mechanismen zu bestimmen. Die Tabelle enthält einen Satz von Standards, die mit der in der Hardware-Registrierung gefundenen Liste von Treibernamen verbunden sind. Ein für den Video-Chipsatz spezifisches Klassenobjekt wird direkt oder indirekt basierend auf dem VGA-generischen Objekt eingerichtet.

**[0072]** Wenn das Nachsehen der Hardware keine zuverlässige Übereinstimmung ergibt, kann ein Zuverlässigkeits- oder Vertrauens-Fudge-Faktor, der nicht Teil der Erfindung ist, benutzt werden. Wenn z.B. das Hardware-Nachsehen feststelle, dass eine XYZ-Marken-Vorrichtung irgendeiner Art benutzt wird, aber die genannte einzelne XYZ-Vorrichtung in der Look-up-Tabelle nicht gefunden wird, kann oft ein generisches Modell von diesem Chipsatz-Hersteller brauchbar sein. Wenn keine Information verfügbar ist, kann der Benutzer eine

Nachricht erhalten, die angibt, dass die Hardware nicht unterstützt wird und dass das Programm in dem Overscan-Bereich nicht laufen kann. Der Benutzer kann dann gefragt werden, zu entscheiden, ob das System in dem "Anwendungs-Werkzeugeleisten"-Modus betrieben werden soll, der im Grunde mit genau der gleichen Funktionalität läuft, aber in einer Fensterumgebung in dem Desktop, anstelle in dem Overscan-Bereich außerhalb des Desktops.

**[0073]** Der nächste alternative Mechanismus, der auch nicht Teil der Erfindung ist, benutzt Oberflächen-Overlays. Der erste Schritt zu dieser Lösung ist, festzustellen, ob das System Oberflächen-Overlays unterstützt. Es erfolgt ein Aufruf an den Videotreiber, um festzustellen, welche Merkmale unterstützt werden und welche anderen Faktoren benötigt werden. Wenn Oberflächen-Overlays unterstützt werden, kann z.B. ein Skalierungsfaktor erforderlich sein.

**[0074]** Zum Beispiel könnte eine bestimmte Videokarte in einer gegebenen Maschine, die 2 MB an Video-RAM benutzt, unskalierte Oberflächen-Overlays mit  $1024 \times 768$  bei 8 Bit pro Pixel, nicht aber mit  $1024 \times 768$  bei 16 Bit pro Pixel unterstützen, weil die Bandbreite der Videokarte oder die Geschwindigkeit der Karte verbunden mit der relativ kleinen Menge an Videospeicher nicht ausreichend sein würde, ein Overlay mit voller Breite zu zeichnen. Oft ist es horizontales Skalieren, das zur Frage steht und den Treiber daran hindert, ein Overlay mit voller Breite zu zeichnen. Ein Overlay ist buchstäblich ein Bild, das oben auf der Primär-Oberfläche gezeichnet wird. Es ist keine Sekundär-Oberfläche, die oben beschrieben ist. Das System sendet buchstäblich sein Signal von den Videotreibern an die Hardware, sodass sie die zwei Signale miteinander vermischt, um ein zweites Signal über das erste zu legen.

**[0075]** Wenn ein System unskalierte Overlays nicht unterstützen kann, vielleicht wegen Bandbreiten- oder Speicherproblemen, ist dieser Mechanismus nicht wünschenswert. Er wird nicht verworfen, wird aber eine Alternative mit niedriger Priorität. Wenn z.B. der Skalierungsfaktor unter 0.1 ist, kann der normale Balken gezeichnet werden und er wird näher an der Kante beschnitten. Wenn der Skalierungsfaktor größer als 10% ist, wird ein anderer Lösungsmechanismus benötigt.

**[0076]** In dem nächsten Satz alternativer Mechanismen, die auch nicht Teil der Erfindung sind, wird eine Sekundär-Oberfläche zugeteilt, deren Größe ausreichend ist, den normalen Desktop-Anzeigebereich plus den zum Anzeigen des oder der Overscan-Balken(s) zu benutzenden Overscan-Bereich einzuschließen. Mit diesen Mechanismen muss sich die zugeteilte Sekundär-Oberfläche im Speicher nicht angrenzend an die Primär-Oberfläche befinden. Diese Lösungen verwenden jedoch mehr Videospeicher als die anderen.

**[0077]** Der erste Schritt ist, eine Sekundär-Oberfläche zuzuteilen, die ausreichend groß ist, um die Videoanzeige (die Primär-Oberfläche) und den zu verwendenden Overscan-Bereich zu enthalten. Wenn die Zuteilung misslingt, bedeutet das, dass nicht genug Videospeicher vorhanden ist, um diese Aufgabe zu erfüllen, und dieser Satz von Mechanismen wird ausgelassen und die nächste Alternative versucht. Nachdem ein neuer Speicherblock zugeteilt ist, wird ein Timer mit sehr kleiner Körnung benutzt, um eine einfache Speicherkopie des Inhalts der Primär-Oberfläche auf die geeignete Stelle dieser Sekundär-Oberfläche auszuführen. Der Timer führt die Kopie mit etwa 85 Mal pro Sekunde aus.

**[0078]** In diesem Satz von alternativen Mechanismen gibt es eine Variante, die Systemseitentabellen verwendet. Dieser Mechanismus fragt die Systemseitentabellen ab, um die gegenwärtige GDI-Oberflächenadresse zu bestimmen, d.h. die physikalische Adresse in der Seitentabelle für die Primär-Oberfläche. Eine Sekundär-Oberfläche wird dann erzeugt, die groß genug ist, um alles, was in dem Videospeicher ist, plus den Speicher zu haken, der für den anzuzeigenden Overscan-Balken benötigt wird. Diese Oberflächenadresse wird dann in die Systemseitentabelle geschoben und als die GDI-Oberflächenadresse geltend gemacht.

**[0079]** Danach, wenn GDI die Primär-Oberfläche durch den Treiber liest oder beschreibt, liest oder beschreibt sie tatsächlich die neue, größere Oberfläche. Das Overscan-Balken-Programm kann anschließend den durch die GDI nicht adressierten Bereich der Oberfläche modifizieren. Die ursprüngliche Primär-Oberfläche kann freigesetzt und die Speichernutzung neu beansprucht werden. Dieser Mechanismus, der speichereffizienter ist der vorher beschriebene Mechanismus, ist die bevorzugte Alternative. Aber die Speichertabellenlösung wird nicht richtig auf einem Chipsatz arbeiten, der eine Koprozessor-Vorrichtung enthält. Wenn die anfängliche Vorrichtungsabfrage ergibt, dass die Vorrichtung einen Koprozessor enthält, wird diese Mechanismus-Variante nicht versucht werden.

**[0080]** Andere Variationen der oben beschriebenen Mechanismen, die nicht Teil der Erfindung sind, werden in abgeleiteten Klassenobjekten berücksichtigt. Die VGA-generischen Mechanismen können z.B. variieren,

wenn die Videokarte mehr als 10 Bit nötig, um die Videoauflösung in dem CR darzustellen. Einige Fälle können 11 Bit benötigen. Solche Register benutzen typischerweise keine zusammenhängenden Bytes, sondern verwenden Erweiterungsbits, um die Adressinformation für die Bits höherer Ordnung zu bezeichnen.

**[0081]** In diesem Beispiel wird das elfte Bit gewöhnlich in einem erweiterten CR-Register spezifiziert, und die erweiterten CR-Register sind gewöhnlich chipspezifisch.

**[0082]** Desgleichen enthält eine Variation des Oberflächen-Overlay-Mechanismusses einen skalierungsfaktor, wie oben beschrieben. Diese Alternative wird in spezifischen Implementierungen durch abgeleitete Klassenobjekte gehandhabt und kann in bestimmten Situationen die beste Lösung sein.

**[0083]** Eine andere Implementierung dieser Technologie verwendet einen "Hooking"-Mechanismus, wie in [Fig. 14](#) gezeigt. Nachdem der Anzeigetreiber durch die Hardware-Registrierung oder das BIOS identifiziert ist, werden bestimmte Programmierschnittstellen-Einsprungpunkte in den Treiber eingehakt, wie z.B. in Schritt **117**. Mit anderen Worten, wenn die Videosystem-Vorrichtungsschnittstelle, z.B. Windows GDI, diese Einsprungpunkte in den Anzeigetreiber ruft, kann das Programm die Gelegenheit ergreifen, die an den Anzeigetreiber übergebenen Parameter zu modifizieren und/oder die von dem Anzeigetreiber zurückgegebenen Werte zu modifizieren.

**[0084]** Durch Einklinken der "Reenable"-Funktion in dem Anzeigetreiber in Schritt **117** kann das Overscan-Balken-Programm in Schritt **119** einen Schirmbereich auf verschiedene Weise zuweisen:

(1) Im Einstellmodus, Schritt **121**, durch Abfangen einer Auflösungs-Änderungsanforderung und Identifizieren der nächsthöheren unterstützten Schirmauflösung und Übergeben dieser höheren Auflösung an den Anzeigetreiber, dann, wenn der Anzeigetreiber die Änderung bestätigt, Abfangen des zurückgegebenen Wertes, der die neue Auflösung widerspiegeln würde, und stattdessen tatsächlich die ursprünglich angeforderte Auflösung zurückgeben. Zum Beispiel fordert die GDI eine Änderung von 640 × 480 Auflösung auf 800 × 600 Auflösung an; das Overscan-Programm fängt die Anforderung ab und modifiziert sie, um den Anzeigetreiber auf die nächste unterstützte Auflösung höher als 800 × 600, z.B. 1024 × 768, zu ändern. Der Anzeigetreiber wird die Schirmauflösung in 1024 × 768 ändern und diese neue Auflösung zurückgeben. Das Overscan-Programm fängt die Rückgabe ab und übergibt stattdessen die ursprüngliche Anforderung, 800 × 600, an die GDI. Der Anzeigetreiber hat einen Speicherbereich von 1024 × 768 zugeteilt und zeigt ihn an. Die GDI und Windows werden den Desktop in einem 800 × 600 Bereich dieser Anzeige anzeigen, wobei Bereiche auf der rechten und unteren Kante des Schirms für das Overscan-Programm verfügbar gelassen werden.

(2) Im geteilten Modus, Schritt **123**, durch Abfangen nur der Rückgabe von dem Anzeigetreiber und Modifizieren des Wertes, um das Verstehen des Betriebssystems der wirklichen Schirmauflösung zu ändern. Zum Beispiel fordert die GDI eine Änderung von 800 × 600 Auflösung auf 1024 × 768 Auflösung an. Das Overscan-Programm fängt die zurückgegebene Bestätigung ab und subtrahiert 32 vor dem Übergeben der Rückgabe an die GDI. Der Anzeigetreiber hat einen Speicherbereich von 1024 × 768 zugeteilt und zeigt ihn an. Die GDI und Windows werden den Desktop in einem 1024 × 736 Bereich dieser Anzeige anzeigen, wobei ein Bereich auf der unteren Kante des Schirms für das Overscan-Balken-Programm verfügbar gelassen wird.

**[0085]** Nach dem Einhaken kann das Overscan-Balken-Programm anzeigen durch:

(1) Verwenden von Standard-API-Aufrufen, um den Balken an einen Offscreen-Puffer auszugeben, wie im nächsten Abschnitt beschrieben, und dann Einhaken des "BitBlit"-Funktions-Einsprungpunktes in den Anzeigetreiber, um die Offset- und Größen-Parameter zu modifizieren, und anschließend die BitBlit zu dem Bereich außerhalb von dem umleiten, von dem die API glaubt, dass er Onscreen ist.

(2) Verwenden von Mechanismen von Primär- und Sekundär-Adressen, oben beschrieben. Das Programm bestimmt die linearen Adressen für die Off-Desktop-Speicherstelle(n), die ihm verfügbar gelassen sind, und kann direkt an diese Speicherstellen zurückgeben.

**[0086]** Phase 2 der Erfindung beginnt mit Malen der neuen Bilder in einen Standard-Offscreen-Puffer, Schritt **118**, wie gewöhnlich in der Technik benutzt, und Sichtbarmachen des Inhalts, Schritt **120**, wie in [Fig. 10](#) beschrieben. Wenn das Programm im "Werkzeugleisten"-Modus ist, Schritt **156**, wird der Offscreen-Puffer in den Standard-Fenster-Client-Raum gemalt, Schritt **166**, und unter Verwendung generischer Fenstersystem-Routinen sichtbar gemacht, Schritt **164**. Andernfalls wird die lineare Fensterpositionsadresse zugeordnet, Schritt **158**, wie in [Fig. 10](#) beschrieben, die vorher erklärt wurde. Sobald der lineare Speicher einer physikalischen Speicheradresse zugeordnet ist, Schritt **142**, kann der Inhalt des Offscreen-Anzeigepuffers direkt in den Videopuffer kopiert werden, Schritt **154** von [Fig. 10](#), oder bezüglich einer Sekundär-Oberfläche gemalt werden.

**[0087]** Die Anwendung der bevorzugten Ausführung enthält eine Standard-Anwendungs-Nachrichtenschleife, Schritt **122**, die System- und Benutzerereignisse verarbeitet. Ein Beispiel einer Verarbeitungsschleife mit minimaler Funktionalität ist in [Fig. 12](#). Hier handhabt die Anwendung einem Minimalsatz von Systemereignissen, z.B. Malen von Anforderungen, Schritt **170**, Systemauflösungsänderungen, Schritt **172**, und Aktivieren/Deaktivieren, Schritt **174**. Hier ist auch, wo Benutzerereignisse, z.B. Tasten- oder Maus-Ereignisse, behandelt werden können, Schritt **184**, erklärt in [Fig. 13](#). Systemmalen-Nachrichten werden durch Malen, wenn angebracht, in den Offscreen-Puffer, Schritt **178**, und Malen des Fenster- oder Anzeigepuffers, Schritt **180**, wenn angebracht, wie früher in [Fig. 10](#) beschrieben, gehandhabt. Systemauflösungsnachrichten werden empfangen, wann immer das System oder der Benutzer die Schirm- oder Farbauflösung ändert. Die Programme setzen alle Register auf die richtigen neuen Werte und ändern dann die Anzeigeauflösung, Schritt **182**, wie früher in [Fig. 9](#) beschrieben, um die modifizierte, neue Auflösung widerzuspiegeln. Benutzernachrichten werden ignoriert, wenn das Programm nicht die aktive Anwendung ist.

**[0088]** [Fig. 13](#) beschreibt ein Verfahren zur Implementierung benutzereingegebener Ereignisse. Bei dieser Ausführung gibt es drei alternative Mechanismen, die benutzt werden, um Cursor- oder Maus-Unterstützung zu implementieren, sodass der Benutzer ein Zeigevorrichtung-Eingabewerkzeug in der Overscan-Bereichs-Benutzerschnittstelle hat.

**[0089]** In dem bevorzugten Mechanismus wird "Cliprect" der GDI modifiziert, um den Anzeigebereich des Overscan-Balkens einzuschließen. Dies verhindert, dass das Betriebssystem den Cursor beschneidet, wenn er sich in den Overscan-Bereich bewegt. Diese Änderung macht nicht unbedingt den Cursor sichtbar oder stellt Ereignis-Rückmeldung bereit, ist aber der erste Schritt.

**[0090]** Einige gegenwärtige Windows-Anwendungen setzen Cliprect fortlaufend zurück. Es ist eine Standard-Programmierprozedur, nach Gebrauch oder Verlust von Eingabefokus zurückzusetzen. Einige Anwendungen verwenden Cliprect, um die Maus auf einen spezifischen Bereich einzuschränken, wie er von der aktiven Anwendung verlangt werden kann. Wann immer die Overscan-Anzeigebalkenschnittstelle den Eingabefokus empfängt, macht sie Cliprect erneut geltend, wobei sie groß genug gemacht wird, damit sich die Maus hinab in den Overscan-Raum bewegt.

**[0091]** Sobald Cliprect erweitert wurde, kann die Maus Nachrichten für das Betriebssystem erzeugen, die Bewegung in dem Erweiterungsbereich widerspiegeln. Die GDI zieht jedoch den Cursor nicht aus dem heraus, was sie als ihre Auflösung versteht, und leitet keine "Out-of-Bounds"-Ereignisnachrichten an eine Anwendung weiter. Das Overscan-Programm benutzt einen V × D-Gerätetreiber und verwandte Rückruf-Funktion, um Hardwaretreiber-Aufrufe bei Ring null zu machen, um die wirklichen physikalischen Deltas oder Änderungen in der Mausposition oder Zustand zu überwachen. Jede Mauspositions- oder Zustandsänderung wird als ein Ereignis an das Programm zurückgegeben, das die Position in dem Menüanzeigebalken grafisch darstellen kann.

**[0092]** Ein alternativer Mechanismus umgeht die Notwendigkeit, Cliprect zu erweitern, um einen Konflikt mit Vorrichtungstreibern zu vermeiden, die Cliprect verwenden, um eine Drehung der virtuellen Anzeige zu ermöglichen. Durch direktes Abfragen der Mauseingabevorrichtung kann das Overscan-Programm "Deltas", Änderungen in Position und Zustand, bestimmen. Wann immer der Cursor die letzte Reihe oder Spalte von Pixeln auf der Standardanzeige berührt, wird er dort durch Setzen von Cliprect auf ein Rechteck, das nur aus dieser letzten Reihe oder Spalte besteht, begrenzt. Eine "virtuelle" Cursor-Position wird aus den von der Eingabevorrichtung verfügbaren Deltas gewonnen. Der wirkliche Cursor ist verborgen, und eine virtuelle Cursor-Darstellung wird explizit in den virtuellen Koordinaten angezeigt, um eine genaue Rückmeldung an den Benutzer zu liefern. Wenn sich die virtuellen Koordinaten von dem Overscan-Bereich zurück auf den Desktop bewegen, wird Cliprect gelöscht, die virtuelle Darstellung entfernt und der wirkliche Cursor auf dem Schirm wiederhergestellt.

**[0093]** Ein dritter alternativer Mechanismus erzeugt ein durchsichtiges Fenster, das sich mit dem wirklichen Windows-Desktop-Anzeigebereich um eine vordefinierte Zahl von Pixeln, z.B. zwei oder vier Pixel, überschneidet. Wenn die Maus in diesen kleinen, durchsichtigen Bereich eintritt, versteckt das Programm den Cursor. Ein Cursor-Bild wird dann in dem Overscan-Balkenbereich in der gleichen X-Koordinate, aber in einer Y-Koordinate, die in den Overscan-Bereich entsprechend versetzt ist, angezeigt. Wenn ein Überschneidungsbereich von zwei Pixeln benutzt wird, verwendet dieses Verfahren eine Körnigkeit von zwei. Diese Nur-API-Lösung liefert folglich nur eine begrenzte Vertikal-Körnigkeit. Dieser alternative Mechanismus stellt sicher, dass alle Implementierungen einen gewissen Grad an Maus-Eingabeunterstützung haben werden, selbst wenn Cliprect- und Eingabevorrichtungstreiber-Lösungen versagen.

**[0094]** [Fig. 7](#) beschreibt den Aufräum-Mechanismus, der ausgeführt wird, wenn das Programm geschlossen wird, Schritt **124**. Die Anzeige wird auf die ursprüngliche Auflösung zurückgesetzt, Schritt **126**, und die CR-Register werden auf ihre ursprünglichen Werte zurückgesetzt, Schritt **128**, und verriegelt, Schritt **130**.

Alternative Beispiele, die nicht Teil der Erfindung sind

1. Verwenden der VESA BIOS Erweiterungen (VBE) anstelle der CRT Controller Register ([Fig. 5](#)), um die lineare Fensterpositionsadresse, Schritt **138**, wenn nötig, zu bestimmen.
2. Verwenden von APIs (Anwendungsprogrammierschnittstellen) **62**, die zur direkten Treiber- und/oder Hardware-Manipulation imstande sind, z.B. Microsoft DirectX und/oder DirectDraw, anstelle der CRT Controller Register und/oder Direktzugriff auf den Anzeigepuffer.
3. Verwenden von APIs (Anwendungsprogrammierschnittstellen) **62**, z.B. Microsoft DirectX und/oder DirectDraw, die zur direkten Treiber und/der Hardware-Manipulation imstande sind, um eine zweite virtuelle Anzeigefläche auf der Primärazeige mit dem gleichen Zweck zu erzeugen, um eine getrennte und ungetrübte grafische Benutzerschnittstelle anzuzeigen.
4. Verwenden von Modifikationen in dem Video-Untersystem des Betriebssystems **63** anstelle der CRT Controller Register und/oder DirectX-Zugriff auf den Anzeigepuffer.
5. Verwenden von Modifikationen in dem Video-Untersystem des Betriebssystems **63**, um eine zweite virtuelle Anzeigefläche auf der Primärazeige mit dem gleichen Zweck zu erzeugen, um eine getrennte und ungetrübte grafische Benutzerschnittstelle anzuzeigen.
6. Einbauen dieser Funktionalität in die wirklichen Videotreiber **64** und/oder Minitreiber. Microsoft Windows liefert Unterstützung für virtuelle Vorrichtungstreiber, die auch direkt mit der Hardware und Treibern verbunden werden könnten. Diese könnten auch eine API einschließen, um Anwendungen mit einer Schnittstelle zu der modifizierten Anzeige bereitzustellen.
7. Einschließen der gleichen Funktionalität, mit oder ohne die VGA-Register, in das BIOS und Bereitstellen einer API, um Anwendungen eine Schnittstelle zu der modifizierten Hardware zu erlauben.
8. Einschließen der gleichen Funktionalität in Hardware-Vorrichtungen, z.B. der Monitor selbst, mit Hardware- und/oder Software-Schnittstellen zu der CPU.

**[0095]** Zusammengefasst, der visuelle Anzeigebereich wird herkömmlich durch die Werte definiert, die in den CRT-Registern unterhalten werden und dem Treiber zugänglich sind. Der normalerweise angezeigte Bereich wird durch VGA-Standards und später durch SVGA-Standards so definiert, dass er eine voreingestellte Zahl von Modi ist, wobei jeder Modus eine bestimmte Anzeigeauflösung enthält, die den Bereich der Anzeige spezifiziert, in dem der Desktop angezeigt werden kann.

**[0096]** Der Desktop kann nur in diesem Bereich angezeigt werden, weil Windows den Videospeicher nicht direkt liest/schreibt, und stattdessen Programmierschnittstellenaufrufe an den Videotreiber verwendet. Und der Videotreiber liest/schreibt einfach unter Verwendung einer Adresse, die in dem Videospeicher liegt. Der Wert, den dieser Mechanismus realisieren muss, ist somit das, was die Videokarte und der Treiber als zum Malen verfügbar geltend machen. Dieser Wert wird von den Registern abgefragt, mit spezifischen Beträgen modifiziert und wieder auf die Karte geschrieben. Anschließend ändert die vorliegende Erfindung den Bereich des beschreibbaren, sichtbaren Anzeigebereichs, ohne die Anzeigeschnittstelle des Betriebssystems über die Änderung zu informieren.

**[0097]** Diese Erfindung ändert nicht unbedingt die CRTCs, um nur den Boden hinzuzufügen. Vorzugsweise wird auch die Spitze ein wenig nach oben bewegt. Dies hält die Anzeige in dem Overscan-Bereich zentriert. Anstatt nur dem Boden 32 Abtastzeilen hinzuzufügen, wird die Spitze des Anzeigebereichs um 16 Zeilen nach oben bewegt.

**[0098]** Diese Erfindung hängt nicht allein von dem Vermögen ab, die CRTCs zu ändern, um den sichtbaren Anzeigebereich zu modifizieren. Alternative Mechanismen definieren andere Verfahren zum Erzeugen von und Zugreifen auf sichtbare Bereiche des Schirms, die außerhalb der Abmessungen des Desktops liegen, auf die die Anzeigeschnittstelle des Betriebssystems zugreift.

**[0099]** Aus einer Betrachtung der Beschreibungen, Zeichnungen und Ansprüche werden andere Ausführungen und Variationen der Erfindung für eine in der Computerwissenschaft erfahrene Person ersichtlich sein.

**[0100]** Im Besonderen, die Sekundär-GUI kann sich in Bereichen befinden, die normalerweise nicht als der herkömmliche Overscan-Bereich angesehen werden. Zum Beispiel kann die Sekundär-GUI in einem kleinen Quadrat genau in der Mitte der normalen Anzeige positioniert werden, um eine von dem einzelnen System und

der Anwendung benötigten Service bereitzustellen. In der Tat können die Verfahren des Lesens und Umschreibens von Schirmanzeigeinformation innerhalb des Umfangs der Erfindung benutzt werden, um die Primär-GUI-Information oder Teile davon in einem zusätzlichen Speicher zu bewahren, und selektiv auf einer zeitgesteuerten oder anderen Basis einen Teil der Primär-GUI durch die Sekundär-GUI zu ersetzen.

**[0101]** Als ein einfaches Beispiel kann ein Sicherheitssystem die Fähigkeit erfordern, dem Benutzer Information ohne Rücksicht auf den Zustand des Computersystem anzuzeigen, und/oder erfordern, dass der Benutzer eine Auswahl trifft, z.B. durch Klicken auf "911?" nach Hilfe ruft. Die vorliegende Erfindung könnte einen Videoanzeigepuffer bereitstellen, in dem ein Teil der Primär-GUI-Schnittstelle dauernd aufgezeichnet und in einer Sekundär-GUI, z.B. in der Mitte des Schirms, angezeigt wurde. Unter gefahrlosen Bedingungen würde die Sekundär-GUI dann effektiv unsichtbar sein, wobei der Benutzer nichts außer der Primär-GUI zur Kenntnis nehmen würde.

**[0102]** Unter den geeigneten Gefahrenbedingungen könnte ein Alarmmonitor die Sekundär-GUI veranlassen, dem Benutzer die "911?" zu präsentieren, durch Überschreiben der in dem Sekundär-GUI-Speicher gespeicherten Kopie der Primäranzeige. Alternativ kann eine Datenbank von Fotografien gespeichert werden und eine als Reaktion auf einen ankommenden Telefonanruf aufgerufen werden, in dem die Anrufer-ID eine mit einem Datenbank-Fotoeintrag verbundene Telefonnummer identifizierte.

**[0103]** Grundsätzlich kann die vorliegende Erfindung eine oder mehrere Sekundär-Benutzerschnittstellen bereitstellen, die immer dann hilfreich sein können, wenn es bequemer oder wünschenswert ist, einen Teil des Gesamtanzeige, entweder außerhalb der Primäranzeige in einem unbenutzten Bereich, z.B. Overscan, oder auch in einem Abschnitt der Primär-GUI direkt oder durch Zeitmultiplexierung, direkt durch Kommunikation mit dem Videospeicher durch Umgehen wenigstens eines Teils des Videospeichers zu steuern, um einen neuen Videospeicher zu erzeugen. Mit anderen Worten, die vorliegende Erfindung kann eine oder mehrere Sekundär-Benutzerschnittstellen außerhalb der Steuerung des Systems, z.B. des Betriebssystems, das die Primär-GUI steuert, bereitstellen.

**[0104]** Zusätzliche Benutzerschnittstellen können für eine Vielfalt verschiedener Zwecke verwendet werden. Zum Beispiel kann eine Sekundär-Benutzerschnittstelle verwendet werden, um einen gleichzeitigen Zugriff auf das Internet, voll bewegtes Video und einen Konferenzkanal bereitzustellen. Eine Sekundär-Benutzerschnittstelle kann einem lokalen Netzwerk zugeordnet werden oder mehrfache Sekundär-Benutzerschnittstellen können gleichzeitigen Zugang und Daten für ein oder mehrere Netzwerke bereitstellen, mit denen ein bestimmter Computer verbunden sein kann.

**[0105]** Nachdem nun die Erfindung nach den Erfordernissen der Patentstatute beschrieben wurde, werden die Fachleute in dieser Technik verstehen, wie Änderungen und Modifikationen in der vorliegenden Erfindung vorzunehmen sind, um ihre spezifischen Anforderungen oder Bedingungen zu erfüllen. Solche Änderungen und Modifikationen können vorgenommen werden, ohne von dem Umfang der Erfindung, wie in den folgenden Ansprüchen dargelegt, abzuweichen.

### Patentansprüche

1. Verfahren zum Anzeigen von Daten auf einem Video-Anzeigesystem in Verbindung mit einer Computer-Betriebssystem-Benutzerschnittstelle, die wenigstens einen Teil eines ersten Anzeigebereiches einnimmt, wobei der erste Anzeigebereich über die Computer-Betriebssystem-Benutzerschnittstelle zugänglich ist und erste Bilddaten anzeigt, und das Video-Anzeigesystem einen adressierbaren Gesamt-Anzeigebereich hat, zu dem der erste Anzeigebereich als ein Teil gehört, und das Verfahren umfasst:

Modifizieren des adressierbaren Gesamt-Anzeigebereiches des Video-Anzeigesystems, indem die Anzeigeparameter des Video-Anzeigesystems so eingestellt werden, dass es einen zweiten Anzeigebereich einschließt, der über die Computer-Betriebssystem-Benutzerschnittstelle nicht zugänglich ist;

Zuweisen des modifizierten ansteuerbaren Gesamt-Anzeigebereiches und entsprechender Speicherressourcen des Video-Anzeigesystems zwischen dem ersten Anzeigebereich und dem zweiten Anzeigebereich; und Schreiben zweiter Bilddaten in die Speicherressource des zweiten Anzeigebereiches, so dass die zweiten Bilddaten auf dem Video-Anzeigesystem zusammen mit den ersten Anzeigedaten angezeigt werden, die in Verbindung mit der Benutzerschnittstelle des Computer-Betriebssystems angezeigt werden.

2. Verfahren nach Anspruch 1, wobei durch das Zuweisen des modifizierten ansteuerbaren Gesamt-Anzeigebereiches die Größe des Teils, der für die Betriebssystem-Benutzerschnittstelle zugänglich ist, relativ zur Größe des ansteuerbaren Gesamt-Anzeigebereiches verringert wird.

3. Verfahren nach Anspruch 2, wobei der modifizierte ansteuerbare Gesamt-Anzeigebereich größer ist als vor der Modifizierung.
4. Verfahren nach Anspruch 1, wobei der modifizierte ansteuerbare Gesamt-Anzeigebereich größer ist als vor der Modifizierung und durch das Zuweisen des modifizierten ansteuerbaren Gesamt-Anzeigebereiches die Größe des ersten Anzeigebereiches vergrößert wird.
5. Verfahren nach Anspruch 4, wobei die vergrößerte Größe des ersten Anzeigebereiches keine Standard-Videoauflösungsmodus-Größe ist.
6. Verfahren nach Anspruch 1, wobei durch das Einstellen der Anzeigeparameter des Video-Anzeigesystems die Anzahl ansteuerbarer Pixel in wenigstens einer Dimension des ansteuerbaren Gesamt-Anzeigebereiches auf weniger als oder genauso viel wie die maximale Anzahl von Pixeln erhöht wird, die effektiv durch das Video-Anzeigesystem angezeigt werden können.
7. Verfahren nach Anspruch 6, wobei durch das Einstellen der Anzeigeparameter des Video-Anzeigesystems die Anzahl ansteuerbarer Pixel erhöht wird, indem Pixel in einem Bildschirmrahmenbereich des Video-Anzeigesystems angesteuert werden.
8. Verfahren nach Anspruch 7, wobei die zweiten Bilddaten in wenigstens einem Teil von Pixeln in dem Bildschirmrahmenbereich angezeigt werden und Bilddaten eines beweglichen Zeigers einschließen, der sich in Zusammenhang mit Benutzereingabe bewegt.
9. Verfahren nach Anspruch 8, wobei der Zeiger eine dazugehörige Spitze aufweist, die außerhalb eines Cursor-Aktivierungspunktes angeordnet ist, der zu der Spitze gehört, und der Cursor-Aktivierungspunkt innerhalb des ersten Anzeigebereiches bleibt, während der Zeiger in dem zweiten Anzeigebereich angezeigt wird.
10. Verfahren nach Anspruch 6, wobei die eingestellten Anzeigeparameter Steuerparameter für eine Steuerung einer Kathodenstrahlröhren-Anzeige sind.
11. Verfahren nach Anspruch 1, wobei der modifizierte ansteuerbare Gesamt-Anzeigebereich so vergrößert wird, dass er einen zweiten Anzeigebereich einschließt, indem die Anzahl ansteuerbarer Pixel in wenigstens einer Dimension des ansteuerbaren Gesamt-Anzeigebereiches erhöht wird.
12. Verfahren nach Anspruch 11, wobei die Dimension, in der die Anzahl ansteuerbarer Pixel erhöht wird, vertikal ist und die Daten unter der Betriebssystem-Benutzerschnittstelle angezeigt werden.
13. Verfahren nach Anspruch 11, wobei die Dimension, in der die Anzahl adressierbarer Pixel erhöht wird, vertikal ist, und die Daten über der Betriebssystem-Benutzerschnittstelle angezeigt werden.
14. Verfahren nach Anspruch 11, wobei die Abmessung, in der die Anzahl ansteuerbarer Pixel erhöht wird, horizontal ist und die Daten links von der Betriebssystem-Benutzerschnittstelle angezeigt werden.
15. Verfahren nach Anspruch 11, wobei die Dimension, in der die Anzahl ansteuerbarer Pixel erhöht wird, horizontal ist und die Daten rechts von der Betriebssystem-Benutzerschnittstelle angezeigt werden.
16. Verfahren nach Anspruch 11, wobei die Dimension, in der die Anzahl ansteuerbarer Pixel erhöht wird, sowohl horizontal als auch vertikal ist und die Daten auf einer vertikalen Seite der Betriebssystem-Benutzerschnittstelle sowie auf einer horizontalen Seite der Betriebssystem-Benutzerschnittstelle angezeigt werden.
17. Verfahren nach Anspruch 1, wobei, wenn der ansteuerbare Gesamt-Anzeigebereich des Video-Anzeigesystems modifiziert wird, so dass er den zweiten Anzeigebereich einschließt, nachdem die Parameter eingestellt werden, der ansteuerbare Gesamt-Anzeigebereich auf eine Standardauflösung vergrößert wird, die von dem Video-Anzeigesystem unterstützt wird.
18. Verfahren nach Anspruch 1, wobei das Einstellen der Anzeigeparameter und das Zuweisen des modifizierten ansteuerbaren Gesamt-Anzeigebereiches des Weiteren umfasst:  
Empfangen einer Aufforderung von dem Betriebssystem, einen ersten, höheren Videoauflösungsmodus zu verwenden;  
Auffordern des Video-Anzeigesystems, einen zweiten höheren Videoauflösungsmodus zu verwenden, der hö-

her ist als der erste höhere Videoauflösungsmodus, um so die Größe des adressierbaren Gesamt-Anzeigebereiches zu vergrößern;

Zuweisen eines Teils des ansteuerbaren Gesamt-Anzeigebereiches, der dem ersten höheren Videoauflösungsmodus entspricht, zu dem ersten Anzeigebereich; und

Zuweisen eines Teils des ansteuerbaren Gesamt-Anzeigebereiches zwischen dem ersten höheren Videoauflösungsmodus und dem zweiten höheren Videoauflösungsmodus zu dem zweiten Anzeigebereich zum Anzeigen der zweiten Bilddaten.

19. Verfahren nach Anspruch 1, wobei das Einstellen der Anzeigeparameter und das Zuweisen des modifizierten ansteuerbaren Gesamt-Anzeigebereiches des Weiteren umfasst:

Empfangen einer Aufforderung von dem Betriebssystem, einen höheren Videoauflösungsmodus zu verwenden, der höher ist als ein aktueller Auflösungsmodus, so dass der ansteuerbare Gesamt-Anzeigebereich auf den höheren Videoauflösungsmodus vergrößert wird;

Zuweisen eines Teils des ansteuerbaren Gesamt-Anzeigebereiches, der dem aktuellen Auflösungsmodus entspricht, zu dem ersten Anzeigebereich; und

Zuweisen des vergrößerten ansteuerbaren Gesamt-Anzeigebereiches zwischen dem höheren Videoauflösungsmodus und dem aktuellen Videoauflösungsmodus zu dem zweiten Anzeigebereich zum Anzeigen der zweiten Bilddaten.

20. Verfahren nach Anspruch 1, wobei das Einstellen der Anzeigeparameter und das Zuweisen des modifizierten ansteuerbaren Gesamt-Anzeigebereiches des Weiteren umfasst:

Empfangen einer Aufforderung von dem Betriebssystem, einen ersten höheren Videoauflösungsmodus zu verwenden;

Auffordern des Video-Anzeigesystems, den ersten höheren Videoauflösungsmodus zu verwenden, so dass sich der Bereich des ansteuerbaren Gesamt-Anzeigebereiches vergrößert;

Zuweisen eines Teils des vergrößerten Bereiches des ansteuerbaren Gesamt-Anzeigebereiches zu dem ersten Anzeigebereich; und

Zuweisen des restlichen Teils des vergrößerten Bereiches zu dem zweiten Anzeigebereich zum Anzeigen der zweiten Bilddaten.

21. Verfahren nach Anspruch 1, wobei wenigstens ein Teil des zweiten Bildes zusammen mit dem ersten Bild, das in Verbindung mit der Betriebssystem-Benutzerschnittstelle angezeigt wird, so angezeigt wird, dass unterbunden wird, dass die Betriebssystem-Benutzerschnittstelle die Daten des Teils des zweiten Bildes überschreibt.

22. Anzeige-Steuerung, die die Anzeige einer sekundären Benutzerschnittstelle auf einem Video-Anzeigesystem zusammen mit der Anzeige einer primären Benutzerschnittstelle ermöglicht, wobei die primäre Benutzerschnittstelle durch ein separat gesteuertes Programm auf einem ersten Anzeigebereich des Video-Anzeigesystems dargestellt wird und das Video-Anzeigesystem einen ansteuerbaren Gesamt-Anzeigebereich hat, und die umfasst:

eine Anzeige-Einstelleinrichtung, die den ansteuerbaren Gesamt-Anzeigebereich des Video-Anzeigesystems modifiziert, indem sie die Anzeigeparameter des Video-Anzeigesystems so einstellt, dass es einen zweiten Anzeigebereich einschließt, der für das separat gesteuerte Programm nicht zugänglich ist;

eine Anzeige-Zuweisungseinrichtung, die den modifizierten ansteuerbaren Gesamt-Anzeigebereich und entsprechende Speicherressourcen des Video-Anzeigesystems zwischen dem ersten Anzeigebereich und dem zweiten Anzeigebereich zuweist; und

einen Anzeige-Übertragungsmechanismus, der Bilddaten für die sekundäre Benutzerschnittstelle in die Speicherressource des zweiten Anzeigebereiches schreibt, so dass die sekundäre Benutzerschnittstelle zusammen mit der primären Benutzerschnittstelle auf dem Video-Anzeigesystem angezeigt wird.

23. Steuerung nach Anspruch 22, wobei die Anzeige-Zuweisungseinrichtung die Größe eines Teils des ansteuerbaren Gesamt-Anzeigebereiches, der für das separat gesteuerte Programm zugänglich ist, relativ zu der Größe des ansteuerbaren Gesamt-Anzeigebereiches verkleinert.

24. Steuerung nach Anspruch 23, wobei die Anzeige-Einstelleinrichtung den ansteuerbaren Gesamt-Anzeigebereich vergrößert.

25. Steuerung nach Anspruch 22, wobei die Anzeige-Einstelleinrichtung den ansteuerbaren Gesamt-Anzeigebereich vergrößert und die Anzeige-Zuweisungseinrichtung die Größe des ersten Anzeigebereiches vergrößert.

26. Steuerung nach Anspruch 25, wobei die vergrößerte Größe des ersten Anzeigebereiches keine Standard-Videoauflösungsmodus-Größe ist.
27. Steuerung nach Anspruch 22, wobei durch das Einstellen der Anzeigeparameter des Video-Anzeigesystems die Anzahl ansteuerbarer Pixel in wenigstens einer Dimension des ansteuerbaren Gesamt-Anzeigebereiches auf weniger als oder genauso viel wie die maximale Anzahl von Pixel erhöht wird, die effektiv on dem Video-Anzeigesystem angezeigt werden können.
28. Steuerung nach Anspruch 27, wobei die Anzeige-Einstelleinrichtung die Anzeigeparameter des Video-Anzeigesystems einstellt, indem Pixel in einem Bildschirmrahmenbereich des Video-Anzeigesystems angesteuert werden.
29. Steuerung nach Anspruch 28, wobei der Anzeige-Übertragungsmechanismus die sekundäre Benutzerschnittstelle in wenigstens einem Teil von Pixeln in dem Bildschirmrahmenbereich anzeigt und einen beweglichen Zeiger anzeigt, der sich in Zusammenhang mit Benutzereingabe bewegt.
30. Steuerung nach Anspruch 29, wobei der bewegliche Zeiger eine dazugehörige Spitze aufweist, die außerhalb eines Cursor-Aktivierungspunktes angeordnet ist, der mit der Spitze verbunden ist, und der Cursor-Aktivierungspunkt innerhalb des ersten Anzeigebereiches bleibt, während der Zeiger in dem Anzeigebereich angezeigt wird, der mit der sekundären Benutzerschnittstelle verbunden ist.
31. Steuerung nach Anspruch 27, wobei die Anzeigeparameter, die durch die Anzeige-Einstelleinrichtung eingestellt werden, Steuerparameter für eine Steuerung einer Kathodenstrahlröhren-Anzeige sind.
32. Steuerung nach Anspruch 22, wobei die Anzeige-Einstelleinrichtung den ansteuerbaren Gesamt-Anzeigebereich so vergrößert, dass er einen zweiten Anzeigebereich anschließt, indem die Anzahl ansteuerbarer Pixel in wenigstens einer Dimension des ansteuerbaren Gesamt-Anzeigebereiches erhöht wird.
33. Steuerung nach Anspruch 32, wobei die Dimension, in der die Anzahl ansteuerbarer Pixel erhöht wird, vertikal ist und die sekundäre Benutzerschnittstelle unter der primären Benutzerschnittstelle angezeigt wird.
34. Steuerung nach Anspruch 32, wobei die Dimension, in der die Anzahl ansteuerbarer Pixel erhöht wird, vertikal ist und die sekundäre Benutzerschnittstelle über der primären Benutzerschnittstelle angezeigt wird.
35. Steuerung nach Anspruch 32, wobei die Dimension, in der die Anzahl ansteuerbarer Pixel erhöht wird, horizontal ist und die sekundäre Benutzerschnittstelle links von der primären Benutzerschnittstelle angezeigt wird.
36. Steuerung nach Anspruch 32, wobei die Dimension, in der die Anzahl ansteuerbarer Pixel erhöht wird, horizontal ist und die sekundäre Benutzerschnittstelle rechts von der primären Benutzerschnittstelle angezeigt wird.
37. Steuerung nach Anspruch 32, wobei die Dimension, in der die Anzahl ansteuerbarer Pixel erhöht wird, sowohl horizontal als auch vertikal ist und die sekundäre Benutzerschnittstelle auf einer vertikalen Seite der primären Benutzerschnittstelle sowie auf einer horizontalen Seite der primären Benutzerschnittstelle angezeigt wird.
38. System nach Anspruch 22, wobei die Anzeige-Einstelleinrichtung den ansteuerbaren Gesamt-Anzeigebereich so modifiziert, dass er den zweiten Anzeigebereich einschließt, indem die Anzeigeparameter so eingestellt werden, dass der ansteuerbare Gesamt-Anzeigebereich auf eine Standardauflösung vergrößert wird, die von dem Video-Anzeigesystem unterstützt wird.
39. Steuerung nach Anspruch 22, wobei die Anzeige-Einstelleinrichtung und die Anzeige-Zuweisungseinrichtung des Weiteren umfassen:  
einen Hooking-Mechanismus, der:  
eine Aufforderung von dem separat gesteuerten Programm empfängt, einen ersten höheren Videoauflösungsmodus zu verwenden;  
das Video-Anzeigesystem auffordert, einen zweiten höheren Videoauflösungsmodus zu verwenden, der höher ist als der erste höhere Videoauflösungsmodus, um so die Größe des ansteuerbaren Gesamt-Anzeigebereiches zu vergrößern;

dem ersten Anzeigebereich einen Teil des ansteuerbaren Gesamt-Anzeigebereiches zuweist, der dem ersten höheren Auflösungsmodus entspricht; und  
dem zweiten Anzeigebereich einen Teil des ansteuerbaren Gesamt-Anzeigebereiches zwischen dem ersten höheren Videoauflösungsmodus und dem zweiten höheren Videoauflösungsmodus zuweist.

40. Steuerung nach Anspruch 22, wobei die Anzeige-Einstelleinrichtung und die Anzeige-Zuweisungseinrichtung des Weiteren umfassen:  
einen Hooking-Mechanismus, der:  
eine Aufforderung von dem separat gesteuerten Programm empfängt, einen höheren Videoauflösungsmodus zu verwenden, der höher ist als ein aktueller Auflösungsmodus, um so die Größe des ansteuerbaren Gesamt-Anzeigebereiches zu vergrößern;  
dem ersten Anzeigebereich einen Teil des ansteuerbaren Gesamt-Anzeigebereiches zuweist, der dem aktuellen Auflösungsmodus entspricht; und  
dem zweiten Anzeigebereich einen Teil des ansteuerbaren Gesamt-Anzeigebereiches zwischen dem höheren Videoauflösungsmodus und dem aktuellen Videoauflösungsmodus zuweist.

41. Steuerung nach Anspruch 22, wobei die Anzeige-Einstelleinrichtung und die Anzeige-Zuweisungseinrichtung des Weiteren umfassen:  
einen Hooking-Mechanismus, der:  
eine Aufforderung von dem separat gesteuerten Programm empfängt, einen ersten höheren Videoauflösungsmodus zu verwenden;  
das Video-Anzeigesystem auffordert, den ersten höheren Videoauflösungsmodus zu verwenden, so dass der ansteuerbare Gesamt-Anzeigebereich vergrößert wird;  
dem ersten Anzeigebereich einen Teil der Vergrößerung des ansteuerbaren Gesamt-Anzeigebereichs zuweist; und  
dem zweiten Anzeigebereich einen verbleibenden Teil der Vergrößerung des ansteuerbaren Gesamt-Anzeigebereiches zuweist.

42. Steuerung nach Anspruch 22 wobei der Anzeige-Übertragungsmechanismus wenigstens einen Teil der sekundären Benutzerschnittstelle zusammen mit der primären Benutzerschnittstelle so anzeigt, dass unterbunden wird, dass die primäre Benutzerschnittstelle den Teil der sekundären Benutzerschnittstelle überschreibt.

43. Steuerung nach Anspruch 22, wobei die Anzeige-Einstelleinrichtung die Anzeigeparameter des Video-Anzeigesystems einstellt, indem sie Funktionsaufrufe an Ansteuer-Software des Video-Anzeigesystems durchführt.

44. Computerlesbares Speichermedium, das Befehle enthält, mit denen ein Computer-Prozessor zum Anzeigen einer sekundären Benutzerschnittstelle auf einem Video-Anzeigesystem in Verbindung mit der Anzeige einer primären Benutzerschnittstelle eines separat gesteuerten Programms auf einem ersten Anzeigebereich des Video-Anzeigesystems, wobei das Video-Anzeigesystem einen ansteuerbaren Gesamt-Anzeigebereich hat, zu dem der erste Anzeigebereich als ein Teil gehört, gesteuert wird, indem:  
der ansteuerbare Gesamt-Anzeigebereich des Video-Anzeigesystems modifiziert wird, indem die Anzeigeparameter des Video-Anzeigesystems so eingestellt werden, dass sie einen zweiten Anzeigebereich einschließen, der für das separat gesteuerte Programm nicht zugänglich ist;  
der modifizierte ansteuerbare Gesamt-Anzeigebereich und entsprechende Speicherressourcen des Video-Anzeigesystems zwischen dem ersten Anzeigebereich und dem zweiten Anzeigebereich zugewiesen werden;  
Bilddaten der sekundären Benutzerschnittstelle in die zweite Speicherressource für den zweiten Anzeigebereich geschrieben werden.

45. Computerlesbares Speichermedium nach Anspruch 44, wobei durch das Zuweisen des modifizierten ansteuerbaren Gesamt-Anzeigebereiches die Größe des Teils, der für das separat gesteuerte Programm zugänglich ist, relativ zu der Größe des ansteuerbaren Gesamt-Anzeigebereiches verringert wird.

46. Computerlesbares Speichermedium nach Anspruch 45, wobei der modifizierte ansteuerbare Gesamt-Anzeigebereich vergrößert wird.

47. Computerlesbares Speichermedium nach Anspruch 44, wobei der modifizierte ansteuerbare Gesamt-Anzeigebereich größer ist als vor der Modifizierung und durch das Zuweisen des modifizierten ansteuerbaren Gesamt-Anzeigebereiches die Größe des ersten Anzeigebereiches vergrößert wird.

48. Computerlesbares Speichermedium nach Anspruch 47, wobei die vergrößerte Größe des ersten Anzeigebereiches keine Standard-Videoauflösungsmodus-Größe ist.

49. Computerlesbares Speichermedium nach Anspruch 44, wobei durch das Einstellen der Parameter des Video-Anzeigesystems die Anzahl ansteuerbarer Pixel in wenigstens einer Dimension des ansteuerbaren Gesamt-Anzeigebereiches auf weniger oder genauso viel wie die maximale Anzahl von Pixeln erhöht wird, die effektiv durch das Video-Anzeigesystem angezeigt werden können.

50. Computerlesbares Speichermedium nach Anspruch 49, wobei durch das Einstellen der Anzeigeparameter des Video-Anzeigesystems Pixel in einem Bildschirmrahmenbereich des Video-Anzeigesystems angesteuert werden.

51. Computerlesbares Speichermedium nach Anspruch 44, wobei die sekundäre Benutzerschnittstelle in wenigstens einem Teil von Pixeln in dem Bildschirmrahmenbereich angezeigt wird und einen beweglichen Zeiger einschließt, der sich im Zusammenhang mit Benutzereingabe bewegt.

52. Computerlesbares Speichermedium nach Anspruch 51, wobei der Zeiger eine dazugehörige Spitze hat, die außerhalb eines Cursor-Aktivierungspunktes angeordnet ist, der zu der Spitze gehört, und der Cursor-Aktivierungspunkt innerhalb des ersten Anzeigebereiches bleibt, während der Zeiger innerhalb der zweiten Benutzerschnittstelle angezeigt wird.

53. Computerlesbares Speichermedium nach Anspruch 44, wobei, wenn der ansteuerbare Gesamt-Anzeigebereich des Video-Anzeigesystems so modifiziert wird, dass er den zweiten Anzeigebereich einschließt, indem die Parameter eingestellt werden, der ansteuerbare Gesamt-Anzeigebereich auf eine Standardauflösung vergrößert wird, die von dem Video-Anzeigesystem unterstützt wird.

54. Computerlesbares Speichermedium nach Anspruch 44, wobei das Einstellen der Anzeigeparameter und das Zuweisen des modifizierten ansteuerbaren Gesamt-Anzeigebereiches durchgeführt wird, indem: eine Aufforderung von der primären Benutzerschnittstelle empfangen wird; das Video-Anzeigesystem aufgefordert wird, einen anderen Videoauflösungsmodus zu verwenden, um so die Größe des ansteuerbaren Gesamt-Anzeigebereiches zu modifizieren; und der modifizierte ansteuerbare Gesamt-Anzeigebereich zwischen dem ersten Anzeigebereich und dem zweiten Anzeigebereich zugewiesen wird.

55. Computerlesbares Speichermedium nach Anspruch 44, wobei wenigstens ein Teil der sekundären Benutzerschnittstelle zusammen mit der primären Benutzerschnittstelle so angezeigt wird, dass unterbunden wird, dass die primäre Benutzerschnittstelle den Teil der sekundären Benutzerschnittstelle überschreibt.

Es folgen 12 Blatt Zeichnungen

*FIG. 1*  
STAND DER TECHNIK

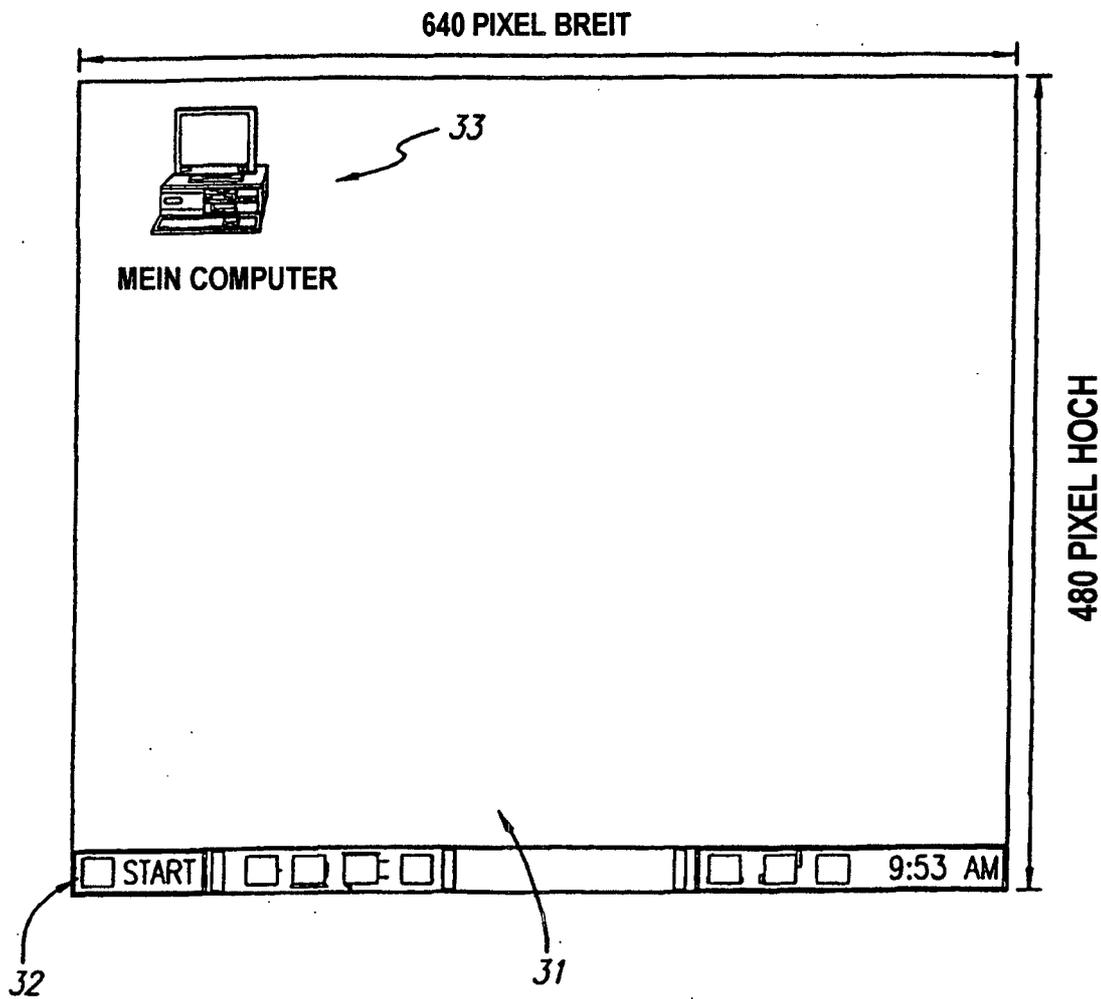


FIG. 2

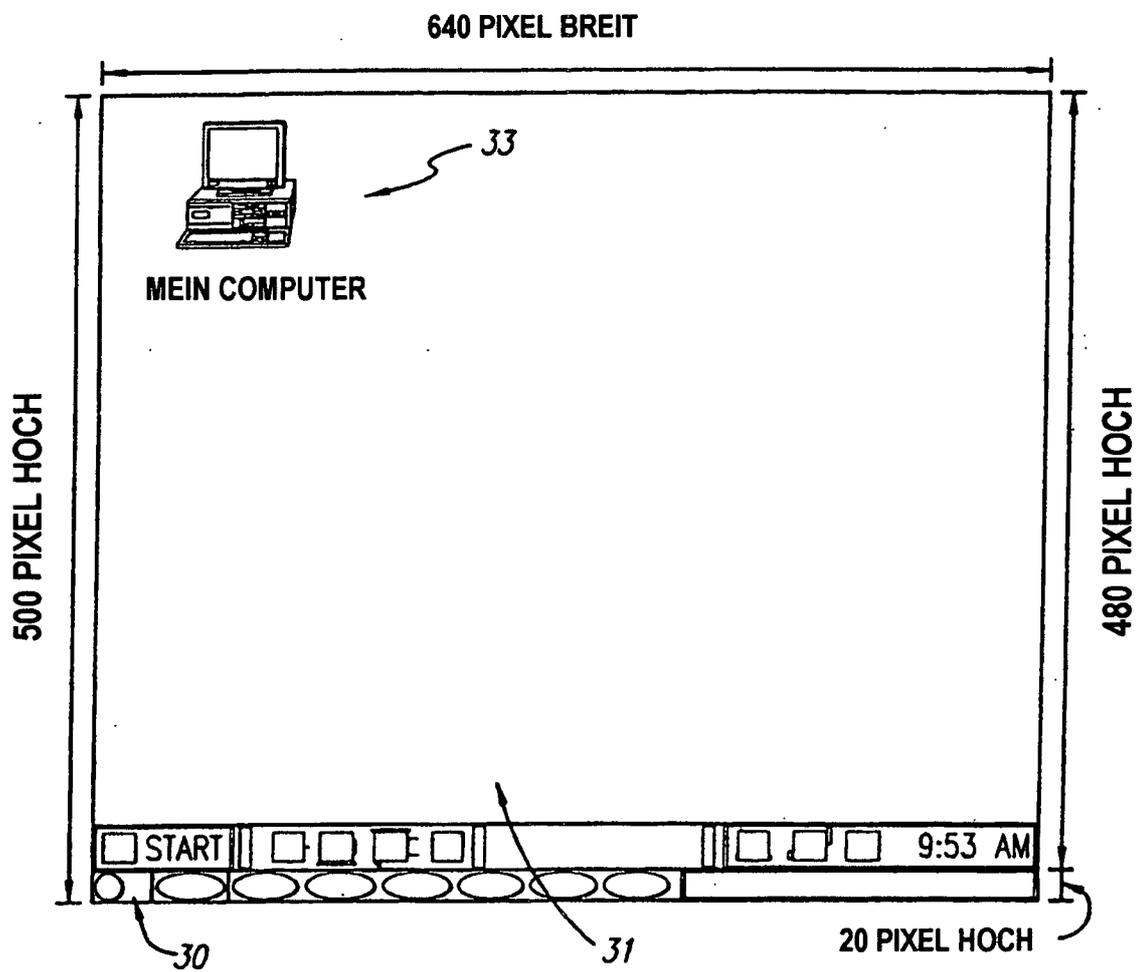


FIG. 3

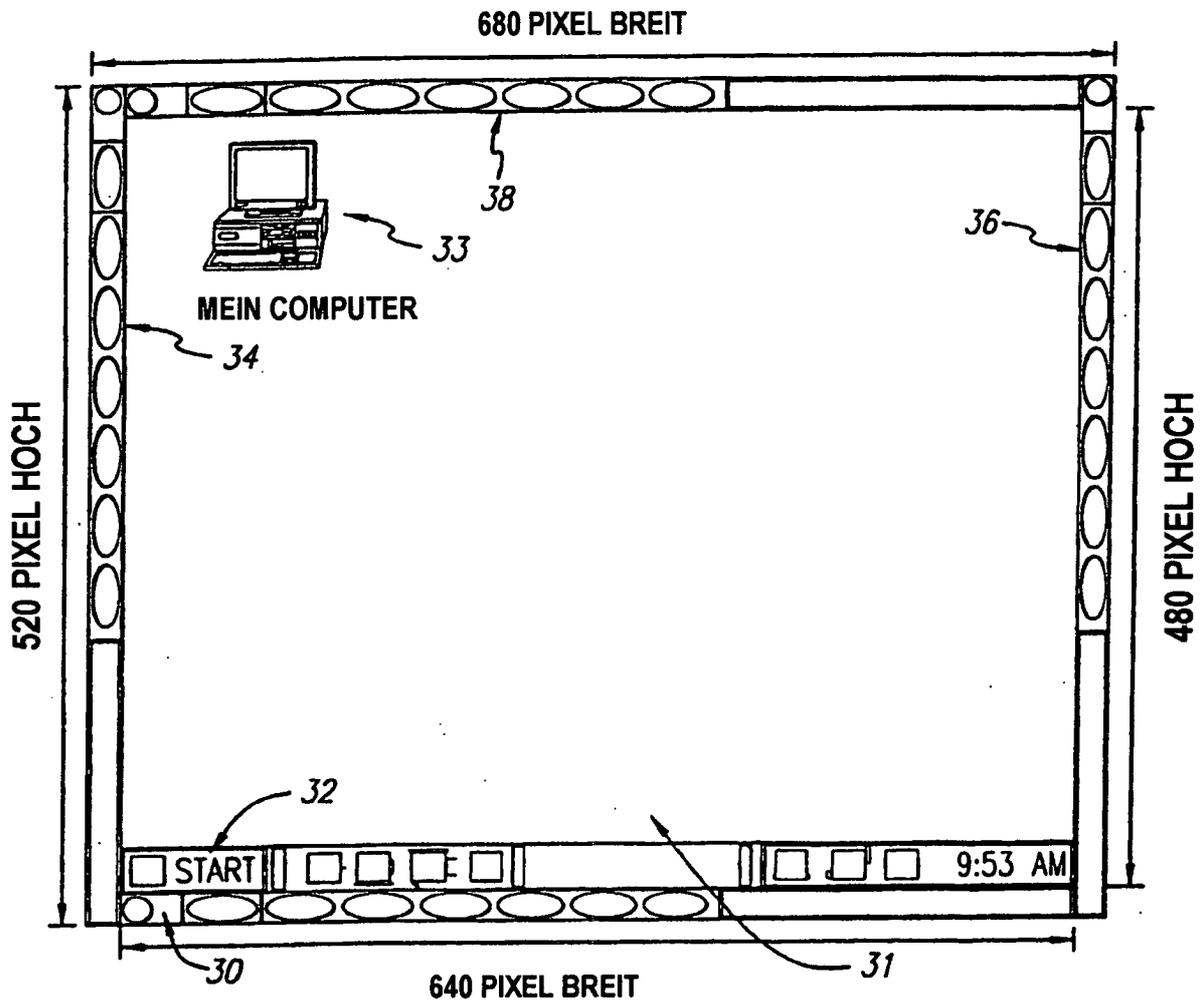


FIG. 4

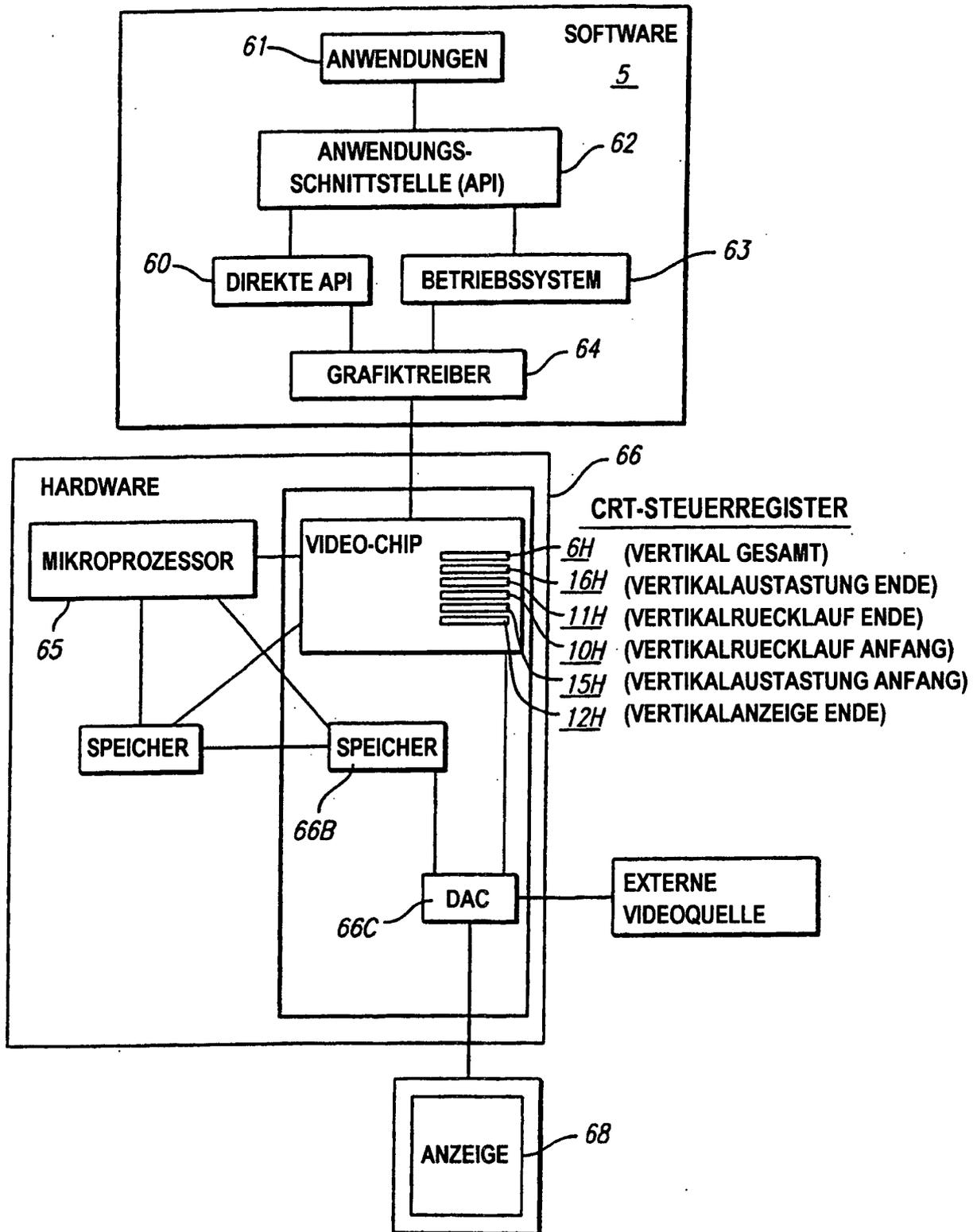


FIG. 5

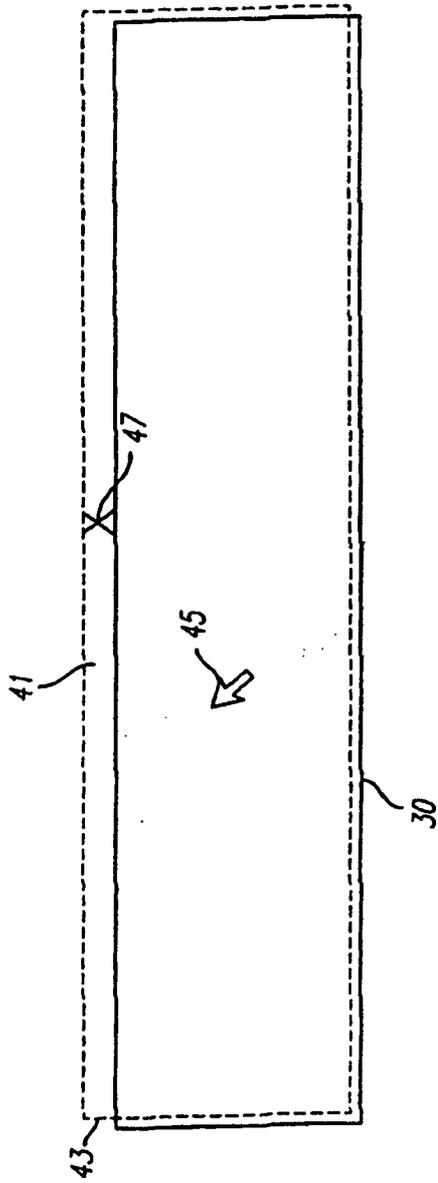


FIG. 6

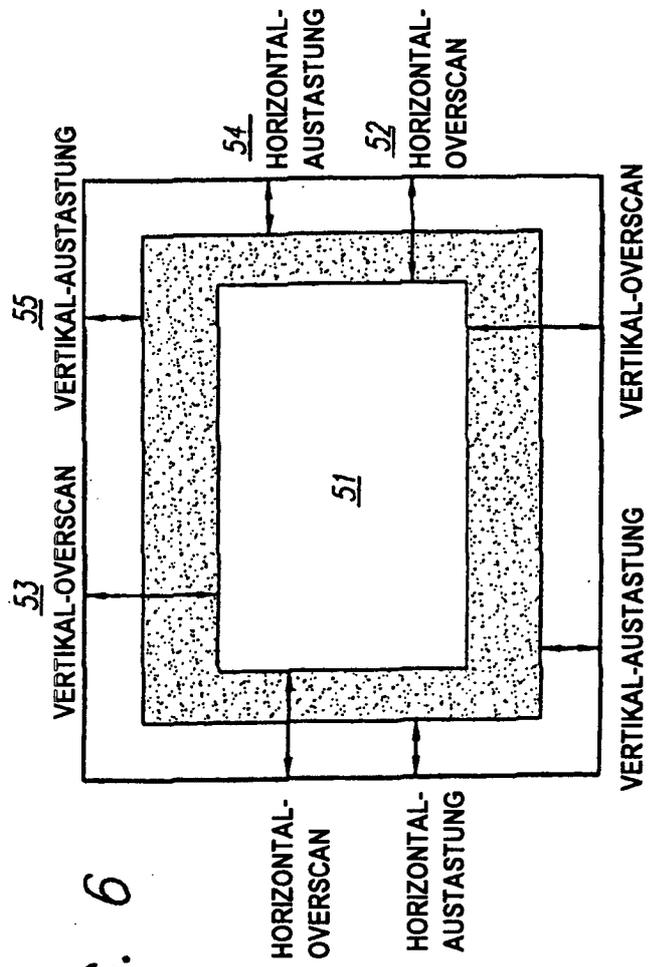


FIG. 7

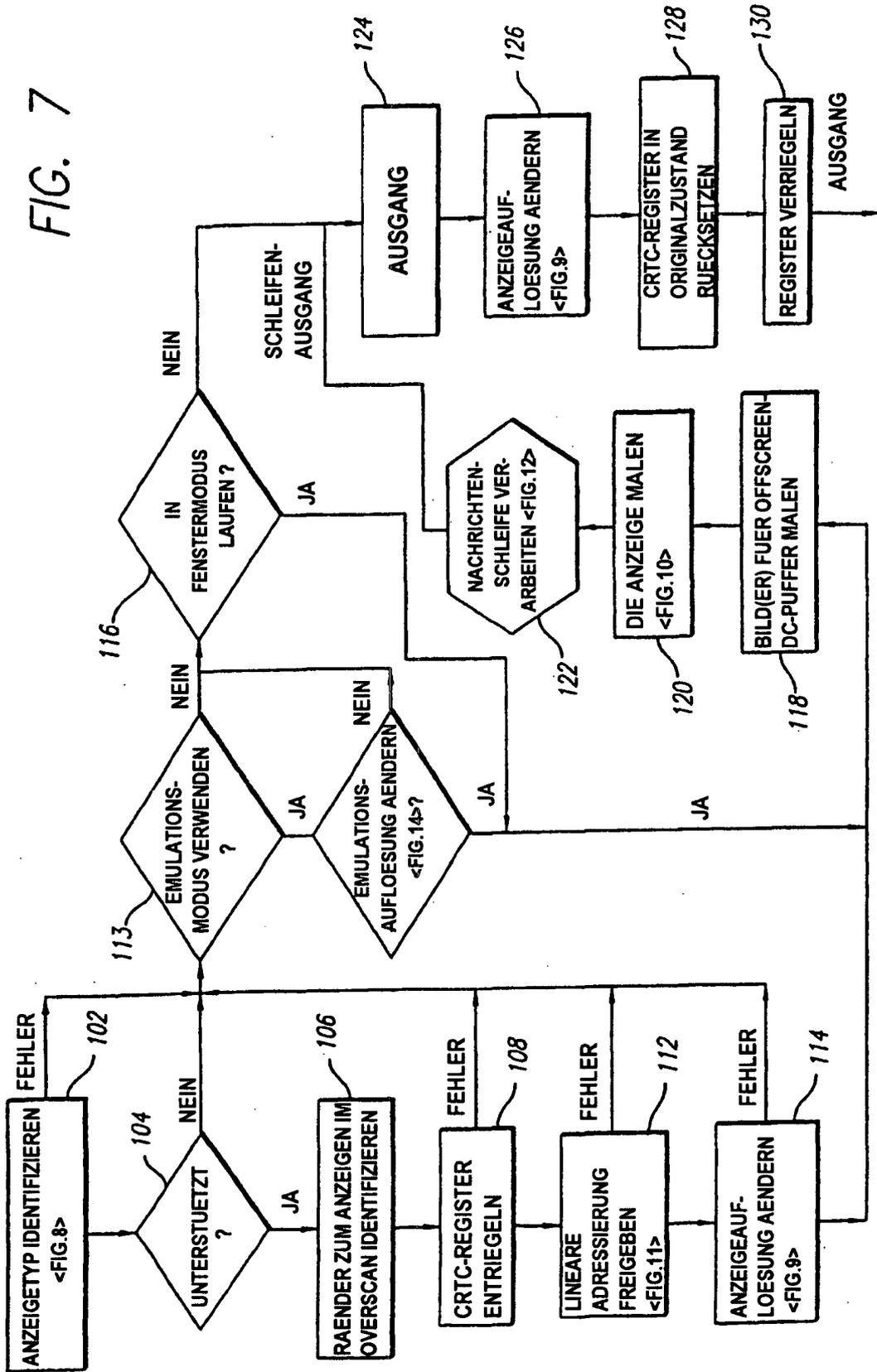


FIG. 8

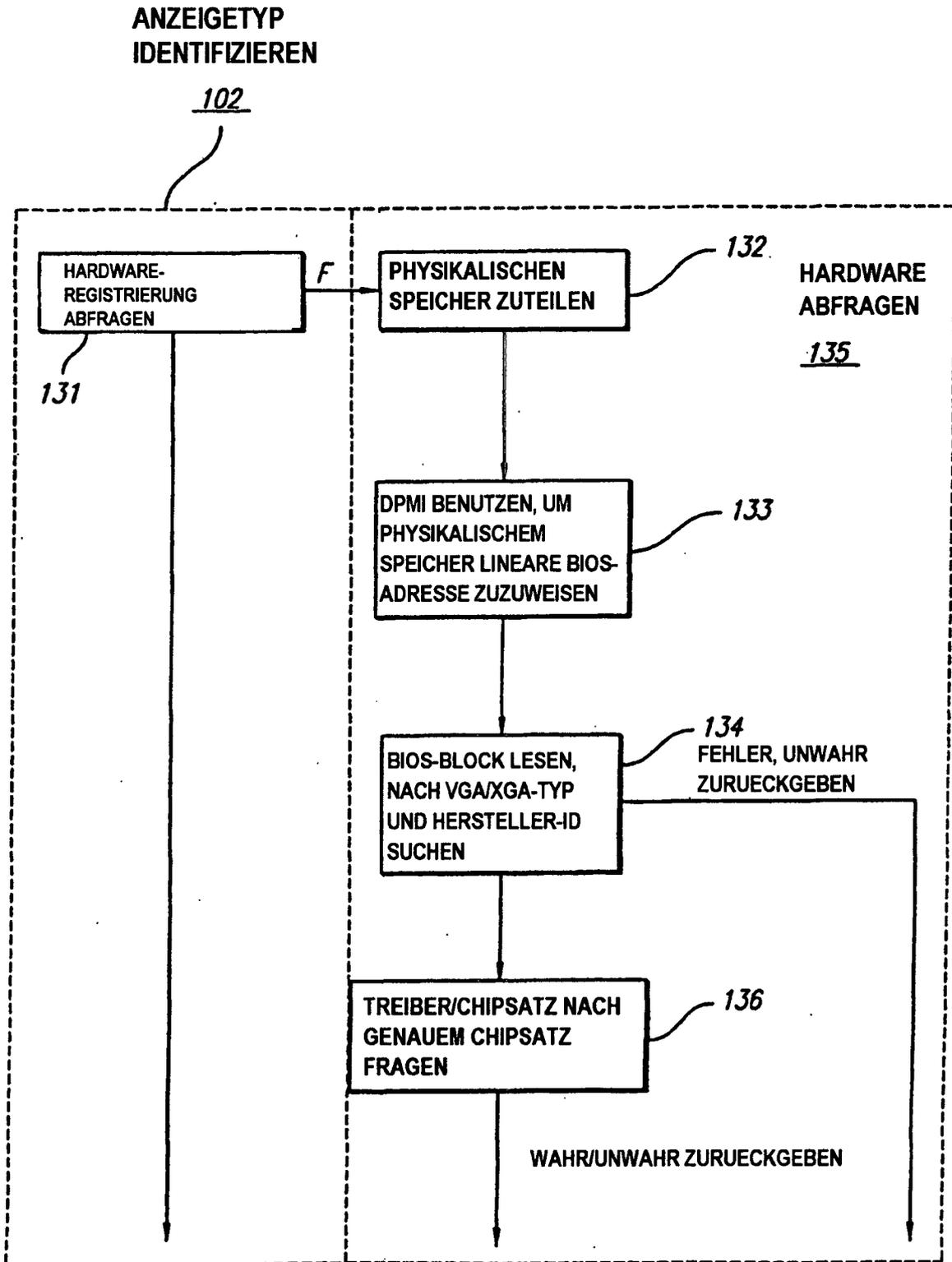
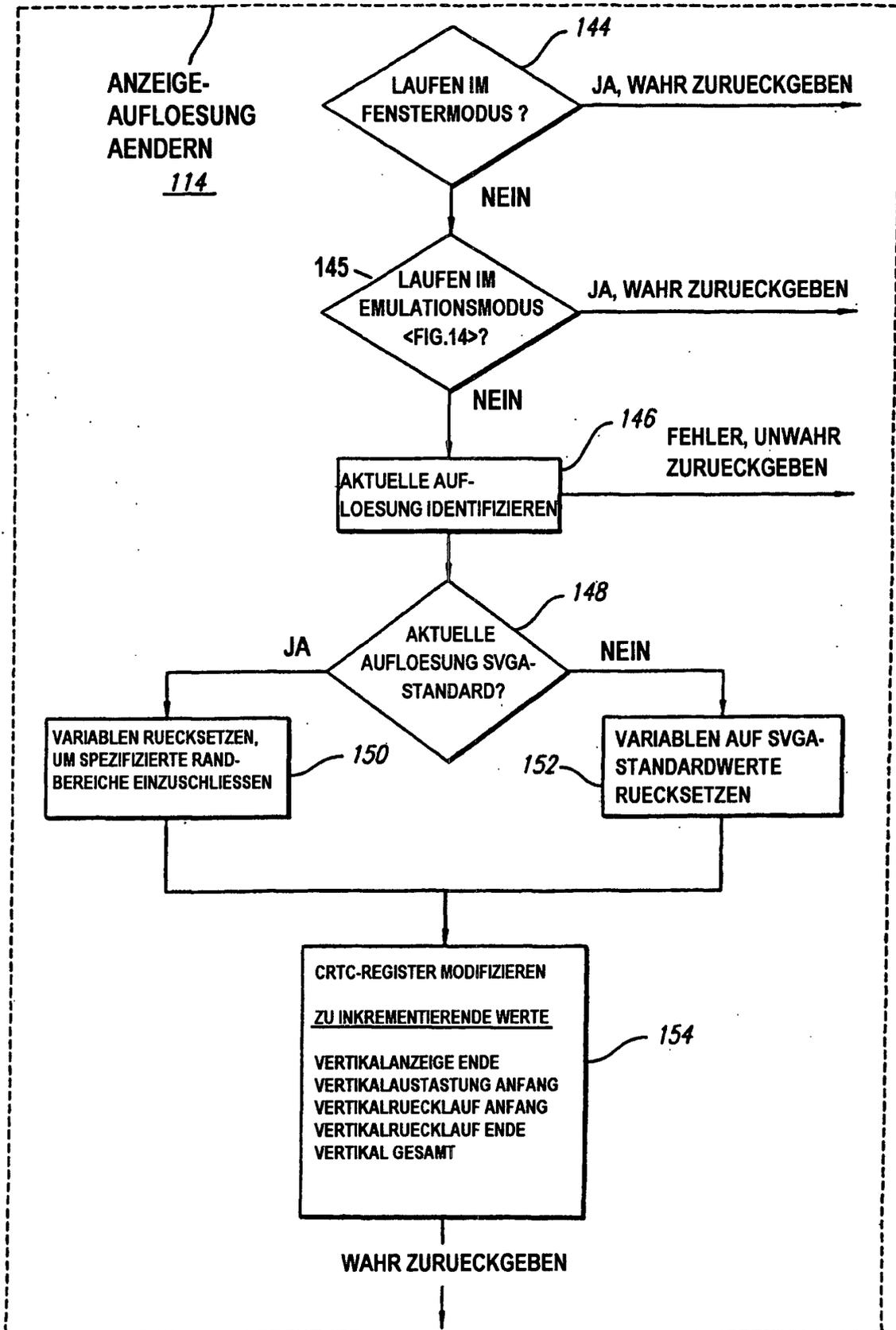


FIG. 9



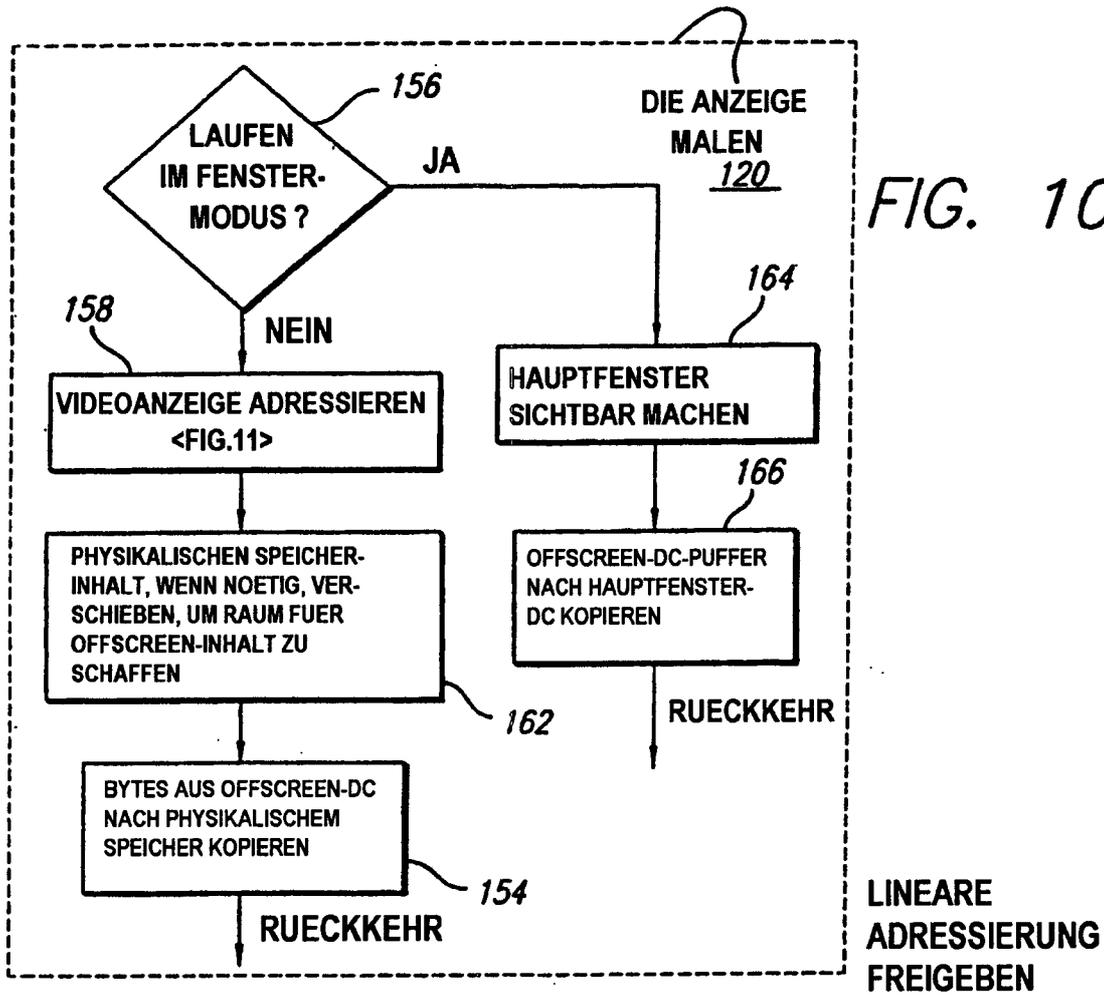


FIG. 10

FIG. 11

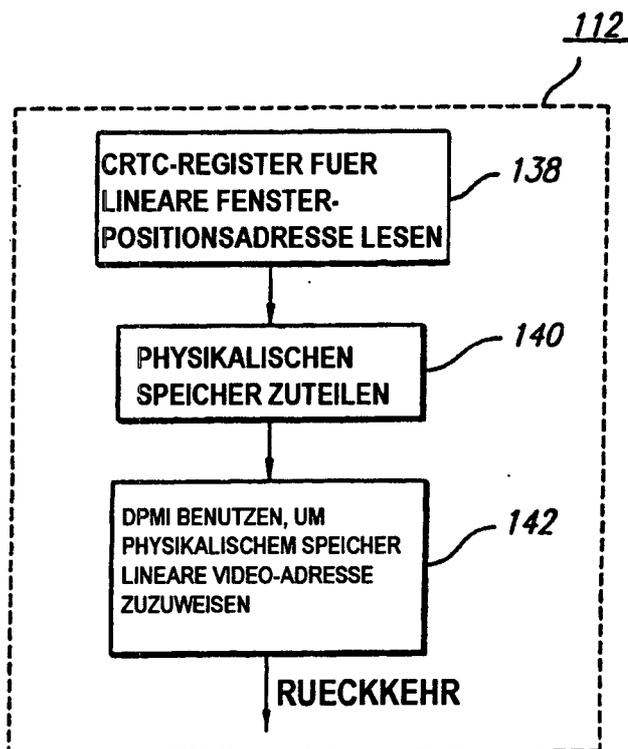


FIG. 12

NACHRICHTENVERARBEITUNGS-  
SCHLEIFEN-BENUTZER-  
SCHNITTSTELLE

122

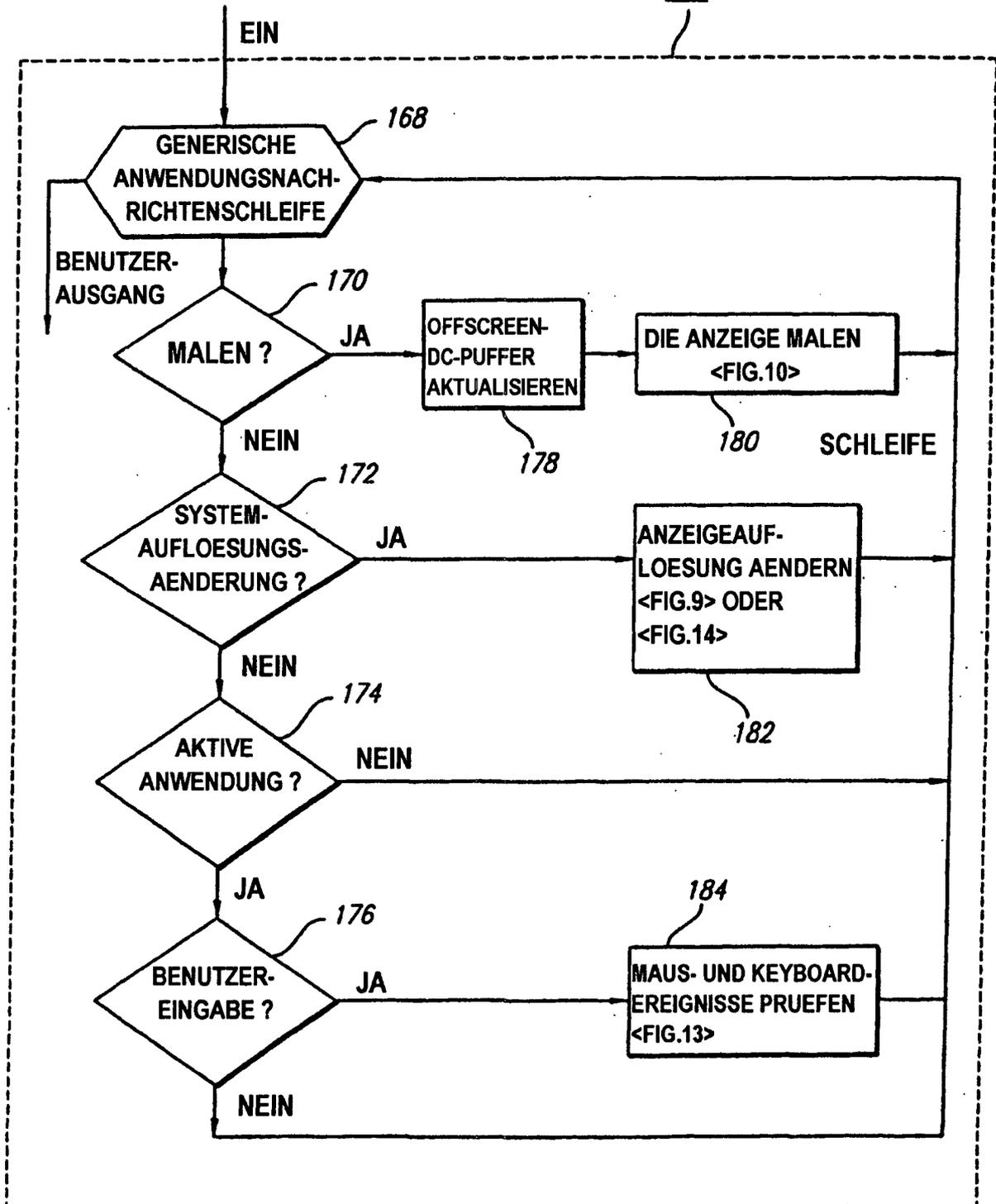


FIG. 13

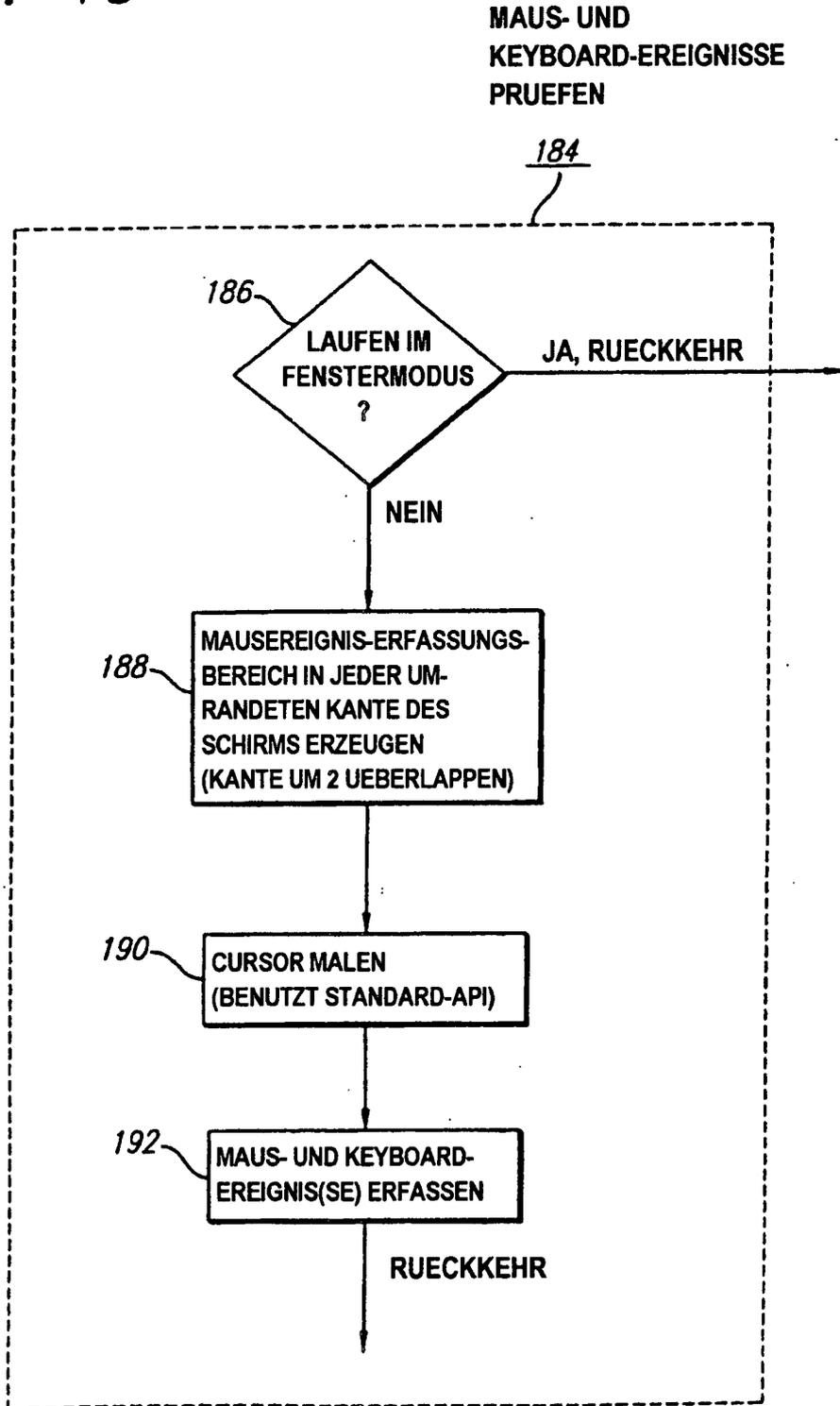


FIG. 14

