



(19)  
Bundesrepublik Deutschland  
Deutsches Patent- und Markenamt

(10) **DE 697 34 658 T2** 2006.08.10

(12)

## Übersetzung der europäischen Patentschrift

(97) **EP 0 917 689 B1**

(21) Deutsches Aktenzeichen: **697 34 658.7**

(86) PCT-Aktenzeichen: **PCT/US97/13655**

(96) Europäisches Aktenzeichen: **97 939 356.8**

(87) PCT-Veröffentlichungs-Nr.: **WO 1998/008192**

(86) PCT-Anmeldetag: **01.08.1997**

(87) Veröffentlichungstag

der PCT-Anmeldung: **26.02.1998**

(97) Erstveröffentlichung durch das EPA: **26.05.1999**

(97) Veröffentlichungstag

der Patenterteilung beim EPA: **16.11.2005**

(47) Veröffentlichungstag im Patentblatt: **10.08.2006**

(51) Int Cl.<sup>8</sup>: **G06T 15/70** (2006.01)  
**G06T 1/00** (2006.01)

(30) Unionspriorität:

**691996 02.08.1996 US**

(73) Patentinhaber:

**Microsoft Corp., Redmond, Wash., US**

(74) Vertreter:

**BOEHMERT & BOEHMERT, 28209 Bremen**

(84) Benannte Vertragsstaaten:

**DE, FR, GB**

(72) Erfinder:

**COHEN, F., Michael, Seattle, US; HE, Li-Wei, Seattle, US; SALESIN, David, Seattle, US**

(54) Bezeichnung: **VERFAHREN UND SYSTEM FÜR VIRTUELLE KINEMATOGRAPHIE**

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 betrifft die Computeranimation. Genauer gesagt betrifft die vorliegende Erfindung ein Erzeugen von Spezifikationen zum Darstellen von Ereignissen in virtuellen, dreidimensionalen, Computer-animierten Umgebungen.

**HINTERGRUND UND ZUSAMMENFASSUNG DER ERFINDUNG**

**[0002]** Das explosive Wachstum der Computernetzwerke, wie dem Internet, hat Benutzern von Computern einen zweckmäßigen Weg bereitgestellt, von entfernt liegenden Stellen Informationen in der Form von Text, Graphiken und Audio- und Videosegmenten zu erhalten. Ein Computer, der mit dem Internet oder einem anderen Computernetzwerk (z.B. Intranet) verbunden ist, kann von einem Benutzer eines Computers auch eingesetzt werden, um in Echtzeit mit anderen Benutzern von Computern zu interagieren, welche mit dem Computernetzwerk verbunden sind. Beispielsweise kann ein Benutzer eines Computers an einem animierten Computer-Videospiel mit einem Benutzer eines entfernten Computers teilnehmen. Manche dieser animierten Computer-Videospiele setzen eine vom Computer erzeugte, virtuelle, dreidimensionale (3D) Umgebung ein, in welcher der Benutzer des Computers animierte virtuelle Charakteren steuert.

**[0003]** Auch virtuelle 3D-Chat-Umgebungen genießen zunehmende Popularität, in welchen Benutzer von Computern miteinander durch animierte, virtuelle 3D-Darsteller (manchmal virtuelle Figuren (avatars) genannt) interagieren, welche von dem Benutzer des Computers gesteuert werden und ihn repräsentieren. Bei diesem Typ einer Chat-Umgebung wird jedem Benutzer eines Computers beispielsweise eine 3D-Anzeige eines Raums zur Verfügung gestellt, in welchem die virtuellen Darsteller oder virtuellen Figuren gemäß der Kommunikation der Benutzer miteinander aufbereitet werden. Die Anordnung und das Positionieren der virtuellen Darsteller stellen den Benutzern der Computer eine 3D-Anzeige bereit, welche anzeigt, welche Benutzer von Computern miteinander kommunizieren. Dieser Typ einer graphischen Anzeige ist in einer herkömmlichen Chat-Umgebung, welche Text als eine Kommunikationsschnittstelle verwendet, nicht möglich.

**[0004]** Diese Form einer Kommunikation innerhalb einer virtuellen 3D-Umgebung, so vielversprechend sie auch ist, weist auch eine Anzahl zugehöriger Probleme auf. Beispielsweise haben Benutzer oft Schwierigkeiten, die virtuelle 3D-Umgebung zu verstehen und zu navigieren, in der simulierten Umgebung die virtuellen Darsteller der Benutzer von Computern zu lokalisieren, mit welchen sie kommunizieren möchten, und ihre Darsteller in einer derartigen Weise anzuordnen, daß alle Benutzer, welche sich miteinander unterhalten, die Darsteller jedes anderen sehen können.

**[0005]** Diese Problemtypen sind den Problemen ähnlich, welchen Kinematographen gegenüberstehen seit das Filmemachen vor einem Jahrhundert begann. Mit den Jahren haben Filmemacher Konventionen oder Regeln für Filme entwickelt, welche es gestatten, Handlungen verständlich und wirksam zu kommunizieren. Diese Regeln für Filme sind, obwohl sie selten explizit angegeben werden, so üblich, daß sie als selbstverständlich angesehen werden und von Zuschauern verstanden werden. Diese Kinematographiekonventionen oder Regeln für Filme setzen Kamerapositionen, die Szenenstruktur und „Interaufnahme“-Konsistenzregeln ein, um kinematographische Informationen zu befördern. Beispielsweise verstehen Zuschauer eine Szene gut, welche mit einer Ansicht einer Landschaft aus großer Höhe beginnt und in eine Ansicht der Landschaft aus geringerer Höhe, welche von einer Straße und einem Automobil dominiert wird, und dann in eine Nahaufnahme einer Person in einem Automobil übergeht.

**[0006]** Kinematographiekonventionen oder Regeln für Filme zum Steuern von Kamerapositionen und der Szenenstruktur haben relativ wenig Aufmerksamkeit in der Computergraphik-Gemeinde erregt. Eine Anzahl von Computeranimationssystemen versucht manche kinematographischen Prinzipien auf Computergraphik in begrenzten Anwendungen anzuwenden, welche Folgendes umfassen: ein Animationsplanungssystem unter Verwendung einer Offline-Planung didaktischer Präsentationen, um komplexe Aufgaben zu erklären; die Erzeugung von halbautonomen Darstellern, welche auf Befehle in natürlicher Sprache reagieren; und der Zusammenbau von kurzen Video-Clip-Sequenzen aus einer Bibliothek von Video-Filmmaterial. Diese Systeme schenken jedoch der Kameraplatzierung oder den Interaufnahme-Konsistenzregeln typischerweise wenig oder keine Aufmerksamkeit (z.B. würde es inkonsistent erscheinen, wenn ein Darsteller, welcher eine Szene links in einem Bild verläßt, sie von rechts wieder betritt).

**[0007]** Es wurden manche interaktive Animationssysteme zum Auffinden der besten Kameraplatzierung be-

schrieben, wenn interaktive Aufgaben durchgeführt werden. Doch diese Systeme versuchen weder Szenensequenzen zu erzeugen noch wenden sie Regeln der Kinematographie beim Entwickeln ihrer Spezifikationen an.

**[0008]** Ein Automatisieren kinematographischer Prinzipien für eine virtuelle 3D-Anwendung, welche von einem Computer ausgeführt werden, stellt schwierige Probleme dar, welche beim herkömmlichen Filmmachen nicht auftauchen. Während informelle Beschreibungen verschiedener Regeln der Kinematographie in verschiedenen Texten erwähnt werden, sind sie typischerweise nicht ausreichend explizit definiert, um in einer formalen Sprache ausgedrückt zu werden, welche zu einer Ausführung durch einen Computer in der Lage ist. Zusätzlich führt ein Durchführen einer automatischen Bild- oder „Kamera-“ Steuerung in Echtzeit auf einem Computer Einschränkungen ein, welche schwieriger zu überwinden sind als diejenigen, welchen menschliche Regisseure gegenüberstehen. Menschliche Regisseure arbeiten typischerweise nach einem Drehbuch, welches zuvor vereinbart wurde, und können das rohe Filmmaterial zu einem späteren Zeitpunkt offline bearbeiten. Dies ist für eine interaktive 3D-Computeranwendung, welche in Echtzeit ausgeführt wird, nicht möglich.

**[0009]** Die Schrift „declarative camera control for automatic cinematography“ (ein technischer Bericht, welcher von dem Department of Computer Science and Engineering, University of Washington und Microsoft Research, veröffentlicht wurde) offenbart ein Kameraplanungssystem, welches eine vollständige Beschreibung der Bewegungen und Handlungen von Darstellern in einer Sequenz akzeptiert. Als Ausgabe stellt es eine Liste von Kameraspezifikationen für jedes Bild her. Das Kameraplanungssystem verwendet Szenen-spezifische Heuristiken, um die Eingabe zu einer Liste von Szenen zu segmentieren. Es verweist dann auf eine Bibliothek Domänen-unabhängiger, deklaratorischer Sprachidiome zur Kamerasteuerung, um mögliche Wege zum Filmen jeder Szene auszuwählen. Ein Domänen-unabhängiger Aufnahmenbeurteiler wählt die beste Umsetzung für den endgültigen Film aus.

**[0010]** Gemäß der vorliegenden Erfindung werden die Probleme mit dem Automatisieren kinematographischer Prinzipien überwunden. Die vorliegende Erfindung umfaßt ein virtuelles Kinematographieverfahren zum Einfangen oder zum Aufbereiten von Ereignissen in virtuellen 3D-Umgebungen gemäß den automatisierten kinematographischen Prinzipien. Das Verfahren umfaßt ein Akzeptieren einer Beschreibung der Ereignisse, welche innerhalb einer spezifizierten Zeitspanne (z.B. eines Computertaktzyklus) aufgetreten sind. Ereignisse weisen typischerweise eine ausgewählte Form auf, wie beispielsweise (Subjekt, Verb, Objekt). Beispielsweise bedeutet ein Ereignis (B, talk, A), daß sich ein virtueller Darsteller B mit einem virtuellen Darsteller A unterhält. Die akzeptierten Ereignisse werden interpretiert, um eine entsprechende Kameraspezifikation herzustellen, welche verwendet wird, um die virtuellen Darsteller anzusehen.

**[0011]** Das Verfahren verwendet zwei Hauptkomponenten, Kameramodule und kinematographische Idiome. Die Kameramodule sind verantwortlich für die geometrische Low-Level-Plazierung spezifischer Kameras in einer Szene und zum Anfertigen subtiler Änderungen in den Positionen der virtuellen Darsteller, um das beste Bild für jede Kameraaufnahme zu erhalten. Die kinematographischen Idiome beschreiben die kinematographische Logik, welche zum Kombinieren von Kameraaufnahmen zu Sequenzen (z.B. Animationssequenzen) verwendet werden. Die Kameramodule und die kinematographischen Idiome werden zusammen verwendet, um virtuelle Filme und Animationen zu erzeugen.

**[0012]** Das Verfahren wird zum Implementieren eines Echtzeitkameracontrollers auf der Grundlage eines endlichen Automaten für eine automatische virtuelle Kinematographie verwendet, welcher ein virtuelles kinematographisches Anwendungsmodul (VC) genannt wird. Das VC wird in der virtuellen Wirklichkeit und bei anderen interaktiven Anwendungen verwendet, um die Aufnahmen mit festem Ansichtspunkt oder deckenbefestigten Kameras zu verbessern, welche von derartigen Anwendungen typischerweise eingesetzt werden. Das VC hilft auch Benutzerschnittstellen mit einem „intelligenten Agenten“ zu verbessern, indem es den Benutzern gestattet, sich selbst mit einem Agenten an Kamerapositionen zu sehen, welche natürlich erscheinen.

**[0013]** Die vorangehenden und andere Merkmale und Vorteile der vorliegenden Erfindung werden unmittelbar aus der folgenden ausführlichen Beschreibung offenkundig, welche unter Bezugnahme auf die begleitenden Zeichnungen fortfährt.

#### KURZE BESCHREIBUNG DER ZEICHNUNGEN

**[0014]** [Fig. 1](#) ist ein Blockdiagramm eines Computersystems, welches zum Implementieren einer beispielhaften Ausführungsform der vorliegenden Erfindung verwendet wird.

- [0015] [Fig. 2](#) ist ein Ablaufdiagramm, welches ein Verfahren für eine Ausführungsform der vorliegenden Erfindung illustriert.
- [0016] [Fig. 3](#) ist ein Blockdiagramm eines Systems für die vorliegende Erfindung.
- [0017] [Fig. 4](#) ist ein Blockdiagramm, welches das Konzept „der Linie“ illustriert.
- [0018] [Fig. 5](#) ist ein Blockdiagramm, welches Kamerapositionen für die Filmkinematographie illustriert.
- [0019] [Fig. 6](#) ist ein Blockdiagramm, welches eine Kameraaufnahme in Nahansicht und in Vollansicht illustriert.
- [0020] [Fig. 7A](#) und [Fig. 7B](#) sind Blockdiagramme, welche die Anordnung von drei Darstellern bei einer Kameraaufnahme illustrieren.
- [0021] [Fig. 8](#) ist ein Blockdiagramm, welches Kameraaufnahmen illustriert, welche für drei sich unterhaltende Darsteller verwendet werden.
- [0022] [Fig. 9](#) ist ein Blockdiagramm, welches das Kameramodul „apex“ illustriert.
- [0023] [Fig. 10](#) ist ein Blockdiagramm, welches das Kameramodul „closeapex“ illustriert.
- [0024] [Fig. 11](#) ist ein Blockdiagramm, welches das Kameramodul „apex3“ illustriert.
- [0025] [Fig. 12](#) ist ein Blockdiagramm, welches das Kameramodul „external“ illustriert.
- [0026] [Fig. 13](#) ist ein Blockdiagramm, welches das Kameramodul „external1to2“ illustriert.
- [0027] [Fig. 14](#) ist ein Blockdiagramm, welches das Kameramodul „exclose2“ illustriert.
- [0028] [Fig. 15](#) ist ein Blockdiagramm, welches das Kameramodul „internal“ illustriert.
- [0029] [Fig. 16](#) ist ein Blockdiagramm, welches das Kameramodul „full“ illustriert.
- [0030] [Fig. 17](#) ist ein Blockdiagramm, welches das Kameramodul „tracking“ illustriert.
- [0031] [Fig. 18](#) ist ein Blockdiagramm, welches das Kameramodul „panning“ illustriert.
- [0032] [Fig. 19](#) ist ein Blockdiagramm, welches das Kameramodul „follow“ illustriert.
- [0033] [Fig. 20](#) ist ein Blockdiagramm, welches das Kameramodul „fixed“ illustriert.
- [0034] [Fig. 21](#) ist ein Blockdiagramm, welches das Kameramodul „raised“ illustriert.
- [0035] [Fig. 22](#) ist ein Blockdiagramm, welches das Kameramodul „group“ illustriert.
- [0036] [Fig. 23](#) ist ein Blockdiagramm, welches das Kameramodul „subjective“ illustriert.
- [0037] [Fig. 24](#) ist ein Blockdiagramm, welches eine Änderung der Position „der Linie“ illustriert.
- [0038] [Fig. 25](#) ist ein Blockdiagramm, welches einen endlichen Automaten für ein Idiom illustriert.
- [0039] [Fig. 26](#) ist ein Blockdiagramm, welches einen endlichen Automaten mit einem Zustand illustriert, welcher einen anderen endlichen Automaten enthält.
- [0040] [Fig. 27](#) ist ein Blockdiagramm, welches ein Idiom für eine Konversation zwischen 2 Darstellern illustriert.
- [0041] [Fig. 28](#) ist ein Blockdiagramm, welches ein Idiom für eine Konversation zwischen 3 Darstellern illustriert.

[0042] [Fig. 29](#) ist ein Ablaufdiagramm, welches ein Verfahren zum Erzeugen eines individuellen Idioms illustriert.

[0043] [Fig. 30](#) ist ein Ablaufdiagramm, welches ein Verfahren zum Erzeugen eines komplexen Idioms illustriert.

[0044] [Fig. 31](#) ist ein Ablaufdiagramm, welches ein Verfahren zum Erzeugen eines High-Level-Idioms illustriert.

[0045] [Fig. 32](#) ist ein Blockdiagramm, welches eine hierarchische Idiomstruktur illustriert.

[0046] [Fig. 33](#) ist ein Ablaufdiagramm, welches ein Verfahren zum Erzeugen eines virtuellen Films aus Idioms illustriert.

[0047] [Fig. 34](#) ist ein Blockdiagramm, welches ein Darstellungssystem für Idiomkoordinaten illustriert.

[0048] [Fig. 35](#) ist ein Blockdiagramm, welches eine Benutzerschnittstelle für eine virtuelle kinematographische Anwendung illustriert.

#### AUSFÜHRLICHE BESCHREIBUNG EINER BEVORZUGTEN AUSFÜHRUNGSFORM

[0049] Unter Bezugnahme auf [Fig. 1](#) ist ein Computersystem **10** mit einem Computer **12**, welches mindestens eine Hochgeschwindigkeitszentraleinheit (CPU) **14** in Verbindung mit einem Speichersystem **16**, einem Eingabegerät **18** und einem Ausgabegerät **20** umfaßt, eine Arbeitsumgebung für eine beispielhafte Ausführungsform der vorliegenden Erfindung. Diese Elemente werden durch eine Bus-Struktur **22** zusammengeschaltet.

[0050] Die dargestellte CPU **14** hat einen bekannten Aufbau und umfaßt ein ALU **24** zum Durchführen von Berechnungen, eine Anzahl von Registern **26** zur temporären Speicherung von Daten und Befehlen und eine Steuerungseinheit **28** zum Steuern des Betriebs des Systems **10**. Alle verschiedenen Prozessoren, einschließlich denjenigen von Digital Equipment, Sun, MIPS, IBM, Motorola, NEC, Intel, Cyrix, AMD, Nexgen und anderen sind gleichermaßen als CPU **14** einsetzbar. Obwohl das Computersystem **10** mit einer CPU **14** gezeigt ist, kann es ersatzweise mehrere Zentraleinheiten umfassen.

[0051] Das Speichersystem **16** umfaßt einen Hauptspeicher **30** und einen sekundären Speicher **32**. Der illustrierte Hauptspeicher **30** ist ein Hochgeschwindigkeits-Direktzugriffsspeicher (RAM) und ein Nur-Lese-Speicher (ROM). Der Hauptspeicher **30** kann jede zusätzliche oder alternative Hochgeschwindigkeits-Speichervorrichtung oder Speicherschaltung umfassen. Der sekundäre Speicher **32** nimmt die Gestalt eines Langzeitspeichers an, wie beispielsweise ROM, optischer oder magnetischer Platten, organischen Speichers oder jedes anderen flüchtigen oder nichtflüchtigen Massenspeichersystems. Durchschnittsfachleute werden erkennen, daß der Speicher **16** verschiedene und/oder eine Kombination alternativer Komponenten umfassen kann.

[0052] Die Eingabe- und Ausgabegeräte **18**, **20** sind ebenfalls bekannt. Das Eingabegerät **18** kann eine Tastatur, eine Maus, ein Zeigergerät, ein Audiogerät (z.B. ein Mikrofon usw.) oder jedes andere Gerät umfassen, welches dem Computersystem **10** eine Eingabe bereitstellt. Das Ausgabegerät **20** kann eine Anzeige, einen Drucker, ein Audiogerät (z.B. einen Lautsprecher usw.) oder ein anderes Gerät umfassen, welches dem Computersystem **10** eine Ausgabe bereitstellt. Die Eingabe-/Ausgabe-Geräte **18**, **20** können auch Netzwerkverbindungen, Modems oder andere Vorrichtungen umfassen, welche zur Kommunikation mit anderen Computersystemen oder – Vorrichtungen verwendet werden.

[0053] Wie es Durchschnittsfachleuten vertraut ist, umfaßt das Computersystem **10** weiterhin ein Betriebssystem und mindestens ein Anwendungsprogramm. Das Betriebssystem umfaßt einen Satz Software, welche den Betrieb des Computersystems und die Zuteilung der Betriebsmittel steuert. Das Anwendungsprogramm umfaßt einen Satz Software, welcher eine Aufgabe durchführt, welche von dem Benutzer gewünscht wird, wobei die Betriebsmittel des Computers verwendet werden, welche durch das Betriebssystem verfügbar gemacht werden. Beide sind in dem dargestellten Speichersystem **16** vorhanden.

[0054] Gemäß den Praktiken von Durchschnittsfachleuten der Computerprogrammierung wird die vorliegende Erfindung nachfolgend unter Bezugnahme auf Handlungen und symbolische Repräsentationen von Operationen beschrieben, welche von dem Computersystem **10** durchgeführt werden, außer es ist anderslautend an-

gegeben. Derartige Handlungen und Operationen werden manchmal als vom Computer ausgeführt bezeichnet: Es ist anerkannt, daß die Handlungen und symbolisch dargestellten Operationen die Manipulation elektrischer Signale, welche Datenbits repräsentieren, durch die CPU **14**, welche eine resultierende Transformation oder Reduktion der elektrischen Signalrepräsentation bewirkt, und das Erhalten von Datenbits an Speicherorten in dem Speichersystem **16**, um dadurch den Betrieb des Computersystems zu rekonfigurieren oder auf andere Weise zu verändern, sowie ein anderes Verarbeiten von Signalen umfassen. Die Speicherorte, an welchen Datenbits erhalten werden, sind physikalische Orte, welche entsprechend der Datenbits bestimmte elektrische, magnetische, optische oder organische Eigenschaften aufweisen.

**[0055]** Die Datenbits können auch auf einem computerlesbaren Medium einschließlich magnetischer Platten und jedes anderen flüchtigen oder nichtflüchtigen Massenspeichersystems erhalten werden, welches von dem Computer **12** lesbar ist. Das computerlesbare Medium umfaßt kooperierende oder zusammengeschaltete computerlesbare Medien, welche ausschließlich auf dem Computersystem **10** existieren oder auf mehrere zusammengeschaltete Computersysteme **10** verteilt sind, welche lokal oder entfernt sein können.

**[0056]** Wie in [Fig. 2](#) gezeigt, umfaßt eine beispielhafte Ausführungsform der vorliegenden Erfindung ein virtuelles Kinematographieverfahren **34** zum Aufbereiten Computer-erzeugter Szenen in virtuellen 3D-Umgebungen. Eine derartige virtuelle 3D-Umgebung stellt einen 3D-Bildraum dar, innerhalb dessen Anzeigebilder virtueller Objekte oder Charakteren von einem Computer erzeugt werden und auf einer Darstellungsfläche aufbereitet werden. Die Kinematographie ist dadurch „virtuell“, daß Bilder nicht mit Kameras gebildet werden, sondern statt dessen gemäß Computer-erzeugten Ansichtsrichtungen, Vergrößerungen usw.

**[0057]** Das Verfahren **34** umfaßt ein Akzeptieren einer Beschreibung von Ereignissen, welche in einer spezifizierten Zeitspanne **36** auftreten (z.B. in einem Computertaktzyklus). Ereignisse werden typischerweise in einer ausgewählten Form repräsentiert, wie beispielsweise (Subjekt, Verb, Objekt). Beispielsweise bedeutet ein Ereignis (B, talk, A), daß sich ein virtueller Darsteller B mit einem virtuellen Darsteller A unterhält. Die Ereignisse werden interpretiert **38**, um eine entsprechende virtuelle Kameraspezifikation **40** herzustellen. Es werden auch Darstellungshinweise erzeugt, um die virtuellen Darsteller zu verschieben und eine Aufnahme besser zu erfassen.

**[0058]** [Fig. 3](#) illustriert ein System **44** für eine beispielhafte Ausführungsform der vorliegenden Erfindung. Das System **44** umfaßt ein Echtzeitanwendungsmodul **46** (z.B. eine Multibenutzer-Chat-Anwendung, ein Computer-Videospiel, eine Videokonferenzanwendung, eine Anwendung mit virtueller Wirklichkeit usw.), welches einem virtuellen kinematographischen Anwendungsmodul **50** eine Beschreibung der Ereignisse und andere Informationen **48** einschließlich statischer geometrischer Informationen, Materialeigenschaften (z.B. Textur und Schattierung) und Platzierung der Lichtquelle bereitstellt. Das virtuelle kinematographische Anwendungsmodul **50** implementiert das Verfahren **34** für eine virtuelle Kinematographie und sendet virtuelle Kamerainformationen und Manipulationsinformationen **52** virtueller Darsteller an ein Aufbereitermodul **54**. Das virtuelle kinematographische Anwendungsmodul **50** kann auch Abfragen **56** von Informationen (z.B. der Position eines virtuellen Darstellers) aus dem Echtzeitanwendungsmodul **46** vornehmen. Das Aufbereitermodul **54** umfaßt Graphik- und Animations-Software, wie beispielsweise Direct3D, welches von der Microsoft Corporation in Redmond, Washington, erhältlich ist. Es kann jedoch auch andere Aufbereiter-Software verwendet werden. Das Aufbereitermodul **54** akzeptiert Animationsparameter **58** (z.B. statische Geometrieinformationen, virtuelle Darstellermodelle, Lichtplatzierung usw.) aus dem Echtzeitanwendungsmodul **46**.

**[0059]** In vorgesehenen Zeitintervallen (z.B. einem Taktzyklus) ist das Echtzeitanwendungsmodul **46** auch für ein Annehmen einer Eingabe von einem Benutzer, wobei virtuelle Darsteller entsprechend bewegt werden, und für ein Zurücksenden der resultierenden Animationsinformationen **60** (z.B. virtuelle Darstellerpositionen) an das Aufbereitermodul **54** verantwortlich. Zusätzlich entscheidet das virtuelle kinematographische Anwendungsmodul **50** in den vorgesehenen Zeitintervallen über die Spezifikation der virtuellen Kamera, von welcher die virtuelle Umgebung aufbereitet wird. Das virtuelle kinematographische Anwendungsmodul **50** nimmt, falls nötig, auch leichte Modifikationen an der Anordnung und den Handlungen der virtuellen Darsteller vor.

#### Filmkinematographie

**[0060]** Ein sich bewegendes Bild oder ein Film ist typischerweise eine Sequenz aus mehreren Szenen, von welchen jede eine im Allgemeinen kontinuierliche Situation oder Handlung einfängt. Jede Szene besteht wiederum aus einer oder mehreren Kameraaufnahmen. Eine Kameraaufnahme ist ein sich bewegendes Bildsegment, welches von dem kontinuierlichen Betrieb der Kamera gebildet wird. Typischerweise umfaßt ein Film eine große Anzahl individueller Kameraaufnahmen, wobei jede Aufnahme typischerweise von einer Sekunde



oder zwei bis zu mehreren zehn Sekunden dauert.

**[0061]** [Fig. 4](#) illustriert eine Kinematographiekonvention oder eine Filmregel, welche bei den meisten sich bewegenden Bildern verwendet wird. Insbesondere Filmregisseure spezifizieren Kamerapazierungen relativ zu einer „Linie“ **62**, welche ein imaginärer Vektor ist, welcher zwei interagierende Darsteller X **64** und Y **66** verbindet. Die Linie **62** ist auch entlang der Bewegungslinie **68** eines Darstellers ausgerichtet oder in der Richtung orientiert, in welche der Darsteller blickt. Eine Kamerapazierung **70** wird relativ zu der Linie **62** spezifiziert.

**[0062]** Wie in [Fig. 5](#) hinsichtlich der Linie **62** gezeigt, wird ein Aufnehmen des Darstellers Y **66** aus einer Kameraposition A **72** eine „externe Umkehrplazierung“ genannt; die rechten zwei Drittel des Darstellers Y **66** liegen im Fokus, während das linke eine Drittel die Rückseite des Kopfs der Darstellerin X **64** zeigt. Ein Aufnehmen der Darstellerin X **64** aus einer Kameraposition B **74** wird eine „parallele Kamerapazierung“ genannt, welche nur Darstellerin X **64** zeigt. Ein Aufnehmen der Darstellerin X **64** aus einer Position C **76** ergibt eine „interne Umkehrplazierung“. Typischerweise nimmt die Darstellerin X **64** bei der Kameraeinstellung C **76** nur die linken zwei Drittel der Aufnahme ein. Ein Aufnehmen aus der Position D **78** führt zu einer Aufnahme mit „apex-Plazierung“, welche sowohl Darstellerin X **64** als auch Darsteller Y **66** zeigt. Ein Aufnehmen aus Position E **80** ergibt eine „interne Umkehrplazierung“ für den Darsteller Y **66**. Ein Aufnehmen aus Position F **82** ergibt eine „parallele Kamerapazierung“ für den Darsteller Y **66**. Ein Aufnehmen von Position **84** wird schließlich eine „externe Umkehrplazierung“ genannt; die linken zwei Drittel der Aufnahme zeigen die Darstellerin X **64** im Fokus, während das rechte eine Drittel die Rückseite des Kopfs des Darstellers Y **66** zeigt. Es können jedoch auch mehr oder weniger Kamerapositionen von einem Filmregisseur verwendet werden.

**[0063]** Kinematographen haben identifiziert, daß bestimmte „Schnitthöhen“ für ansprechende Kompositionen sorgen, während andere unangenehme Resultate ergeben (z.B. ist ein Bild eines Darstellers, welches an den Knöcheln abgeschnitten ist, nicht ansprechend). Es gibt fünf verwendbare Kameraentfernungen, welche typischerweise in der Technik verwendet werden: (1) eine extreme Nahaufnahmenansicht, welche am Nacken abgeschnitten ist; (2) eine Nahaufnahmenansicht, welche unter der Brust oder an der Taille abgeschnitten ist; (3) eine mittlere Ansicht, welche am Schritt oder unter den Knien abgeschnitten ist; (4) eine Vollansicht, welche den ganzen Darsteller zeigt; und (5) eine Distanzansicht, welche eine entfernte Perspektive des Darstellers bereitstellt.

**[0064]** Unter Verwendung der Schnitthöhen können Darstellerpositionen, welche für eine bestimmte Nahaufnahme natürlich aussehen, zu nah zusammen erscheinen, wenn sie von weiter hinten gesehen werden. Individuelle Aufnahmen erfordern auch eine subtil unterschiedliche Plazierung der Darsteller, damit sie auf dem Bildschirm natürlich aussehen. Wie beispielsweise in [Fig. 6](#) gezeigt, sieht eine Nahaufnahmenansicht **86** der Darstellerin X **88** und eines anderen Darstellers Y **90** hinsichtlich der Linie **62** völlig natürlich aus. Eine Vollansicht **92** zeigt jedoch die Darstellerin X **88** und den Darsteller Y **90** näher zusammen als erwartet, und folglich ist die Vollansicht **92** unangenehm.

**[0065]** Genauso erfordern Aufnahmen mit mehreren Darstellern oft ein Verschieben der Darstellerpositionen, um sie sachgerecht abzubilden. Wie in [Fig. 7A](#) hinsichtlich der Linie **62** gezeigt, verstecken oder blockieren eine erste Darstellerin X **94** und ein zweiter Darsteller Y **96** eine dritte Darstellerin Z **98** teilweise. Ein Verschieben der Position der dritten Darstellerin Z **98** wird die Aufnahme sachgerecht abbilden, um die dritte Darstellerin Z **98** einzuschließen. Wie in [Fig. 7B](#) gezeigt, bleiben die erste Darstellerin X **94** und der zweite Darsteller Y **96** in den gleichen Positionen, doch die dritte Darstellerin Z **98** wurde verschoben. Die Positionen der ersten Darstellerin X **94** und des zweiten Darstellers Y **96** können jedoch auch verschoben sein, was die dritte Darstellerin Z **98** in ihrer ursprünglichen Position läßt, um das gleiche Ergebnis zu erzielen.

#### Filmkinematographische Heuristiken

**[0066]** Filmemacher haben zahlreiche Heuristiken (d.h. Regeln) zum Auswählen guter Aufnahmen artikuliert und haben informell Einschränkungen spezifiziert, welche auf aufeinanderfolgende Aufnahmen angewendet werden, um gute Szenen zu erzeugen. Die wichtigeren Heuristiken umfassen: „Don't cross the Line“; „Avoid jump cuts“; „Use establishing shots“; „Let the actor lead“; „Break movement“; und „Maintain continuous movement“. Es können jedoch auch mehr oder weniger Heuristiken verwendet werden.

**[0067]** Die Regel „Don't cross the Line“ gewährleistet, daß aufeinanderfolgende Aufnahmen eines sich bewegenden Darstellers die Richtung der offenkundigen Bewegung beibehalten. Wenn einmal eine anfängliche Aufnahme von einer Seite der Linie **62** aufgenommen wurde, wie in [Fig. 5](#) gezeigt, sollten auch nachfolgende Aufnahmen von dieser Seite angefertigt werden, außer es wird eine neutrale Einführungsaufnahme verwendet,

um den Übergang auf die andere Seite zu zeigen. Eine Bewegung des Darstellers Y **66** nach rechts in [Fig. 5](#) würde für eine Kamera auf der anderen Seite der Line **62** nach links erscheinen.

**[0068]** Eine Änderung der Kameraaufnahme innerhalb einer Szene (d.h. ein Schnitt) sollte eine merkliche Differenz in der Größe, der Ansicht oder der Anzahl der Darsteller zwischen den beiden Aufnahmen sein, um „Avoid jump cuts“ zu befolgen. Ein Schnitt, welcher diese Bedingungen nicht erfüllt, ist als ein abrupter Schnitt bekannt und ergibt im Allgemeinen eine ruckartige, nachlässige Wirkung.

**[0069]** Die Regel „Use establishing shots“ führt eine Hintergrundszene ein, bevor zu Nahaufnahmen übergegangen wird. Wenn es eine neue Entwicklung in der Szene gibt, wird die geänderte Hintergrundsituation erneut eingeführt.

**[0070]** „Let the actor lead“ gestattet einem Darsteller, alle Bewegung in Gang zu setzen, wobei die Kamera folgt. Die Kamera kommt etwas vor dem Darsteller zur Ruhe.

**[0071]** Für die Regel „Break movement“ wird eine Szene, welche Bewegung illustriert, in mindestens zwei Aufnahmen zerteilt. Typischerweise wird jede Aufnahme so geschnitten, daß der Darsteller erscheint, wie er sich quer über eine Hälfte der Darstellungsfläche bewegt. Eine Änderung der Entfernung zwischen Kamera und Subjekt wird beim Umschalten vorgenommen.

**[0072]** Die Regel „Maintain continuous movement“ besagt, daß über einen Schritt hinweg die offenkundige Geschwindigkeit des sich bewegendes Objekts konstant sein sollte und die Bewegungsrichtungen nicht gegenüberliegend sein sollten.

#### Sequenzen von Kameraaufnahmen

**[0073]** Kinematographen verwenden Formeln zum Einfangen spezifischer Handlungen als Aufnahmesequenzen. Bei einem Dialog von drei Darstellern beispielsweise könnte ein Filmemacher mit einer Einführungsaufnahme aller drei Personen beginnen, bevor er zu einer externen Umkehraufnahme von zwei der Darstellern übergeht, wie sie sich unterhalten, wobei gelegentlich Aufnahmen von Reaktionen des dritten Darstellers eingestreut werden. Bücher, wie beispielsweise D. Arijon, Grammar of the Film Language, Hasting House, New York, 1976, stellen eine informelle Kompilation von Formeln zusammen mit einer Erörterung der Situationen bereit, bei welchen ein Filmemacher eine Formel gegenüber einer anderen vorziehen könnte.

**[0074]** [Fig. 8](#) illustriert eine übliche Formel zum Darstellen einer Unterhaltung mit drei Darstellern. Die Formel, welche aus [Fig. 6.29](#) in dem Buch von D. Arijon angepaßt ist, stellt ein Verfahren zum Darstellen von Unterhaltungen mit drei Darstellern bereit. Die erste Aufnahme ist eine externe Umkehraufnahme **100** über die Schulter von Darsteller A **102** (runder Kopf) in Richtung auf den Darsteller B **104** (achteckiger Kopf) und auf den Darsteller C **106** (dreieckiger Kopf). Eine Ansicht der externen Umkehraufnahme **100** wird im Kasten **108** illustriert (Darsteller B **104'** und C **106'** und die Rückseite des Kopfs des Darstellers A **102'**).

**[0075]** Die zweite Aufnahme **110** und die dritte Aufnahme **112** sind externe Umkehraufnahmen der Darsteller B **104** und C **106**. Ansichten dieser Aufnahmen werden in den Kästen **116** bzw. **118** gezeigt. Die externe Umkehraufnahme **110** zeigt in Kasten **116** die linken zwei Drittel der Rückseite des Kopfs des Darstellers C **106'''** (dreieckiger Kopf) und das rechte eine Drittel des Darstellers B **104'''** (achteckiger Kopf). Die externe Umkehraufnahme **112** zeigt in Kasten **118** die rechten zwei Drittel des Kopfs des Darstellers B **104'''** und das linke eine Drittel des Darstellers C **106'''**. Die vierte Aufnahme **120** ist eine interne Umkehrreaktionsaufnahme des Darstellers A **102'''**. Eine Ansicht dieser Aufnahme **120** wird in Kasten **122** gezeigt. Es wird nur die Vorderseite des Kopfs des Darstellers A **102'''** gezeigt, da es keine anderen Darsteller „hinter“ dem Darsteller A **102'''** gibt. Die Rückseite des Kopfs wird nicht gezeigt, da dies eine interne Umkehrreaktionsaufnahme ist.

**[0076]** In seinem Buch schreibt Arijon vor, daß eine Bearbeitungsreihenfolge für eine typische Sequenz, welche diese Einstellung verwendet, zwischen der ersten Aufnahme **100** (Kasten **108**) und der vierten Aufnahme **120** (Kasten **122**) abwechseln würde, während sich Darsteller B **104** und C **106** mit Darsteller A **102** unterhalten. Wenn die Darsteller B **104** und C **106** sich miteinander zu unterhalten beginnen, geht die Sequenz zu einem Abwechseln zwischen der zweiten Aufnahme **110** (Kasten **116**) und der dritten Aufnahme **112** (Kasten **118**) mit einer gelegentlichen Reaktionsaufnahme **120** (Kasten **122**) über. Die Aufnahme **100** wird periodisch eingeführt (z.B. alle 15 Sekunden), um die ganze Gruppe wieder einzuführen.

**[0077]** Während es eine breite Vielfalt von Formeln für ein Zusammenstellen von Aufnahmen in einer Se-



quenz gibt, neigt jeder Filmregisseur dazu, sich auf einen kleinen Untersatz der Formeln zu verlassen. Die bestimmten Formeln, welche von irgendeinem individuellen Regisseur verwendet werden, verleihen den Filmen dieses Regisseurs einen bestimmten Charakter oder Stil. Bei einer beispielhaften Ausführungsform der vorliegenden Erfindung wird der Stil des Regisseurs von den bestimmten Formeln diktiert, welche in dem virtuellen kinematographischen Anwendungsmodul **50** (hier nachfolgend das VC genannt) codiert werden.

#### Architektur

**[0078]** Die oben stehend beschriebene Filmkinematographiekompetenz ist in Computer-Software in dem VC **50** als Handlungen und symbolische Repräsentationen von Operationen codiert, welche von Computersystem **10** durchgeführt werden. Das VC **50** enthält zwei Hauptkomponenten: Kameramodule und Idiome. Kameramodule implementieren virtuelle Kameras, aus welchen Computer-erzeugte Bilder unter Verwendung der unterschiedlichen Kamerapläzierungen ausgebildet werden, welche oben stehend beschrieben und in [Fig. 5](#) illustriert sind. Die Kameramodule können auch die Position der virtuellen Darsteller leicht modifizieren, um eine bessere Aufnahme zu komponieren und den Verhaltensrahmen der virtuellen Darsteller zu beeinflussen. Idiome beschreiben die Logik, welche zum Vereinigen von Aufnahmen aus den Kameramodulen zu Sequenzen verwendet werden, wie auch oben stehend für Filme beschrieben wurde.

#### Kameramodule

**[0079]** Jedes Kameramodul nimmt als Eingabe eine Anzahl der virtuellen Darsteller, primäre Darsteller genannt, an, wobei die genaue Zahl von dem bestimmten Kameramodul abhängt. Jedes Kameramodul platziert die virtuelle Kamera in einer bestimmten Entfernung von den Darstellern, um die fünf oben stehend diskutierten Schnitthöhen anzupassen. Es können jedoch auch mehr oder weniger Schnitthöhen verwendet werden. Die virtuelle Kameraposition platziert auch die Darsteller an bestimmten Orten in der Aufnahme. Zusätzlich kann das Kameramodul die Darsteller etwas neu positionieren, um die Aufnahme zu verbessern. Schließlich wird die Kamerapläzierung automatisch gewählt, um die oben stehend erklärte Regel, die Line **62** nicht zu kreuzen, zu befolgen.

**[0080]** Bei einer beispielhaften Ausführungsform der vorliegenden Erfindung wurden 19 unterschiedliche Kameramodule implementiert, wie nachfolgend angegeben. Es können jedoch auch mehr oder weniger Kameramodule verwendet werden.

**[0081]** Wie in [Fig. 9](#) gezeigt, nimmt Kameramodul apex(actor1, actor2) **124** als Eingabe zwei Darsteller A **126** und B **128** (entsprechend actor1 und actor2) an und platziert die virtuelle Kamera **130** so, daß der erste Darsteller A **126** auf einer Seite der Aufnahme zentriert ist und der zweite Darsteller B **128** auf der anderen Seite hinsichtlich der Line **62** zentriert ist. Die Kameraentfernung ist eine Funktion der Entfernung zwischen den beiden Darstellern A **126** und B **128**, welche schematisch durch Kreise angegeben sind.

**[0082]** Wie in [Fig. 10](#) gezeigt, implementiert Kameramodul closeapex(actor1, actor2) **132** auch eine apex-Kamerapläzierung. Das Kameramodul closeapex unterscheidet sich jedoch von dem Kameramodul apex() darin, daß das letztere immer eine Nahaufnahmen-Kameraentfernung für die virtuelle Kamera **134** hinsichtlich der Line **62** verwendet. Um eine ansprechendere Aufnahme zu komponieren, kann dieses Kameramodul **132** die Darsteller A **136** und B **138** näher zusammen bewegen.

**[0083]** Wie in [Fig. 11](#) gezeigt, stellt das Kameramodul apex3(actor1, actor2, actor3) **140** die Darsteller A **142**, B **144** und C **146** in einer gleichseitigen Dreiecksanordnung auf. Das Kameramodul apex3() stellt die virtuelle Kamera **148** in der gleichen relativen Position wie apex() **124** und closeapex() **132** auf, aber es wird immer die Seite der Linie **62** wählen, welche für die aktuelle Aufnahme von Interesse ist.

**[0084]** Wie in [Fig. 12](#) gezeigt, nimmt das Kameramodul external(actor1, actor2) **150** als Eingabe zwei Darsteller A **152** und B **154** an und platziert die virtuelle Kamera **156** so, daß der erste Darsteller A **152** über die Schulter von dem zweiten Darsteller B **154** gesehen wird, wobei der erste Darsteller A **152** zwei Drittel der Aufnahme belegt und der zweite Darsteller B **154** das andere Drittel der Aufnahme belegt. Genauso platziert das Kameramodul external(actor2, actor1) die virtuelle Kamera **156'** so, daß der zweite Darsteller B **154** über die Schulter von dem ersten Darsteller A **152** gesehen wird, wobei der zweite Darsteller B **154** zwei Drittel der Aufnahme belegt und der erste Darsteller A **152** das andere Drittel der Aufnahme belegt. Diese Ansichten des Kameramoduls external() sind in den Kästen **116**, **118** der [Fig. 8](#) illustriert.

**[0085]** Das Kameramodul 2shot(actor1, actor2) (nicht illustriert) ist dem Kameramodul external ähnlich, aber

es verwendet eine Distanzansicht der beiden Darsteller.

[0086] Wie in [Fig. 13](#) gezeigt, implementiert das Kameramodul `external1to2(actor1, actor2, actor3)` **158** eine externe Kameraplatzierung zwischen einem Darsteller A **160** und den beiden anderen Darstellern B **162** und C **164**. Das Kameramodul `external1to2()` platziert die virtuelle Kamera **166** so, daß die beiden Darsteller B **162** und C **164** über die Schulter von dem Darsteller A **160** gesehen werden, wobei die Darsteller B **162** und C **164** zwei Drittel der Aufnahme belegen und Darsteller A **160** den Rest der Aufnahme belegt. Diese Ansicht wird in dem Kasten **108** der [Fig. 8](#) illustriert. Dieses Kameramodul **166** kann auch die Positionen der Darsteller modifizieren, um eine bessere Aufnahme zu komponieren.

[0087] Wie in [Fig. 14](#) gezeigt, stellt das Kameramodul `exclose2(actor1, actor2, actor3)` **168** den Fokus und die Orientierung der virtuellen Kamera **170**, welche sich in der gleichen Position wie die Kamera **166** bei Kameramodul `extern1to2()` **158** befindet, so ein, das nur zwei Darsteller der drei Darsteller in der Aufnahme erscheinen. Beispielsweise erscheinen die Darsteller B **174** und C **176** in der Aufnahme, während der Darsteller A **172** nicht in der Aufnahme erscheint.

[0088] Wie in [Fig. 15](#) gezeigt, kann Kameramodul `internal(actor1, [actor2])` **178** entweder aus einem oder aus zwei Darstellern bestehen. Falls beide Darsteller A **180** und B **182** spezifiziert sind, wird die Kamera **184** entlang der gleichen direkten Sichtverbindung platziert, wie das Kameramodul `external()` **150**, jedoch näher dran und mit einem kleineren Sichtfeld, so daß der erste Darsteller A **180** ganz allein gesehen wird, wobei er zwei Drittel der Darstellungsfläche belegt. Falls ein einzelner Darsteller A **180** spezifiziert ist, wird dann die Linie **62** entlang der Richtung ausgerichtet, in welche der Darsteller A **180** blickt; die Kameraplatzierung **184'** wird auf die gleiche Art relativ zu dieser Linie gewählt.

[0089] Wie in [Fig. 16](#) gezeigt, stellt das Kameramodul `full(actor, theta)` **186** einen Darsteller **188** in das Zentrum der Aufnahme. Die Kameraentfernung zu dem Darsteller **188** wird so eingestellt, daß der ganze Körper des Darstellers **188** gezeigt wird. Der Winkel zwischen der Orientierung der Kamera **190** und der Orientierung des Darstellers **188** ist Theta **192**.

[0090] Die nächsten drei verwandten Kameramodule `tracking()`, `panning()` und `follow()` werden verwendet, wenn sich ein Darsteller bewegt. Sie unterscheiden sich von den gerade beschriebenen, vorhergehenden Kameramodulen darin, daß sie eine sich bewegende Kamera definieren, welche dynamisch die Position und/oder die Orientierung ändert, um eine gewünschte Platzierung des Darstellers nahe des Zentrums der Aufnahme hinsichtlich eines zweiten Darstellers zu halten. Bei den oben stehend beschriebenen Kameramodulen war die Kameraposition fest. Die Kameramodule `tracking()`, `panning()` und `follow()` verwenden minimale und maximale Entfernungseinschränkungen, um zu bestimmen, wie weit die Kamera von der Linie **62** angeordnet wird.

[0091] Wie in [Fig. 17](#) gezeigt, stellt das Kameramodul `tracking(actor1, actor2, mindist, maxdist)` **194** die Kamera **196** entlang einer Linie **198** senkrecht zu der Linie **62** ein. Die Kamera **196'** bewegt sich dann mit dem Darsteller A **200'**, wobei die gleiche Orientierung **198'** zu Darsteller A **200'** bei der Annäherung an Darsteller B **202** erhalten wird.

[0092] Wie in [Fig. 18](#) gezeigt, stellt das Kameramodul `panning(actor1, actor2, mindist, maxdist)` **204** die Kamera **206** von der Linie **62** weg vor dem Darsteller A **208** ein und verschwenkt sie dann am Ort **210**, um der Bewegung des Darstellers A **208'** beim der Annäherung an Darsteller B **212** zu folgen.

[0093] Wie in [Fig. 19](#) gezeigt, vereinigt das Kameramodul `follow(actor1, actor2, mindist, maxdist)` **214** die Kameramoduloperationen `tracking()` **194** und `panning()` **204**. Zunächst verhält es sich wie eine schwenkende Kamera **216**, doch beim Passieren des Darstellers A **218** eines vorbestimmten Punkts **220** beginnt die Kamera **222** dem Darsteller A **218** von hinten zu „folgen“ **222'** (d.h. `tracking`) statt dem Darsteller A **218'** zu gestatten, sich bei der Annäherung an Darsteller B **224** in die Entfernung wegzubewegen.

[0094] Die verbleibenden Kameramodule führen sonstige Funktionen durch. Wie in [Fig. 20](#) gezeigt, wird das Kameramodul `fixed(cameraspec)` **226** verwendet, um einen bestimmten festen Ort, eine Orientierung und ein Sichtfeld **228** für eine Kamera **230** zu spezifizieren. Es wird verwendet, um eine Übersichtsaufnahme einer Szene **232** bereitzustellen.

[0095] Wie in [Fig. 21](#) gezeigt, stellt das Kameramodul `raised()` **234** eine Platzierung relativ zu der vorhergehenden Kameraposition bereit. Die neue Kameraposition **236** ist weiter hinten L2 und höher H2 als die alte Kameraposition **238** bei L1, H1, weist jedoch die gleiche Orientierung **240** zu der Szene **244** wie die vorherge-

hende Kamera **238** auf.

[0096] Wie in [Fig. 22](#) gezeigt, deckt das Kameramodul group(conversation) **246** alle Darsteller **248**, **250**, **252** in einer Unterhaltung ab. Die Aufnahme group() von der Kamera **254** wird durch die graphischen Informationen **256** festgestellt (z.B. des begrenzenden Kastens), welche von dem Echtzeitanwendungsmodul **46** gesendet werden.

[0097] Wie in [Fig. 23](#) gezeigt, synchronisiert das Kameramodul subjective(actor, target) **258** die Kamera **260** auf eine Position auf dem Kopf **262** des Darstellers. Von dem Kopf **262** des Darstellers basiert eine Orientierung auf einem Vektor **264** von dem Kopf **262** des Darstellers zu einem Ziel **266** (z.B. ein anderer Darsteller oder ein Objekt in der Szene).

[0098] Die folgenden drei Kameramodule sind nicht mit individuellen Figuren illustriert, da die Funktion von jedem leicht bestimmt ist. Das Kameramodul subjectzoom(actor) ist dem Kameramodul subjective() **258** ähnlich, doch die Brennweite der Kamera **258** wird mit der Zeit und mit der Bewegung des Darstellers in der Szene länger. Das Kameramodul pov(actor) simuliert die Stirn eines Darstellers. Dieses Kameramodul folgt der Position und Orientierung der Stirn des Darstellers. Das Kameramodul null() läßt die Kamera in ihrer vorhergehenden Position. Dieses Kameramodul gestattet, daß eine Kameraorientierung unverändert bleibt.

#### Auswählen virtueller Kamerapazierungen hinsichtlich der Linie

[0099] Wie oben stehend beschrieben wurde und in [Fig. 4](#) gezeigt ist, wird die Linie **62** relativ zu den beiden Darstellern **64**, **66** in der Aufnahme definiert. Folglich variiert die Linie **62** selbst unvermeidlich von Aufnahme zu Aufnahme. Die Regeln der Filmkinematographie diktieren, daß wenn die Linie **62** konstant bleibt, die Kamera **70** auf der gleichen Seite der Linie **62** bleiben sollte. Hinsichtlich der Linie **62** können die Kameramodule für eine gegebene Spezifikation (z.B. actor<sup>1</sup>, actor<sup>2</sup>) einen von zwei Fällen beschreiben, welche symmetrischen Positionen auf den beiden Seiten der Linie **62** entsprechen. Falls die Linie **62** sich nicht von einer Aufnahme zu der nächsten ändert, ist die Auswahl des bestimmten Falls trivial: wähle die Kameraposition **70** auf der gleichen Seite der Linie **62**.

[0100] Wenn sich die Linie **62** ändert, wenn beispielsweise einer der beiden Darsteller seine Position in der Aufnahme ändert, ist die Auswahl nicht leicht bestimmt. Hinsichtlich der [Fig. 24](#) und gemäß einer beispielhaften Ausführungsform der vorliegenden Erfindung, wird der bestimmte Fall derartig ausgewählt, daß eine Kameraorientierung **268**, welche hinsichtlich einer neuen Linie **270** gewählt wurde, der Orientierung der vorhergehenden Kamera **70** am nächsten ist. Die oben stehend beschriebenen Kameramodule, welche eine Auswahl von mehr als einer Kameraorientierung aufweisen, wählen die neue Linie am nächsten zu der Orientierung der vorhergehenden Kamera. Es können jedoch auch andere Kameraorientierungsverfahren verwendet werden, wenn sich die aktuelle Linie ändert.

#### Beeinflussen des Darstellens mit Kameramodulen

[0101] Wie zuvor angegeben, sind die Kameramodule in der Lage, die Aufnahme durch Beeinflussen der Positionen und des Verhaltensrahmens der Darsteller in der Aufnahme subtil zu verbessern. Da die Echtzeitanwendung **46** primär dazu dient, die Darsteller zu manipulieren, müssen die Änderungen, welche von dem VC **50** vorgenommen werden, subtil genug sein, um die Kontinuität zwischen den Aufnahmen nicht zu stören.

[0102] Beispielsweise bewegt das Kameramodul closeapex() **132** die beiden primären Darsteller näher zusammen, wenn die Entfernung zwischen ihnen größer ist als irgendein Minimum, wie im Kasten **86** der [Fig. 6](#) gezeigt. Das Kameramodul apex3() **140** stellt die Positionen der drei primären Darsteller so ein, daß in der Aufnahme kein Darsteller von einem anderen verdeckt wird, wie in [Fig. 7B](#) gezeigt. Manche Kameramodule können Darsteller vollständig aus der Aufnahme entfernen, um Situationen zu vermeiden, bei welchen ein Darsteller nur teilweise auf der Darstellungsfläche erscheint oder einen anderen primären Darsteller in der Szene verdeckt. Beispielsweise entfernt das Kameramodul internal() **178** den zweiten Darsteller B **182** aus der Aufnahme, wie in [Fig. 15](#) gezeigt, auf beinahe die gleiche Weise, auf welche eine wirkliche interne Umkehraufnahme gefilmt wird.

[0103] Die Kameramodule beeinflussen auch den Verhaltensrahmen eines Darstellers. Beispielsweise den Grad, bis zu welchem sich der Kopf des Darstellers bewegt, wenn er nickt, oder die Beine des Darstellers schwingen, wenn er läuft. Eine Änderung wird auf alle Darsteller in der Aufnahme (nicht nur primäre Darsteller) durch Einstellen eines Parameters angewandt, welcher jedem Darsteller zugeordnet ist und welchen der Auf-

bereiter **54** zum Skalieren der Bewegungen des Darstellers verwenden kann. Eine einfache Funktion  $L$  (Skaliergleichung) wird verwendet, um diesen Parameter einzustellen. Die Funktion steigt monoton an, wenn sich die Größe des Darstellers auf dem Bildschirm vermindert, und wird wie folgt dargestellt:

$$L = \arctan(\text{Abstand} - D_n) \cdot k + 1,0$$

wobei  $D_n$  die Entfernung zu dem Ort ist, an welchem der Darsteller einen normalen Darstellungspegel aufweist (z.B. 1,0) und  $k$  skaliert  $L$  auf innerhalb des Bereichs von 0,5 bis 1,5.

#### Verdeckungserfassung mit Kameramodulen

**[0104]** Kameramodule weisen eine zusätzliche Verantwortlichkeit auf, welche über das Anordnen der virtuellen Kamera hinaus geht. Sie sind auch verantwortlich für ein Erfassen, wenn einer oder mehrere der primären Darsteller in der Szene verdeckt werden (z.B. Darstellerin **Z 98** in [Fig. 7A](#)). Im Fall einer Verdeckung inkrementiert das Kameramodul einen Verdeckungszähler oder setzt den Zähler auf null, wenn der verdeckte Darsteller unverdeckt wird. Dieser Zähler wird von den Idiomen zum Entscheiden verwendet, ob zu einer unterschiedlichen Aufnahme gewechselt wird. Beispielsweise könnte ein Idiom, welches typischerweise eine Aufnahme für ein vorbestimmtes Zeitintervall halten würde (z.B. 20 Zeitzyklen), entscheiden, Aufnahmen schneller zu wechseln, wenn eine Verdeckung erfaßt wird.

#### Verwendung der Kameramodule

**[0105]** Unter Bezugnahme auf [Fig. 3](#), sendet die Echtzeitanwendung **46** in irgendeinem vorbestimmten Zeitintervall (d.h. bei jedem Bild einer Animationssequenz) dem VC **50** eine Beschreibung der Ereignisse, welche in diesem Zeitintervall aufgetreten sind und für einen gewünschten Darsteller signifikant sind. Ereignisse weisen die Form (Subjekt, Verb, Objekt) auf. Beispielsweise bedeutet ein Ereignis (B, talk, A), daß der Darsteller B sich mit dem Darsteller A unterhält, und ein Ereignis (A, move, Bar) bedeutet, daß Darsteller A in die Bar gehen wird. Das Subjekt ist ein Darsteller, während das Objekt ein Darsteller, eine aktuelle Unterhaltung, ein festes Objekt (z.B. die Bar) oder null sein kann. Bei der beispielhaften Ausführungsform können Verben eines aus der Gruppe (idle, talk, react, lookat, join, signal, move) sein, wie nachfolgend erklärt wird. Es können jedoch auch mehr oder weniger Verben verwendet werden, und auch andere Ereignisformen können verwendet werden.

**[0106]** Das VC **50** verwendet die aktuellen Ereignisse plus einen vorhandenen Zustand der Animation (z.B. wie lange die aktuelle Aufnahme gedauert hat), um eine entsprechende virtuelle Kameraspezifikation zu produzieren, welche an das Aufbereitermodul **54** geliefert wird. Zum Produzieren einer virtuellen Kameraspezifikation kann das VC **50** die Echtzeitanwendung **46** für zusätzliche Informationen abfragen **56**, wie beispielsweise einen spezifischen Ort und einen Begrenzungskasten der verschiedenen virtuellen Darsteller. Ein Begrenzungskasten ist ein Kasten, welcher alles oder einen Teil eines virtuellen Darstellers einschließt, und ist Durchschnittsfachleuten bekannt. Das VC **50** kann auch subtile Änderungen an den Positionen und der Bewegung der virtuellen Darsteller vornehmen, was Darstellungshinweise genannt wird. Alle Änderungen an den Positionen der virtuellen Darsteller werden auch an den Aufbereiter **54** geliefert **52**.

**[0107]** Der Aufbereiter **54** bereitet die Szene unter Verwendung der aktuellen Beschreibung der Umgebung, der Animationsparameter **58** für die Darsteller, welche durch die Echtzeitanwendung **46** gesendet werden, und der Kameraspezifikationen und Darstellungshinweise **52**, welche von dem VC **50** gesendet werden, auf, um eine Szene in der Animationssequenz zu produzieren.

#### Idiome

**[0108]** Das VC **50** verwendet Idiome zusätzlich zu den Kameramodulen. Ein einzelnes Idiom codiert die Kompetenz zum Einfangen eines bestimmten Situationstyps, wie beispielsweise eine Unterhaltung zwischen zwei virtuellen Darstellern oder der Bewegung eines einzelnen virtuellen Darstellers von einem Punkt zu einem anderen. Das Idiom ist verantwortlich zum Entscheiden, welche Aufnahmetypen geeignet sind und unter welchen Bedingungen eine Aufnahme in eine andere übergehen sollte. Das Idiom entscheidet auch, wann die Situation sich ausreichend verändert hat, um sich außerhalb der Kompetenzdomäne des Idioms bewegt zu haben, beispielsweise wenn eine dritte Person in eine Unterhaltung von zwei Personen eintritt.

**[0109]** Wie in [Fig. 25](#) gezeigt, ist ein Idiom in dem VC **50** als ein hierarchischer endlicher Automat (FSM) **272** implementiert. Jeder Zustand **274**, **276** innerhalb des FSM **272** ruft ein oder mehrere Kameramodule auf. Folg-

lich entspricht jeder Zustand einer separaten Aufnahme in der erzeugten Animation. Jeder Zustand umfaßt auch eine Liste von Bedingungen, welche, wenn eine erfüllt ist, bewirkt, daß der FSM **272** entlang eines der Wege **278** bis **284** zu einem anderen Zustand gelangt. Beispielsweise bewirkt eine Bedingung C ein Verlassen des Zustands 1 **274** zum Zustand 2 **276** entlang des Wegs **280**. Im Ergebnis wird implizit ein Schnitt erzeugt, wann immer ein Weg in dem FSM **272** durchlaufen wird. Die FSMs sind darin hierarchisch, daß jeder Zustand selbst ein vollständiger FSM mit bestimmten Eingangs- und Ausgangszuständen sein kann, deren FSM ausgeführt wird, wenn in den Zustand eingetreten wird.

[0110] [Fig. 26](#) ist ein Beispiel eines hierarchischen FSM, bei welchem ein Zustand selbst ein FSM ist. Das Idiom A **288** weist zwei Knoten auf: den Knoten 1 **290**; und den Knoten 2 **292**. Die beiden Knoten sind mit Wegen **294** bis **300** verbunden. In diesem Fall wird der Knoten, welcher mit 2 **292** bezeichnet ist, eine Mutter genannt, weil er ein FSM **302** mit vier Tochterzuständen **304** bis **310** und den Wegen **312** bis **326** ist, wie in Idiom B illustriert wird. Es ist zu beachten, daß der Mutterzustand **292** auf die gleiche Weise behandelt wird, wie jeder andere Zustand, ungeachtet ob irgendwelche Tochterzustände eingeschlossen sind.

[0111] Idiome sind durch die folgende Syntax definiert:

```
DEFINE_IDIOM_IN_ACTION(<name>)
```

```
    <actions>
```

```
END_IDIOM_IN_ACTION
```

```
DEFINE_IDIOM_OUT_ACTION(<name>)
```

```
    <actions>
```

```
END_IDIOM_OUT_ACTION
```

wobei <name> der Name eines Idioms (z.B. 2Talk, was zwei sich unterhaltende, virtuelle Darsteller zeigt), und <actions> die Handlungen sind, welche für ein Idiom stattfinden, wenn in dieses eingetreten wird (IN\_ACTION) und bevor das Idiom verlassen wird (OUT ACTION). Ein Beispiel einer Handlung ist WHEN(talking(A, B) GO-TO(1)), was bedeutet, daß wenn sich die virtuellen Darsteller A und B unterhalten, das VC **50** zu Zustand 1 in dem endlichen Automaten für dieses Idiom geht. Handlungen werden in der spezifischen Programmiersprache geschrieben, welche zum Implementieren der Idiome verwendet wird.

[0112] Handlungen für jeden Zustand in dem endlichen Automaten für ein Idiom werden durch die folgende Syntax definiert:

```
DEFINE_STATE_ACTIONS(<name>)
```

```
    <actions>
```

```
END_STATE_ACTIONS
```

wobei <name> der Name einer Handlung ist (z.B. COMMON für Handlungen, welche bei jedem Zustand üblich sind, 1 für Zustand eins) und <actions> die Handlungen sind, welche für den Zustand stattfinden.

[0113] Ein Zustand in dem endlichen Automaten für ein Idiom kann in-actions und out-actions aufweisen. In-actions werden durch die folgende Syntax definiert:

```
DEFINE_STATE_IN_ACTIONS(<state>)
```

```
    <actions>
```

```
END_STATE_IN_ACTION
```

wobei <state> der Name des Zustands ist, für welchen die in-actions ausgeführt werden, und <actions> sind die Handlungen in-action. In-actions werden verwendet, wenn ein bestimmter Zustand in dem endlichen Automaten in einem Idiom eine spezielle Ausführung erfordert, wenn in den Zustand eingetreten wird. In-actions sind nicht erforderlich und können null für alle oder die meisten Zustände in dem endlichen Automaten eines Idioms sein. Out-actions werden durch die folgende Syntax definiert:



```
DEFINE_STATE_OUT_ACTIONS(<state>)
```

```
    <actions>
```

```
END_STATE_OUT_ACTION
```

wobei <state> der Name des Zustands ist, für welchen die out-actions ausgeführt werden, und <actions> sind die Handlungen out-action. Out-actions werden verwendet, wenn ein bestimmter Zustand in dem endlichen Automaten in einem Idiom eine spezielle Ausführung erfordert, wenn der Zustand verlassen wird. Out-actions sind nicht erforderlich und können null für alle oder die meisten Zustände in dem endlichen Automaten eines Idioms sein.

**[0114]** Die oben stehend beschriebenen Kameramodule (z.B. external() **150**) werden von Kameraeinstellungsmodulen definiert. Kameraeinstellungsmodule weisen die folgende Syntax auf:

```
DEFINE_SETUP_CAMERA(<idiom>)
```

```
    <camera actions>
```

```
END_SETUP_CAMERA_MODULES
```

wobei <idiom> der Name eines Idioms ist und <camera actions> die Handlungen zum Einstellen der Kameramodule für das Idiom sind. Bei einer Ausführungsform der vorliegenden Erfindung werden zwei Typen von Kameraeinstellungsmodulen verwendet: Erzeugungsmodule und Verknüpfungsmodule.

**[0115]** MAKE\_MODULE(<module id>, <type>, <parameter list>) erzeugt ein neues Kameraeinstellungsmodul des angegebenen Typs mit den spezifizierten Parametern und gibt ihm eine spezifizierte, identifizierende Modulnummer. LINK\_MODULE(<state>, <module id>, <name>) ordnet den spezifizierten Zustand dem spezifizierten Kameramodul zu.

**[0116]** Bei einer Ausführungsform der vorliegenden Erfindung wird Idiom-Code beim Programmieren von Makros in der Programmiersprache C++ geschrieben. Handlungs- und andere Anweisungen, welche innerhalb der Idioms definiert sind, werden unter Verwendung einer C/C++ Programmsyntax geschrieben. Es können jedoch auch andere Programmiersprachen verwendet werden. Die Schlüsselworte, welche in Großbuchstaben geschrieben sind (z.B. MAKE\_MODULE), sind Programmakros. Idioms können jedoch auch ohne Verwendung von Programmakros implementiert werden. Eine Liste von Programmakros, welche zum Definieren von Idioms verwendet werden, erscheint in Anhang A.

**[0117]** Wenn Quell-Code, welcher die Idiomakros enthält, kompiliert wird, werden die Makros durch den C-Vorprozessor (C++ wird typischerweise als eine Erweiterung der Programmiersprache C implementiert) in C/C++-Quell-Code erweitert. Die Definition jedes Idiomakros umfaßt einen oder mehrere Aufrufe von C++-Prozeduren, welche die gewünschten Handlungen für das Idiomakro tatsächlich erzielen. Beispielsweise wird das Makro DEFINE\_SETUP\_CAMERA\_MODULES() zu einem Aufruf einer C++-Routine SetupCameraModules() erweitert, um die Kameramoduleinstellung zu erzielen.

**[0118]** Bei einer Ausführungsform der vorliegenden Erfindung gibt es einige globale Variablen, welche bei Zustandshandlungen verwendet werden können und welche Folgendes umfassen: Occluded, die Anzahl aufeinanderfolgender Zeiteinheiten, welcher einer oder mehrere der primären Darsteller verdeckt war; T, die Anzahl Zeitzyklen oder -Einheiten in dem aktuellen Zustand; IdiomT, die Gesamtanzahl Zeiteinheiten, welche bis jetzt in diesem Idiom als Ganzes verbraucht wurde; D[A, B], die Entfernung zwischen den Darstellern (gemessen in Einheiten von „Kopfdurchmessern“, welche nachfolgend erklärt werden); und forwardedge[x], rearedge[x], centerline[x], die Ränder des Begrenzungskastens des virtuellen Darstellers x relativ zu den Koordinaten der Darstellungsfläche. Es können jedoch auch mehr oder weniger globale Variablen verwendet werden.

**[0119]** Es gibt auch eine Anzahl vordefinierter Steuerungsstrukturen, welche bei einer Ausführungsform der vorliegenden Erfindung verwendet werden: STAY, was bedeutet, für eine weitere Zeiteinheit in dem gleichen Zustand zu verbleiben; GOTO(x), was einen Übergang zu einem Zustand x bedeutet; RETURN, was eine Rückkehr zu dem Mutterzustand bedeutet; und CALL (idiom, <Parameter List>), was ein Ausführen des spezifizierten Idioms durch Übergeben der spezifizierten Parameterliste bedeutet. Es können jedoch auch mehr oder weniger vordefinierte Steuerungsstrukturen verwendet werden.

**[0120]** Die Konzepte, welche beim Konstruieren von Idioms beteiligt sind, werden durch Beispiele illustriert.



Bei dem ersten Beispiel wird ein Idiom zum Darstellen einer Unterhaltung zwischen zwei virtuellen Darstellern, 2Talk genannt, illustriert. Bei dem zweiten Beispiel wird das Idiom 2Talk als ein C++-Primitividiom zum Herstellen eines komplexeren Idioms, 3Talk genannt, zum Darstellen einer Unterhaltung unter drei virtuellen Darstellern verwendet.

#### Das Beispielidiom 2Talk

**[0121]** Das Beispielidiom 2Talk codiert ein Verfahren zum Filmen von zwei virtuellen Darstellern, welche sich unterhalten und aufeinander reagieren. Es verwendet nur externe Umkehraufnahmen (**72, 84**) (**Fig. 5**) der beiden Darsteller. **Fig. 27** zeigt den endlichen Automaten 2Talk **328**. Die Prozedur 2Talk nimmt als Parameter zwei virtuelle Darsteller A und B, welche sich unterhalten.

**[0122]** Er weist vier Zustände **330** bis **336** auf. Der erste Zustand **330** verwendet ein Kameramodul external **150** (**Fig. 12**), welches A zeigt, welcher mit B spricht. Der zweite Zustand **332** wird für die umgekehrte Situation, wenn B mit A spricht, verwendet. Der dritte und der vierte Zustand **334, 336** verwenden Anordnungen des Kameramoduls external **150**, um Reaktionsaufnahmen von jedem der Darsteller einzufangen.

**[0123]** Wenn das Idiom 2Talk **328** aktiviert ist, folgt es einem der beiden Anfangswege **338, 340**, welche an dem kleinen Kreis beginnen, welcher ein „Eingangspunkt“ **342** genannt wird. Der zu verwendende Weg (**338** oder **340**) wird durch den folgenden Quell-Code bestimmt:

```
DEFINE_IDIOM_IN_ACTION(2Talk)
    WHEN (talking(A, B))
        DO (GOTO(1);)
    WHEN (talking(B, A))
        DO (GOTO(2);)
END_IDIOM_IN_ACTION
```

**[0124]** Dieser Idiom-Code prüft, ob A mit B spricht oder ob B mit A spricht und geht sofort zu dem entsprechenden Zustand in dem Idiom 2Talk über (**338, 340**), in diesem Fall entweder Zustand 1 bzw. 2 (**330, 332**).

**[0125]** Beim Eintreten in einen Zustand führt er einen Satz von in-actions für den Zustand aus. Die in-actions sind oft null, wie es für alle Zustände in dem Beispielidiom 2Talk der Fall ist. Wenn einmal in einen Zustand eingetreten wurde, wird das Kameramodul des Zustands (z.B. external() **150**) aufgerufen, um die virtuelle Kamera zu positionieren. Der Zustand führt dann eine Sequenz von Handlungen in einem vorbestimmten Zeitintervall T (z.B. jedem Taktzyklus) aus. Diese Handlungen umfassen eine Ausführung von C++-Prozeduren, wie oben stehend beschrieben. Zusätzlich können die Handlungen verwendet werden, um bedingte Übergänge (**344** bis **358**) zu anderen Zuständen zu beeinflussen. Wenn der Zustand schließlich verlassen wird, führt es einen Satz von out-actions aus, welche in dem Fall des Idioms 2Talk wieder null sind.

**[0126]** In dem Idiom 2Talk werden die Kameraeinstellungsmodule wie folgt definiert:

```
DEFINE_SETUP_CAMERA_MODULES(2Talk)
MAKE_MODULE(1, external, (A, B))
MAKE_MODULE(2, external, (B, A))
LINK_MODULE(1, 1, „A talks“)
LINK_MODULE(2, 2, „B talks“)
LINK_MODULE(3, 1, „A reacts“)
LINK_MODULE(4, 2, „B reacts“)
END_SETUP_CAMERA_MODULES
```

**[0127]** Beispielsweise erzeugt MAKE\_MODULE(1, external, (A, B)) ein Kameramodul vom Typ external() **150** mit zwei virtuellen Darstellern A und B und ordnet ihm die Identifikationsnummer 1 zu. LINK\_MODULE(1, 1, „A talks“) bedeutet, daß wann immer in den Zustand 1 eingetreten wird, eine externe Umkehraufnahme des Darstellers A (d.h. „A talks“) über die Schulter von Darsteller B verwendet wird, wie durch MAKE\_MODULE(1, external (A, B)) definiert. LINK\_MODULE(3, 1, „A reacts“) bedeutet, daß in Zustand 3 der Darsteller A (z.B. durch Nicken) auf die Unterhaltung mit Darsteller B reagiert, wie durch MAKE\_MODULE(1, external(A, B)) definiert.

**[0128]** Der erste in jedem Zustand auszuführende action-Code kann in einem Block spezifiziert werden, welcher allen Zuständen gemeinsam ist. Dies ist primär ein Kurzmechanismus, um zu vermeiden, daß die gleichen Paare (Bedingung, Weg) in jedem Zustand des Idioms erneut spezifiziert werden müssen. Die Idiomatiken können jedoch auch ohne einen Block verwendet werden, welcher allen Zuständen gemeinsam ist.

**[0129]** Gemeinsame Handlungen in dem Idiom 2Talk sind:

```
DEFINE_STATE_ACTIONS(COMMON)
    WHEN (T < 10)
        DO (STAY;)
    WHEN (!talking(A, B) && !talking(B, A))
        DO (RETURN;)
END_STATE_ACTIONS
```

**[0130]** Es können jedoch auch mehr oder weniger gemeinsame Handlungen verwendet werden. Ein gemeinsamer Block wird mit der gleichen Syntax definiert wie jeder andere nicht gemeinsame Zustand.

**[0131]** Die erste Anweisung (WHEN (T < 10)) setzt eine Prüfung in Gang, um zu sehen, ob die so weit in diesem Zustand verbrachte Gesamtzeit T geringer als 10 Zeiteinheiten (z.B. 10 Zyklen des Taktgebers) ist. Falls weniger als 10 Zeiteinheiten abgelaufen sind, bleibt der aktuelle Zustand unverändert. Ein EXCEPTION-Mechanismus, welcher nachfolgend erklärt wird, hat Vorrang über das Prüfen der abgelaufenen Zeit und kann die Aufnahme unterbrechen. Falls die Aufnahme mindestens zehn Zyklen angedauert hat, A und B sich jedoch nicht mehr länger unterhalten, kehrt das Idiom dann zu dem Idiom zurück, welches es aufgerufen hat. Die Variable T ist eine globale Variable, welche jedem Zustand zugänglich ist.

**[0132]** Der actions-Code in dem gemeinsamen Block verwendet eine Domänen-spezifische Prozedur, welche talking(A, B) genannt wird und welche wahr zurückgibt, falls und nur falls die aktuelle Ereignisliste (A, talk, B) umfaßt (d.h. der virtuelle Darsteller A unterhält sich mit dem Darsteller B). In dem oben stehend gezeigten Code ist das Ausrufezeichen „!“ der NICHT-Operator von C/C++, und das doppelte Kaufmannsund „&&“ ist der UND-Operator von C/C++. Die Handlungsanweisungen werden der Reihe nach beurteilt. Folglich haben frühere Anweisungen Vorrang über Anweisungen, welcher später in dem Code aufgeführt werden.

**[0133]** Zustand 1 **330** des Idioms 2Talk **328** wird verwendet, um den Darsteller A abzubilden, welcher sich mit B unterhält. Zusätzlich zu den gemeinsamen Handlungen ist die Liste der Handlungen, welche in dem Zustand 1 **330** zu jeder Zeiteinheit T ausgeführt werden, wie folgt:

```
DEFINE_STATE_ACTIONS(1)
    WHEN (talking(B, A))
        DO (GOTO(2);)
    WHEN (T > 30)
        DO (GOTO(4);)
END_STATE_ACTIONS
```

**[0134]** Es können jedoch auch mehr oder weniger Handlungen verwendet werden. Falls B sich jetzt mit A unterhält, dann ist ein Übergang **344** zum Zustand 2 **332** erforderlich, um diese Situation einzufangen. Falls sich ein Darsteller für mehr als 30 Zeiteinheiten T in der gleichen Aufnahme befunden hat, sollte es einen Übergang **356** zum Zustand 4 **336** geben, um eine Reaktionsaufnahme von dem anderen Darsteller zu bekommen.

**[0135]** Der Zustand 2 **332**, welcher den Fall anspricht, daß sich der Darsteller B mit dem Darsteller A unterhält, ist vollständig symmetrisch: der Code ist genau der gleiche wie für den oben stehend gezeigten Zustand 1 **330**, außer daß A und B vertauscht sind (z.B. talking(A, B) und die Zustände 1 **330** und 3 **334** an Stelle der Zustände 2 **332** und 4 **336** verwendet werden (z.B. DO(GOTO(1)) und DO(GOTO(3))).

```
DEFINE_STATE_ACTIONS(2)
```

```
    WHEN (talking(A, B))
```

```
        DO (GOTO(1);)
```

```
    WHEN (T > 30)
```

```
        DO (GOTO(3);)
```

```
END_STATE_ACTIONS
```

**[0136]** Der action-Code für den Zustand 3 **334** wird nachfolgend gezeigt: Es können jedoch auch mehr oder weniger Handlungen für den Zustand 3 **334** verwendet werden.

```
DEFINE_STATE_ACTIONS(3)
```

```
    WHEN (talking(A, B))
```

```
        DO (GOTO(1);)
```

```
    WHEN (talking(B, A) || T > 15)
```

```
        DO (GOTO(2);)
```

```
END_STATE_ACTIONS
```

**[0137]** Falls Darsteller A sich jetzt mit B unterhält, dann wird Übergang **352** zu Zustand 1 **330** erforderlich, um diese Situation einzufangen. Falls Darsteller B sich jetzt mit dem Darsteller A unterhält oder „||“ falls sich ein Darsteller für mehr als 15 Zeiteinheiten T in der gleichen Aufnahme befunden hat, sollte es einen Übergang **350** zu Zustand 2 **332** geben.

**[0138]** Es ist zu beachten, daß Zustand 3 **334** einen Übergang **352** zurück zu Zustand 1 **330** vornehmen kann, welcher das gleiche Kameramodul verwendet, wie es hier in Zustand 3 **334** verwendet wurde. In diesem Fall werden die beiden Aufnahmen ohne einen Schnitt zu einer einzelnen Aufnahme vereinigt.

**[0139]** Schließlich ist der Zustand 4 **336** symmetrisch zu Zustand 3 **334**.

```
DEFINE_STATE_ACTIONS(4)
```

```
    WHEN (talking(A, B))
```

```
        DO (GOTO(1);)
```

```
    WHEN (talking(B, A) || T > 15)
```

```
        DO (GOTO(2);)
```

```
END_STATE_ACTIONS
```

**[0140]** Da die out-actions für das Idiom 2Talk **328** null sind, wurde das Idiom 2Talk **328** vollständig beschrieben. Es können auch mehr oder weniger Zustände und mehr, weniger oder alternative Handlungen innerhalb jedes Zustands verwendet werden, um das Idiom 2Talk zu definieren. Das Idiom 2Talk **328** kann jetzt als Unteroutine für Idiome eines höheren Levels (z.B. zum Behandeln von Unterhaltungen mit drei Darstellern) verwendet werden.

#### Das Beispielidiom 3Talk

**[0141]** Das Beispielidiom 3Talk wird aus dem Idiom 2Talk **328** erzeugt. Der endliche Automat für das Idiom 3Talk **360**, welcher Unterhaltungen unter drei virtuellen Darstellern behandelt, wird in [Fig. 28](#) gezeigt. Dieses Idiom **360** implementiert die kinematische Behandlung von drei Darstellern, welche von Arijon beschrieben und in [Fig. 8](#) illustriert wurde. Der FSM 3Talk weist die gleichen Typen von Komponenten auf wie 2Talk **328**: er weist Zustände **362** bis **368** und Wege **370** bis **390** auf, welche Übergänge zwischen Zuständen repräsentieren. Zusätzlich verwendet dieser FSM einen EXCEPTION-Mechanismus, wie nachfolgend diskutiert wird.

[0142] Das Idiom 3Talk weist vier Zustände **362** bis **368** auf. Der erste Zustand **362**, 1 bezeichnet, ist eine Einführungsaufnahme aller drei Darsteller entsprechend der ersten Kameraposition **100** in [Fig. 8](#). Der zweite Zustand **364**, 2AND3 bezeichnet, ist ein Mutterzustand, welcher das Idiom 2Talk **328** aufruft (CALL), und entspricht den Kameras 2 **110** und 3 **112** in [Fig. 8](#). Die letzten beiden Zustände, 4A **366** und 4B **368**, fangen schließlich die Reaktionsaufnahme des ersten Darstellers ein; Diese beiden Zustände entsprechen der Kameraaufnahme 4 **122** der [Fig. 8](#).

[0143] Alle vier Zustände weisen Handlungen auf, welche denjenigen ähnlich sind, welche in 2Talk **328** beschrieben wurden. Die beiden Zustände 4A **366** und 4B **368** wurden als separate Zustände implementiert, da sie in dem Idiom 3Talk unterschiedlich funktionieren, sogar obwohl sie die Szene beide aus der gleichen virtuellen Kamera drehen. Der Zustand 4A **366** wird in der Eröffnungssequenz oder nach einer neuen Einführungsaufnahme verwendet, wobei gestattet wird, daß sich Aufnahmen aller drei Darsteller mit Reaktionsaufnahmen des Darstellers A abwechseln. Im Gegensatz dazu wird der Zustand 4B **368** nur verwendet, wenn eine Zweierunterhaltung zwischen Darsteller B und C einmal dominant wird, um eine gelegentliche Reaktion des Darstellers A zu bekommen und dann schnell zu der Zweierunterhaltung zwischen B und C zurückzukehren.

[0144] Der eine Zustand, welcher sich von den zuvor betrachteten Zuständen unterscheidet, ist der Zustand **364**, welcher 2AND3 bezeichnet wird. Anders als die vorhergehenden Zustände weist der Zustand 2AND3 **364** zunächst in-actions auf, welche durch den folgenden Makro-Code illustriert werden:

```

        DEFINE_STATE_IN_ACTION(2and3)
REGISTER_EXCEPTION(left_conversation, A, LEFT_CONVERSATION);
        REGISTER_EXCEPTION(too_long, 100, TOO_LONG);
REGISTER_EXCEPTION(reacts, A, GET_REACTION);
        CALL(2Talk, (B, C));
        END_STATE_IN_ACTION

```

[0145] Diese in-actions registrieren eine Anzahl von EXCEPTIONS, welche beim Auslösen bewirken, daß ein Tochteridiom verlassen wird und die Steuerung zu dem Mutterzustand zurückkehrt.

[0146] Zum Registrieren einer EXCEPTION wird ein Makro REGISTER\_EXCEPTION() verwendet. Jeder Befehl REGISTER\_EXCEPTION(<procedure>, <parameters> <test>, <exception name>) übernimmt 3 Parameter: <procedure>, den Namen einer Prozedur, um zu prüfen, ob die Ausnahmebehandlung ausgelöst werden soll oder nicht; <parameters>, einen beliebigen Parametersatz, welcher an diese Funktion weitergegeben wird; und den <exception name>, welcher ein Aufzählungstyp ist. Beispielsweise registriert REGISTER\_EXCEPTION(too\_long, 100, TOO\_LONG); die EXCEPTION TOO\_LONG für einen Aufruf der Prozedur too\_long() mit einem Parameter von 100 (d.h. der Anzahl Zeiteinheiten T, um auf eine Zeitspanne zu prüfen, welche TOO\_LONG ist).

[0147] Die finale in-action-Handlung (d.h. CALL(2Talk, (B, C)) des Zustands 2AND3 **364** ruft das Idiom 2Talk **328** auf, wobei ihm die Darsteller B und C als Parameter weitergegeben werden. Alle registrierten Ausnahmebehandlungen werden implizit geprüft, bevor die Handlungen in jedem Zustand des Tochteridioms ausgeführt werden.

[0148] Das Idiom 2Talk **328** kehrt zurück, entweder wenn es ein RETURN in einer seiner Handlungen ausführt oder wenn eine der EXCEPTIONS ausgelöst wird. An diesem Punkt wird die Steuerung an den Mutterzustand zurückgegeben, und seine Handlungen werden ausgeführt. Die Handlungen für den Zustand 2AND3 **364** sind wie folgt:

```

DEFINE_STATE_ACTIONS(2and3)
    WHEN (EXCEPTION_RAISED(LEFT_CONVERSATION))
        DO (GOTO(1);)
    WHEN (EXCEPTION_RAISED(TOO_LONG))
        DO (GOTO(1);)
    OTHERWISE
        DO (GOTO(4b);)
END_STATE_ACTION

```

**[0149]** In diesem Fall, falls entweder die EXCEPTION LEFT\_CONVERSATION oder TOO\_LONG ausgelöst wurde, wird dann ein Übergang **378** zurück zu dem Zustand 1 **362** vorgenommen, um eine weitere Einführungsaufnahme zu bekommen. Andernfalls wird ein Übergang **378** zu dem Zustand 1 **362** vorgenommen, um eine Reaktionsaufnahme zu bekommen.

**[0150]** Zusätzlich zu in-actions weist der Zustand 2AND3 out-actions auf. Die out-actions des Zustands 2and3, welche unmittelbar vor einem Vornehmen des Übergangs zu dem neuen Zustand beurteilt werden, werden verwendet, um die EXCEPTIONS zu entfernen, welche von den in-actions eingerichtet wurden. Die out-actions werden durch den folgenden Makro-Code illustriert:

```

DEFINE_STATE_OUT_ACTION(2and3)
    DELETE_EXCEPTION(LEFT_CONVERSATION);
    DELETE_EXCEPTION(TOO_LONG);
    DELETE_EXCEPTION(GET_REACTION);
END_STATE_OUT_ACTION

```

**[0151]** Diese out-actions entfernen die EXCEPTIONS für LEFT\_CONVERSATION, TOO\_LONG, und GET\_REACTION, welche in den in-actions registriert wurden und eventuell in den Zustandshandlungen für den Zustand 2and3 ausgeführt wurden.

#### Idiomhierarchie

**[0152]** Bei einer Ausführungsform der vorliegenden Erfindung wird eine Idiomhierarchie erzeugt, welche mit individuellen Idiomen beginnt. Wie in dem Ablaufdiagramm in [Fig. 29](#) gezeigt, wird ein Verfahren **392** zum Erzeugen eines individuellen Idioms verwendet. Ein individuelles Idiom wird durch Definieren der Zustände in einem Idiom **394** und durch Auswählen der Kameramodule erzeugt, welche verwendet werden, um die definierten Zustände des Idioms **396** anzuschauen. Dann werden Handlungen definiert, welche Übergänge von einem definierten Zustand zu einem anderen **398** umfassen. Dieses Verfahren **392** wurde verwendet, um das Idiom 2Talk **328** zu erzeugen. Das Idiom 2Talk **328** wurde durch Definieren **394** von vier Zuständen **330** bis **336** erzeugt, welche die 4 Kameraaufnahmen (**100**, **110**, **112**, **120**) repräsentieren, welche in [Fig. 8](#) gezeigt sind. Die vier Zustände **330** bis **336** werden durch Auswählen **396** des Kameramoduls external() **150** angeschaut, um alle vier Zustände anzuschauen. Dann wurden Handlungen für die vier Zustände definiert **398**, welche Übergänge aus einem Zustand zu einem anderen bewirken.

**[0153]** Wie in dem Ablaufdiagramm in [Fig. 30](#) gezeigt, wird ein anderes Verfahren **400** zum Erzeugen eines komplexen Idioms verwendet, wenn individuelle Idiome einmal mit dem Verfahren **392** erzeugt werden. Das komplexe Idiom wird in einer ähnlichen Weise erzeugt, wie das Erzeugen eines individuellen Idioms. Die Zustände des komplexen Idioms werden definiert **402**. Für einen oder mehrere der Zustände wird ein individuelles Idiom ausgewählt, um die Handlungen des Zustands **404** zu definieren. Es werden Kameramodule, welche verwendet werden, um die definierten Zustände des Idioms anzuschauen, für Zustände ausgewählt, für welche kein individuelles Idiom ausgewählt wurde (falls vorhanden) **406**. Es werden Handlungen definiert, welche Übergänge zwischen den definierten Zuständen **408** umfassen. Dieses Verfahren **400** wurde verwendet, um das Idiom 3Talk **360** zu erzeugen. Wie oben stehend beschrieben, ist 3Talk ein Idiom, welches gestattet, daß

sich drei virtuelle Darsteller unterhalten. Da ein individuelles Idiom für eine Unterhaltung für zwei virtuelle Darsteller bereits erzeugt worden ist (d.h. 2Talk **328**), wurde das Idiom 2Talk in dem Zustand verwendet, welcher ein Anschauen einer Unterhaltung zwischen nur zwei der virtuellen Darsteller erfordert.

**[0154]** Wie in dem Ablaufdiagramm in [Fig. 31](#) gezeigt, wird wieder ein anderes Verfahren **410** zum Erzeugen einer Idiomhierarchie verwendet. Dieses Verfahren **410** umfaßt das Auswählen von zwei oder mehr individuellen Idiomen **412** und das Vereinigen der ausgewählten individuellen Idiome, um ein komplexes Idiom **414** zu erzeugen. Es werden ein oder mehrere komplexe Idiome und ein oder mehrere individuelle Idiome ausgewählt **416**. Das ausgewählte komplexe und individuelle Idiom werden vereinigt, um ein Verb-Idiom **418** zu bilden. Die Verb-Idiome sind High-Level-Idiome in der Idiomhierarchie. Beispielsweise wird das Verb „converse“ mit einer Kombination aus dem individuellen Idiom 2Talk und dem vereinigten Idiom 3Talk erzeugt. Das Verb-Idiom „converse“ gestattet eine Unterhaltung zwischen zwei oder drei virtuellen Darstellern unter Verwendung einer Kameraaufnahme external() **150**.

**[0155]** [Fig. 32](#) zeigt einen Abschnitt einer Idiomhierarchie **420** für das Verb „converse“. Das Verb „converse“ **422** ist eine Kombination aus dem Idiom 3Talk **424** und dem Idiom 2Talk **426**. Dieses Idiom 3Talk (anders als das oben stehend beschriebene Idiom 3Talk **360**) verwendet die Kameramodule external() und internal() **428**, **430** und ruft das Idiom 2Talk **426** auf. Das Idiom 2Talk **426** verwendet nur das Kameramodul external() **428**. Andere Verb-Idiome weisen eine ähnliche Hierarchie auf, welche Kameramodule auf dem niedrigsten Level, individuelle Idiome über den Kameramodulen, komplexe Idiome über den individuellen Idiomen und Verb-Idiome über den komplexen Idiomen umfassen.

#### Individueller Regiestil unter Verwendung der Idiomhierarchie

**[0156]** Andere Verb-Idiome oder andere High-Level-Idiome werden auf eine ähnliche Weise aus individuellen Low-Level-Idiomen erzeugt. Beispielsweise ist ein Verb-Idiom „move“ eine Kombination aus einem individuellen Idiom 1Move und einem komplexen Idiom 2Move. Das individuelle Idiom 1Move gestattet, daß sich ein virtueller Darsteller bewegt und verwendet die Kameramodule tracking() **194**, panning() **204** und follow() **214**. Das komplexe Idiom 2Move wurde aus dem individuellen Idiom 1Move erzeugt und gestattet, daß sich zwei virtuelle Darsteller bewegen. Die Verfahren **392**, **400**, und **410** können verwendet werden, um eine praktisch unbegrenzte Anzahl von Idiomen zu erzeugen. Bei einer Ausführungsform der vorliegenden Erfindung können Verb-Idiome aus der Gruppe (idle, converse, react, lookat, join, signal, move) sein. Es können jedoch auch mehr oder weniger Verben verwendet werden.

**[0157]** Beim Erzeugen von Gruppen oder Bibliotheken von Verb-Idiomen oder von anderen High-Level-Idiomen kann ein Regisseur die virtuellen Darsteller leicht manipulieren, um einen virtuellen Film „aufzunehmen“. Der virtuelle Film würde mit Verfahren und Kamerawinkeln „gedreht“, welche praktisch identisch mit den Verfahren und Kamerawinkeln sind, welche für einen wirklichen Film verwendet werden. Die Verb-Idiome können leicht zugeschnitten werden, um zu einem individuellen Regiestil zu passen. Beispielsweise kann Regisseur A das Verb-Idiom „converse“ verwenden, wie oben stehend beschrieben. Regisseurin B kann jedoch auf der Grundlage ihres persönlichen Regiestils wünschen, daß ein Kameramodul internal() **178** in dem individuellen Idiom 2Talk **328** an Stelle des Kameramoduls external() **150** verwendet wird. Regisseurin B kann das Idiom 2Talk anpassen, um das Kameramodul internal() **178** zu verwenden, um ihre persönlichen Vorlieben zu erfüllen.

**[0158]** Wie in dem Ablaufdiagramm in [Fig. 33](#) gezeigt, verwendet ein Verfahren **432** zur virtuellen Regie die oben stehend beschriebenen Idiome und Kameramodule, um einen virtuellen Film zu „drehen“. Das VC wählt das Idiom auf der Grundlage der Ereignisse (Subjekt, Verb, Objekt) und den Run-Time-Zustand aus einer Gruppe (z.B. einer Bibliothek) von Verb-Idiomen **434** automatisch aus. Die ausgewählten Verb-Idiome werden vereinigt, um virtuelle Szenen **436** zu erzeugen. Die virtuellen Szenen werden vereinigt, um einen virtuellen Film **438** zu erzeugen. Die Regisseurin ist frei, um einen virtuellen Film auf der Grundlage ihres eigenen Stils unter Verwendung der VC-Kameramodule und der Idiomhierarchie zu erzeugen.

#### Bewegung der virtuellen Darsteller auf der Darstellungsfläche

**[0159]** Ein Einfangen von Bewegung auf einer Videodarstellungsfläche, welche verwendet wird, um die Animation mit den virtuellen Darstellern zu präsentieren, stellt spezielle Probleme dar. Es kann insbesondere wünschenswert sein, eine Aufnahme nicht nur dann zu beenden, wenn ein Ereignis durch das Echtzeitsystem ausgelöst wird, sondern auch wenn ein Darsteller eine bestimmte Position auf dem Bildschirm erreicht (wie beispielsweise den Rand der Darstellungsfläche). Die oben stehend beschriebenen globalen Variablen forward-



edge[x], rearedge[x], centerline[x] werden verwendet, um diese Art von Prüfungen zu erleichtern.

[0160] Wie in [Fig. 34](#) gezeigt, werden diese Variablen in einem Darstellungsflächen-Koordinatensystem **440** gemessen, welches relativ zu der Orientierung und der Position jedes Darstellers **442** eingerichtet ist. Der Rand der Darstellungsfläche, zu welchem der Darsteller blickt, wird bei +1 **444** definiert, während der Rand hinter dem Darsteller bei -1 **446** ist. Die Mittellinie der Darstellungsfläche ist bei null **448**. Folglich kann beispielsweise ein Zustand in einem Idiom bestimmen, ob der Darsteller **442** gerade den Rand der Darstellungsfläche erreicht hat, indem er prüft, ob forwardedge[x] größer als 1 ist. Ein Zustand kann auch prüfen, ob der Darsteller vollständig aus der Darstellungsfläche herausgelaufen ist, indem er prüft, ob rearedge[x] kleiner als -1 ist. Es können auch andere Verfahren und Koordinatensysteme verwendet werden, um Idiome zum Einfangen von Bewegung auf einer Darstellungsfläche zu entwickeln.

#### Party-Anwendung

[0161] Die Verfahren **34**, **392**, **400**, **410**, **432** und das System **44** werden auf eine simulierte „Party“-Umgebung angewandt. [Fig. 35](#) zeigt eine Benutzerschnittstelle **450** für einen Party-Simulator, welcher einem Teilnehmer einer Party vorgelegt wird. Die Party findet in einem virtuellen Raum statt, welcher von zwei virtuellen Bars mit virtuellen Barkeepern und jeweils zwei virtuellen Barhockern, einer Statue und anderen virtuellen Gästen bevölkert ist. Die Gäste können herumgehen, sich umschauchen, sich miteinander unterhalten, zu den Bars gehen, wo sie trinken oder sich mit den Barkeepern unterhalten können, oder sie können einfach gar nichts tun.

[0162] Ein Teilnehmer steuert die Handlungen eines virtuellen Darstellerprotagonisten **452** auf einer hohen Abstraktionsebene. Der Rest der Darsteller **454** kann automatisch als Software-Agenten oder durch andere Teilnehmer der Party gesteuert werden. Insbesondere kann der Teilnehmer (Verb, Objekt)-Paare aufrufen, welche in (Subjekt, Verb, Objekt)-Tripel übersetzt werden, bei welchen der Darstellerprotagonist **452** das Subjekt ist. Aktuelle Verben umfassen talk, react (welches bewirkt, daß der Darsteller nickt), goto, drink (welches bewirkt, daß der Darsteller zum nächsten leeren Barhocker geht), lookat (welches bewirkt, daß der Darsteller seinen Kopf dreht, das gewählte Objekt für einige Sekunden anstarrt und dann seinen Kopf zurückdreht) und idle.

[0163] Jeder Aufruf eines Verbs bewirkt eine Änderung der Handlung des Protagonisten, kurz nachdem die entsprechende Taste gedrückt wird. Die Teilnehmer der Party sind dadurch „höflich“, daß sie warten, bis ein Darsteller in einer Unterhaltung zu sprechen aufhört, bevor sie selber zu sprechen beginnen.

[0164] Eine zusätzliche Schnittstellentaste **456** gestattet den Darstellern, welche alleine sind oder sich in einer Unterhaltung befinden, zu „wählen“, ob sie einen Darsteller akzeptieren oder zurückweisen, welcher einen Wunsch signalisiert, an der Unterhaltung teilzunehmen. Das Verb signal wird implizit erzeugt, wenn ein Darsteller sich auf eine kurze Entfernung dem Ziel des goto angenähert hat. Andere Tasten der Benutzerschnittstelle **458**, **460** ermöglichen eine zusätzliche Funktionalität zum Manipulieren der virtuellen Darsteller.

[0165] Zu jeder Zeiteinheit, welche hinsichtlich der maximalen Bildrate festgestellt wird und 5 Hz beträgt, sendet der Party-Simulator bei einer Implementierung eine Liste der interessierenden (Subjekt, Verb, Objekt)-Tripel an den privaten VC des Protagonisten. Interessierende Tripel sind diejenigen, welche den Protagonisten (oder andere in der gleichen Unterhaltung wie der Protagonist) als Subjekt oder als Objekt einbeziehen.

[0166] Der Party-Simulator ist für alle Low-Level-Bewegungen der Darsteller verantwortlich. Dies umfaßt Umhergehen, Mundbewegung beim Sprechen sowie Drehen des Kopfs, um den aktuellen Sprecher bei einer Unterhaltung anzusehen oder in Richtung auf das Ziel eines lookat zu schauen.

[0167] Bei einer Ausführungsform der vorliegenden Erfindung werden der Party-Simulator, der Aufbereiter **54**, das VC **50** und die Echtzeitanwendung **46** so implementiert, daß sie gleichzeitig auf dem gleichen, Pentium-basierten Computersystem **10** laufen. Bei einer anderen Ausführungsform der vorliegenden Erfindung werden der Party-Simulator und die Komponenten des Systems **44** als individuelle, entfernte Module implementiert, welche durch ein Computernetzwerk (z.B. das Internet, ein Intranet) verbunden sind. Der Quell-Code für die Schnittstelle **450** wird als eine Kombination aus Visual C++ und Visual Basic von Microsoft Corporation in Redmond, Washington, implementiert. Es können jedoch auch andere Programmiersprachen verwendet werden, um die Schnittstelle **450** zu implementieren. Der Aufbereiter **54** verwendet Direct3D (Rendermorphics) von Microsoft Corporation in Redmond, Washington, um jedes Bild zu erzeugen. Es kann jedoch auch andere Aufbereitungs-Software verwendet werden. Das vollständige System läuft mit einer Geschwindigkeit von ungefähr 5 Zeiteinheiten pro Sekunde, von welchen die meiste Zeit in dem Aufbereiter **54** verbracht wird.

**[0168]** Das VC **50** hat sich als ziemlich robust beim Einrichten in Echtzeit des Party-Simulators ohne vorherige Informationen über die Ereignisse oder die Unterhaltungen bei der Party erwiesen. Es gibt eine gute Übereinstimmung zwischen der High-Level-Steuerung der Handlungen des Darstellers und der Idiom-Level-Steuerung, welche in dem VC **50** codiert ist.

**[0169]** Eine Verwendung des Systems **44** ist sehr intuitiv. Der Benutzer muß die Kamera mit der automatisierten Kamerasteuerung bei dieser beispielhaften Ausführungsform nicht steuern. Die Aufnahmen und Schnitte stellen eine sehr informative Erzählung der Ereignisse bereit. Zuschauer, welche den Teilnehmern über die Schulter blicken, können der Handlung auch leicht folgen, ohne die spezifischen Befehle zu kennen, welche durch den Benutzer erteilt werden.

**[0170]** Die folgenden Prinzipien der Kinematographie werden in dem VC **50** wie folgt angesprochen: die Kamerawinkel und -Bewegung, eingefangen in den Kameramodulen; Don't cross the line, Auswahl einer sachgerechten Instanz beim Kameramodul; Avoid jump cuts, Idiomstruktur erzwingt deutliche Änderungen (oder keine Änderungen) beim Kameramodul, wenn Idiomzustände geändert werden; Use establishing shot, Idiomhierarchie; Let the actors lead, Handlung ruft Wegübergang im Idiom auf; Break movement, Idiom-Wegübergänge werden durch globalen T-Wert bedingt; Maintain continuous movement, Instanzauswahl beim Kameramodul und bei Idiom-Wegübergängen; Aufnahmegestaltung, Kameramodulanordnung, Darstellungshinweise zum Neupositionieren von Darstellern und der Stil, die spezifischen Idiome und die Idiomhierarchie, welche von einem bestimmten Regisseur entworfen wurden.

**[0171]** Durch Codieren der Kompetenz, welche von wirklichen Filmemachern entwickelt wurde, in einem hierarchischen endlichen Automaten erzeugt das VC **50** beim Ablauf der Handlung automatisch eine Kamerasteuerung für individuelle Aufnahmen und Sequenzen dieser Aufnahmen. Die Ergebnisse haben sich als ziemlich robust gegenüber den a priori unbekannten Vorfällen in der Echtzeitumgebung des Party-Simulators erwiesen. Zusätzlich behandelt das VC **50** die Low-Level-Kameramodule, welche sich wirksam mit geometrischen Einzelheiten befassen; gestattet eine Domänen-abhängige Idiomhierarchie, um eine Regiekompetenz einzufangen, welche einfach, robust und skalierbar ist. Das VC **50** kann dem Aufbereiter **54** auch Darstellungshinweise präsentieren. Die Hinweise umfassen ein überlegtes Entfernen von Darstellern, um die Sicht der virtuellen Kamera frei zu machen; ein subtiles Neuordnen der Darsteller; und ein Modifizieren des Darstellungsverhaltens gemäß der Größe des Darstellers auf der Darstellungsfläche.

**[0172]** Das VC **50** wird bei Anwendungen der virtuellen Realität und anderen interaktiven Anwendungen verwendet, um sie ausgehend von den Aufnahmen mit festem Ansichtspunkt oder an der Decke befestigten virtuellen Kameras zu verbessern, welche derartige Anwendungen heute typischerweise einsetzen. Während es primär unter Bezugnahme auf eine virtuelle 3D-Chat-Umgebung beschrieben wurde, hilft das VC **50** auch, die nächste Generation „intelligenter Agenten“-Benutzerschnittstellen zu verbessern, indem den Benutzern gestattet wird, sich selbst mit einem Agenten in Kamerapositionen zu sehen, welche natürlich erscheinen.

**[0173]** Bei einer virtuellen 3D-Chat-Umgebung interagieren Computerbenutzer miteinander durch animierte virtuelle 3D-Darsteller (manchmal virtuelle Figuren genannt), welche von den Computerbenutzern gesteuert werden und sie repräsentieren. Die virtuellen Darsteller werden in einer Anzeige eines 3D-Raums aufbereitet, in welchem die virtuellen Darsteller gemäß den Benutzern angeordnet werden, welche miteinander kommunizieren. In einer ähnlichen Weise kann das VC **50** eine „intelligenter Agent“-Benutzerschnittstelle bereitstellen, bei welcher der intelligente Agent und jede zugrundeliegende Computeranwendung oder -Anwendungen in Bezug zueinander in einem 3D-Anzeigeraum angeordnet und positioniert werden.

**[0174]** Ein intelligenter Agent ist typischerweise eine hoch interaktive Benutzerschnittstellenvorrichtung (z.B. ein Charakter), welcher in verschiedenen Zusammenhängen, wie beispielsweise beim Bereitstellen von Hilfe oder Assistenz für einen Benutzer bei einer Software-Anwendung, verwendet wird. Bei einer Ausführungsform kann das VC **50** den intelligenten Agenten so positionieren oder aufbereiten, daß er nicht die Anwendungsdarstellungsfläche verdeckt, für welche der intelligente Agent assistiert. Bei einer anderen Ausführungsform wird dem menschlichen Benutzer des Systems eine virtuelle Position in dem 3D-Anzeigeraum zugeordnet, dessen Position auch durch das VC **50** eingesetzt wird, um Kameraansichten und die resultierenden Anzeigen zu bilden.

**[0175]** Es versteht sich, daß die hier beschriebenen Programme, Prozesse und Verfahren keinen bestimmten Typ einer Computervorrichtung betreffen oder darauf eingeschränkt sind, außer es ist anderslautend angegeben. Es können gemäß den hier beschriebenen Lehren verschiedene Typen von universellen oder spezialisierten Computervorrichtungen verwendet werden oder Operationen durchgeführt werden.

**[0176]** Mit Blick auf die breite Vielfalt von Ausführungsformen, auf welche die Prinzipien der Erfindung angewendet werden können, versteht es sich, daß die illustrierten Ausführungsformen nur beispielhaft sind und nicht als den Schutzzumfang der Erfindung einschränkend aufgefaßt werden dürfen. Statt dessen werden als die Erfindung alle derartigen Ausführungsformen beansprucht, welche innerhalb des Schutzzumfangs der folgenden Ansprüche liegen.

### Patentansprüche

1. Verfahren zur virtuellen Kinematographie, wobei das Verfahren umfaßt:  
Empfangen von kinematographischen Ereignissen in Echtzeit unter Verwendung eines Echtzeitanwendungsmoduls (**46**);  
Interpretieren der empfangenen kinematographischen Ereignisse in Echtzeit unter Verwendung einer kinematographischen Logik, wobei die kinematographische Logik einen oder mehrere endliche Automaten (**272**) zum Interpretieren der empfangenen kinematographischen Ereignisse umfaßt; und  
automatisches Erzeugen einer Spezifikation einer Kameraansicht aus den interpretierten kinematographischen Ereignissen, wobei die Spezifikation der Kameraansicht verwendet wird, um eine virtuelle Szene zu betrachten.
2. Verfahren nach Anspruch 1, wobei die Spezifikation der Kameraansicht wenigstens eine Spezifikation umfaßt, die aus einer Gruppe ausgewählt ist, welche apex, closeapex, apex3, external, 2long, external1to2, exclose2, internal, full, tracking, panning, follow, fixed, raised, group, subjective, subjectzoom, pov, und null Spezifikationen der Kameraansicht umfaßt.
3. Verfahren nach Anspruch 1 oder 2, welches des weiteren ein Auswählen von Entfernungen der virtuellen Kameraansichten umfaßt.
4. Verfahren nach Anspruch 3, wobei die Entfernungen der virtuellen Kameraansichten wenigstens eine Entfernung ausgewählt aus einer Gruppe umfassen, die eine Entfernung der extremen Nahaufnahmenansicht, eine Entfernung der Nahaufnahmenansicht, eine Entfernung der mittleren Ansicht, eine Entfernung der Vollansicht und eine Entfernung der Distanzansicht umfaßt.
5. Verfahren nach einem der vorhergehenden Ansprüche, wobei die kinematographische Logik Filmheuristiken zum Auswählen von Filmaufnahmen umfaßt.
6. Verfahren nach Anspruch 5, wobei die Filmheuristiken wenigstens eine Heuristik umfassen, welche aus einer Gruppe gewählt ist, die "Don't cross the Line", "Avoid jump cuts", "Use establishing shots", "Let the action lead", "Break movement" und "Maintain continuous movement" umfaßt.
7. Verfahren nach einem der vorhergehenden Ansprüche, wobei die kinematographische Logik Kameramodule und virtuelle kinematographische Idiome umfaßt.
8. Verfahren nach Anspruch 7, welches des weiteren umfaßt:  
Erzeugen eines oder mehrerer individueller virtueller kinematographischer Idiome, wobei das Erzeugen umfaßt:  
Definieren eines oder mehrerer individueller virtueller kinematographischer Idiomzustände;  
Auswählen eines Kameramoduls, welches zu jedem definierten individuellen virtuellen kinematographischen Idiomzustand zugehörig ist; und  
Definieren von Idiomhandlungen, welche Übergänge zwischen den definierten individuellen virtuellen kinematographischen Idiomzuständen umfaßt;  
wobei die Idiomhandlungen verwandt werden, um eine virtuelle Szene in Echtzeit darzustellen.
9. Verfahren nach Anspruch 8 oder 9, wobei die Idiomhandlungen die Registrierung und das Heben von Ausnahmen umfaßt.
10. Verfahren nach Anspruch 8, welches des weiteren umfaßt:  
Definieren von einem oder mehreren komplexen virtuellen kinematographischen Idiomzuständen;  
Auswählen eines individuellen virtuellen kinematographischen Idioms für einen oder mehrere der definierten komplexen virtuellen kinematographischen Idiomzustände;  
Definieren von komplexen Idiomhandlungen, welche Übergänge zwischen den definierten komplexen virtuellen kinematographischen Idiomzuständen umfassen; und

Definieren von Bedingungen, welche geprüft werden, um zu entscheiden, welcher Übergang auftritt; wobei das komplexe virtuelle kinematographische Idiom verwandt wird, um eine virtuelle Szene in Echtzeit auf einem Computerbildschirm darzustellen.

11. Verfahren nach Anspruch 8, welches des weiteren umfaßt:

Auswählen einer ersten Gruppe von individuellen virtuellen kinematographischen Idioms;  
Kombinieren der ersten Gruppe von individuellen virtuellen kinematographischen Idioms, um ein komplexes virtuelles kinematographisches Idiom zu erzeugen, wobei das komplexe virtuelle kinematographische Idiom komplexe virtuelle kinematographische Idiomzustände, Kameramodule und komplexe virtuelle kinematographische Idiomhandlungen umfaßt;  
Auswählen einer zweiten Gruppe von individuellen virtuellen kinematographischen Idioms und des komplexen virtuellen kinematographischen Idioms; und  
Kombinieren der zweiten Gruppe von individuellen virtuellen kinematographischen Idioms und des komplexen virtuellen kinematographischen Idioms, um ein virtuelles kinematographisches High-Level Idiom zu erzeugen;  
wobei das virtuelle kinematographische High-Level Idiom verwandt wird, um eine virtuelle Szene in Echtzeit auf einem Computerbildschirm darzustellen.

12. Verfahren nach Anspruch 11, wobei das virtuelle kinematographische High-Level Idiom ein virtuelles kinematographisches Verbidium ist.

13. Verfahren nach Anspruch 7, welches des weiteren umfaßt:

Auswählen einer Gruppe von virtuellen kinematographischen Verbidiums;  
Kombinieren der ausgewählten Gruppe von virtuellen kinematographischen Verbidiums, um eine Vielzahl von virtuellen Filmszenen zu erzeugen; und  
Kombinieren von aus diesen virtuellen Filmszenen ausgewählten Szenen, um einen virtuellen Film zu erzeugen.

14. Verfahren nach Anspruch 7, welches des weiteren umfaßt:

Auswählen einer Gruppe von virtuellen kinematographischen Idioms, wobei die virtuellen kinematographischen Idioms einen oder mehrere virtuelle kinematographische Idiomzustände, ein oder mehrere Kameramodule und individuelle Idiomhandlungen umfaßt;  
Ändern der Kameramodule innerhalb der virtuellen kinematographischen Idioms, basierend auf einem spezifischen Filmregiestil, um eine zugeschnittene, modifizierte Gruppe von virtuellen kinematographischen Idioms zu erzeugen;  
Kombinieren der modifizierten Gruppe von virtuellen kinematographischen Idioms, basierend auf den spezifischen Filmregiestilen, um eine Vielzahl von virtuellen Filmszenen zu erzeugen; und  
Kombinieren von aus den virtuellen Filmszenen ausgewählten Szenen, um einen virtuellen Film zu erzeugen.

15. Verfahren nach einem der vorhergehenden Ansprüche, welches des weiteren Modifizieren der Anordnung und Handlungen der virtuellen Darsteller umfaßt, um der gewählten Spezifikation und Entfernung der virtuellen Kameraansicht zu entsprechen.

16. Verfahren nach einem der vorhergehenden Ansprüche, welches des weiteren ein Erzeugen eines virtuellen kinematographischen Kameramoduls (**50**) umfaßt, wobei das Erzeugen eines virtuellen kinematographischen Kameramoduls (**50**) umfaßt:

Auswählen eines virtuellen Objekts zur Ansicht;  
Auswählen einer Spezifikation der virtuellen Kameraansicht zur Ansicht des Objekts; und Verbinden der ausgewählten Spezifikation der virtuellen Kamera mit dem virtuellen Objekt.

17. Computerprogramm, welches Computerprogrammcodemittel umfaßt, welche so ausgelegt sind, daß alle Schritte eines der vorhergehenden Ansprüche ausgeführt werden, wenn das Programm auf einem Rechner ausgeführt wird.

18. Computerprogramm nach Anspruch 17, welches auf einem computerlesbaren Medium abgelegt ist.

19. System für virtuelle Kinematographie, wobei das System umfaßt:

ein Echtzeitanwendungsmodul (**46**) zum Erstellen kinematographischer Ereignisse als Antwort auf eine Benutzereingabe, welche in Echtzeit auf einem Clientrechner empfangen wird;  
ein virtuelles kinematographisches Modul (**50**) zum Interpretieren der kinematographischer Ereignisse in Echt-

zeit und zum automatischen Bestimmen der Kameraposition basierend auf den kinematographischen Ereignissen, wobei das virtuelle kinematographische Modul (**50**) eine oder mehrere endliche Automaten (**272**) zum Interpretieren der kinematographischen Ereignisse umfaßt; und ein Bildaufbearbeitungsanwendungsmodul (**54**) zum Aufbereiten einer virtuellen Umgebung als Antwort auf Befehle von dem virtuellen kinematographischen Modul (**50**).

20. System nach Anspruch 19, wobei das virtuelle kinematographische Modul (**50**) ein Kameramodul und virtuelle kinematographische Idiome umfaßt.

21. System nach Anspruch 19 oder 20, wobei die kinematographischen Ereignisse interaktive Animationsereignisse umfassen; und des weiteren umfaßt:  
einen Echtzeit Kameracontroller, zum Übersetzen der interaktiven Animationsereignisse in Positionen der Kameraansicht; und  
einen Aufbereiter (**54**) zum Aufbereiten einer virtuellen Umgebung unter Verwendung der Positionen der Kameraansicht.

22. System nach Anspruch 21, welches des weiteren einen Echtzeitkameracontroller für interaktive Anwendungen und Aufnahmen mit festem Ansichtspunkt umfaßt.

23. System nach einem der Ansprüche 19 bis 22, welches des weiteren eine Vielzahl von Clientrechner, die mit einem Netzwerk verbunden sind, umfaßt, wobei der erste erwähnte Clientrechner mit dem Netzwerk verbunden ist und wenigstens zwei Clientrechner jeweils einen virtuellen Darsteller in der virtuellen Umgebung aufweisen, wobei jeder virtuelle Darsteller separat von dem ihm zugeordneten Clientrechner gesteuert wird.

Es folgen 17 Blatt Zeichnungen

FIG. 1

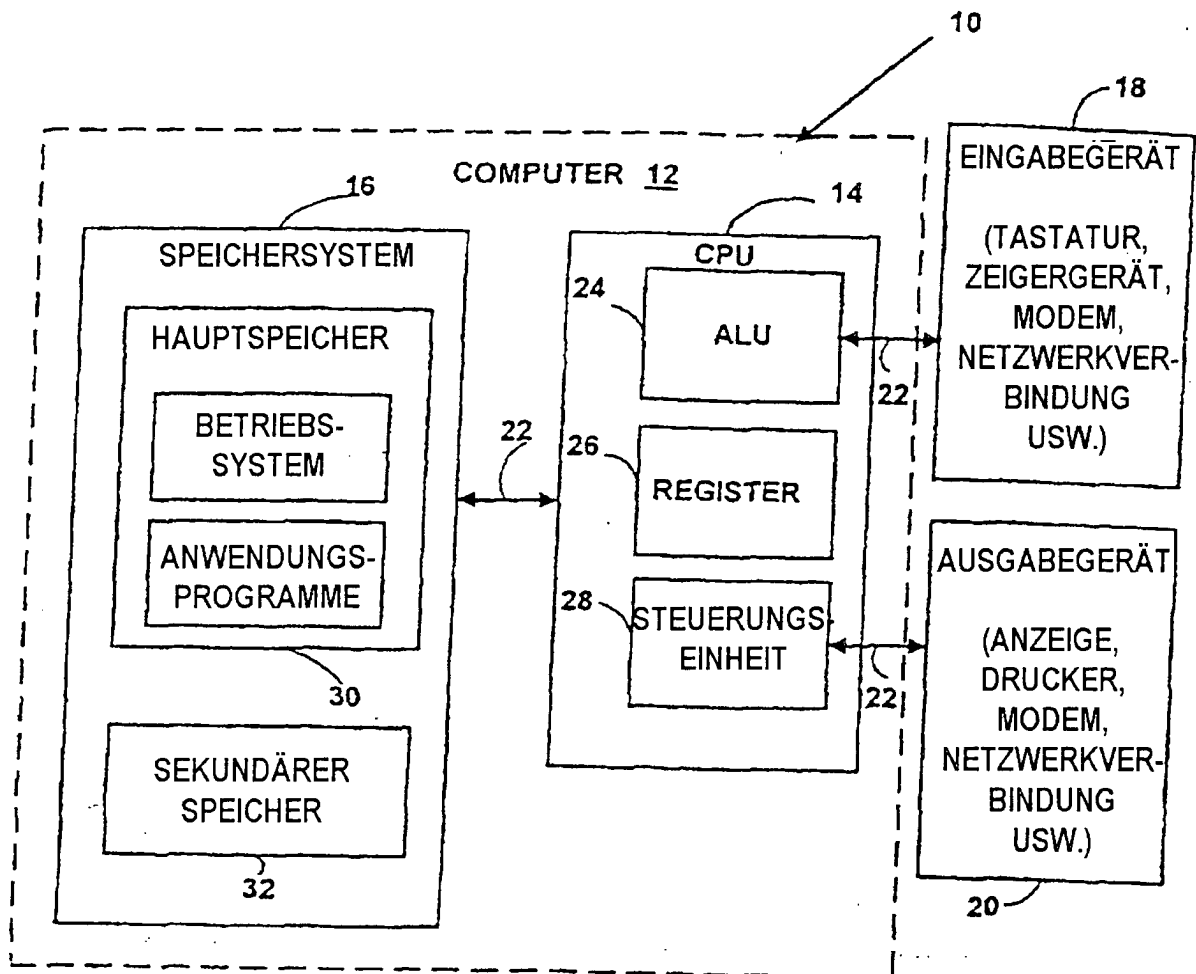




FIG. 2

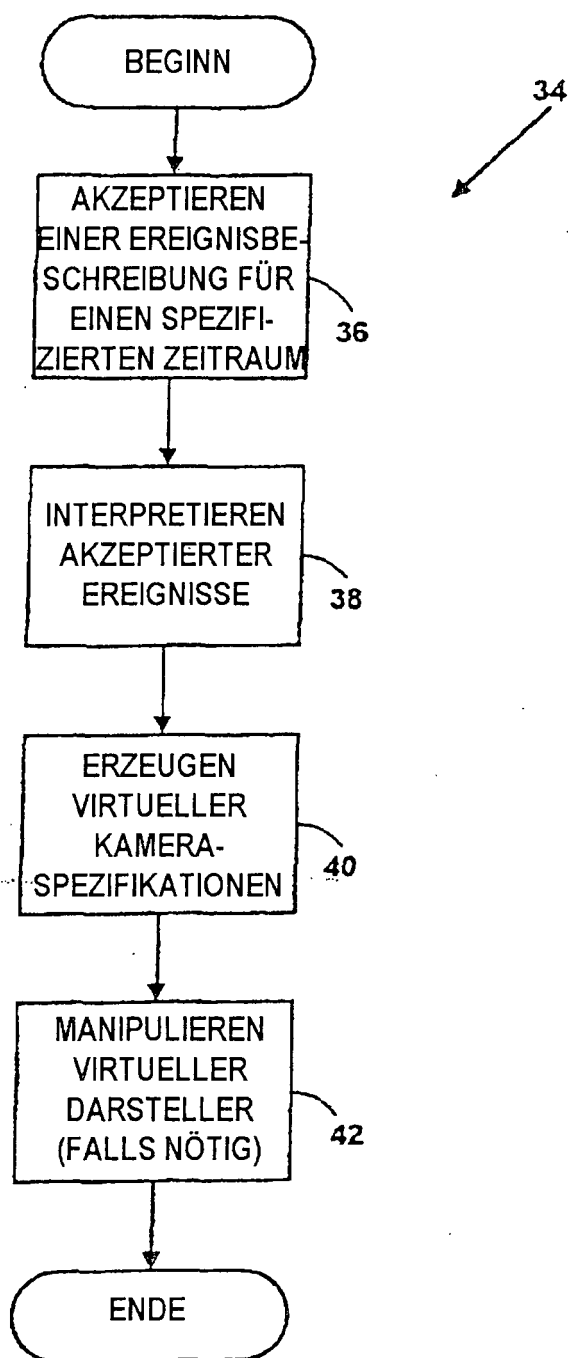


FIG. 3

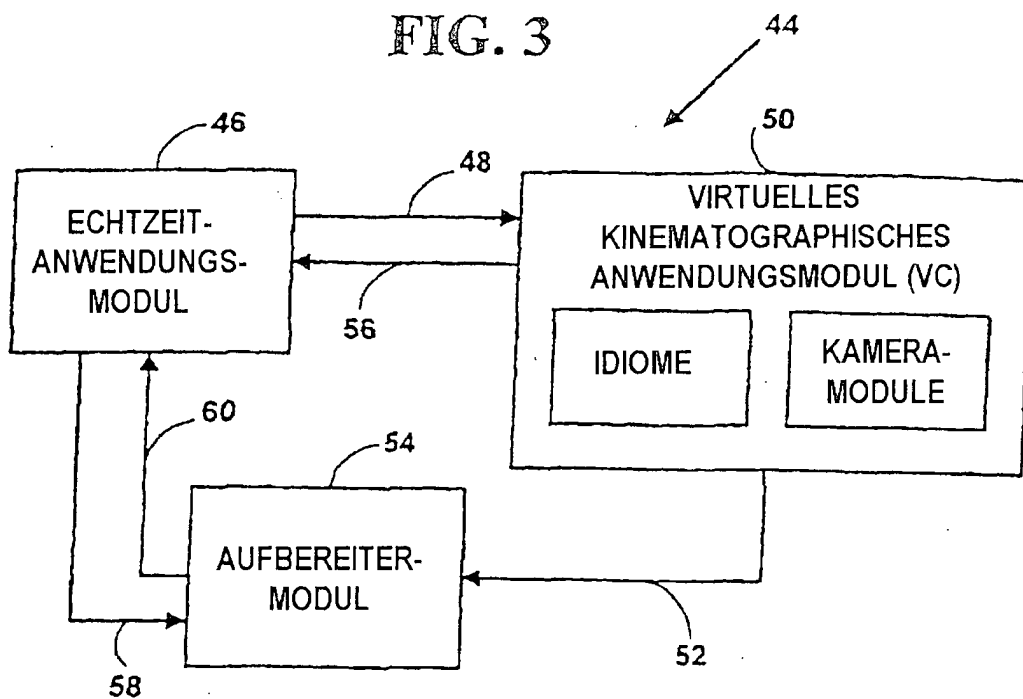
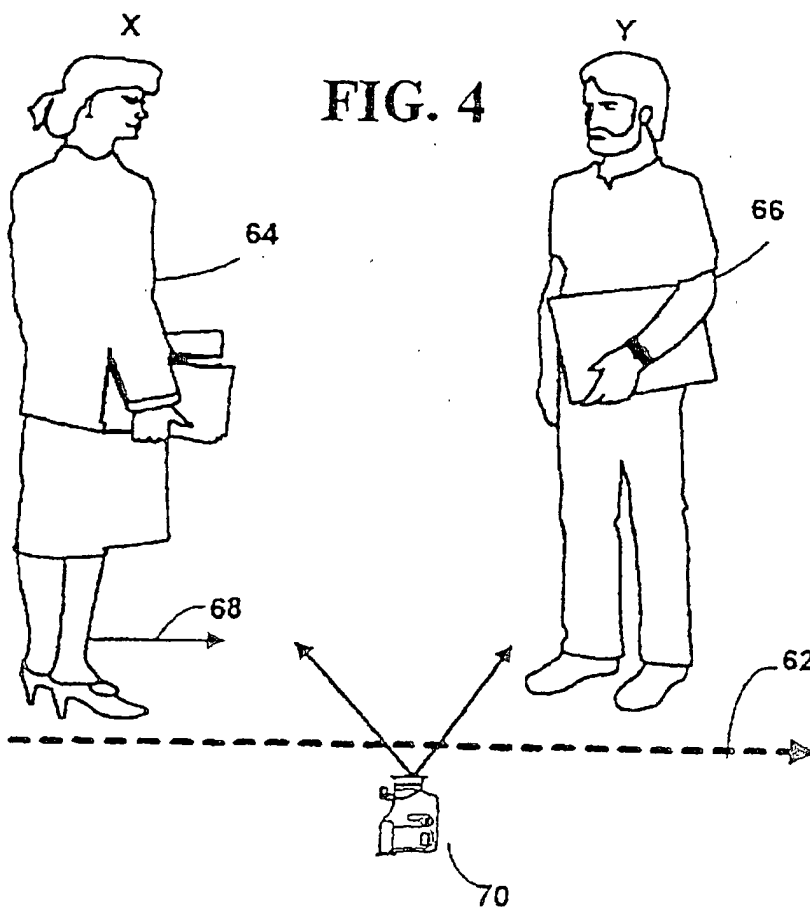


FIG. 4



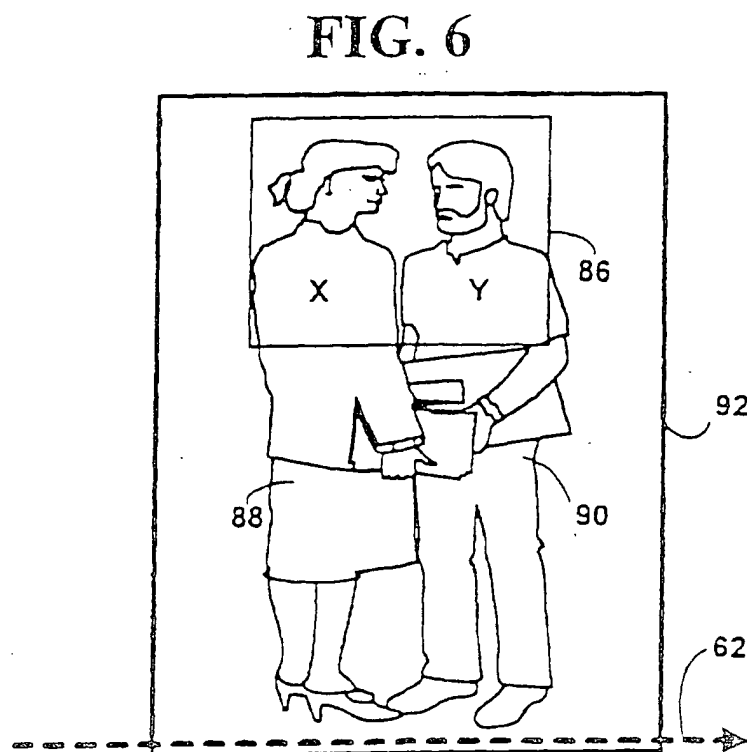
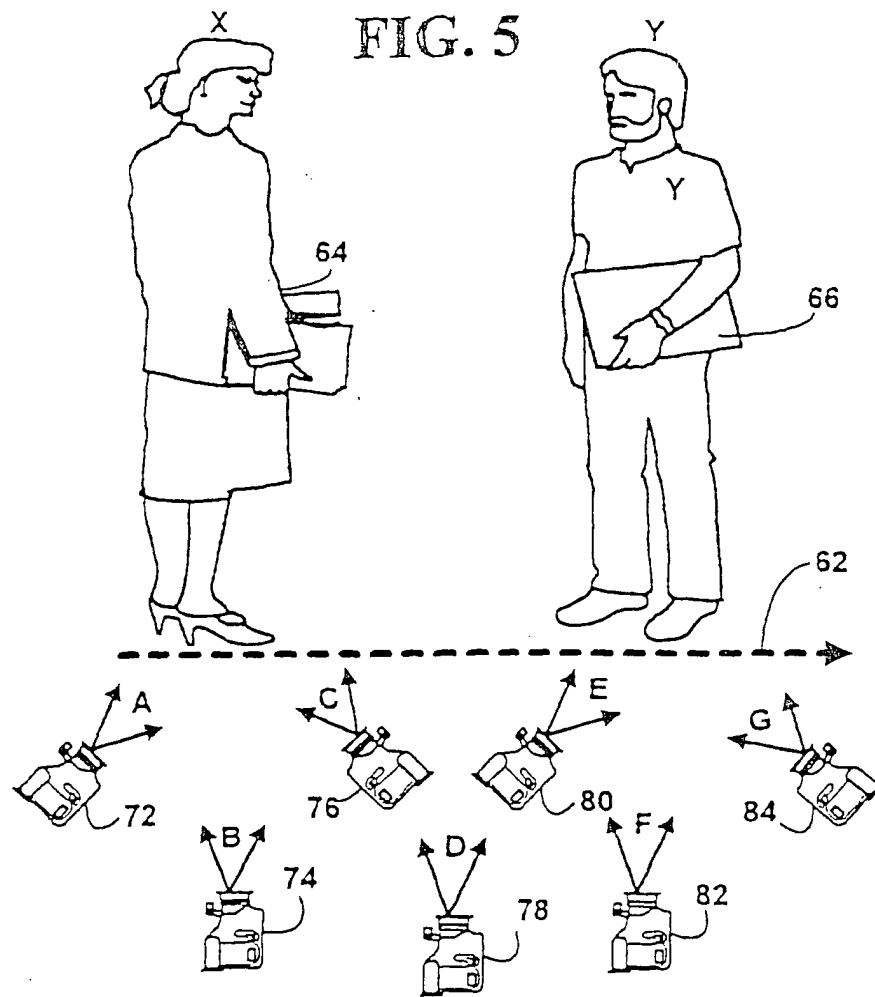


FIG. 7A

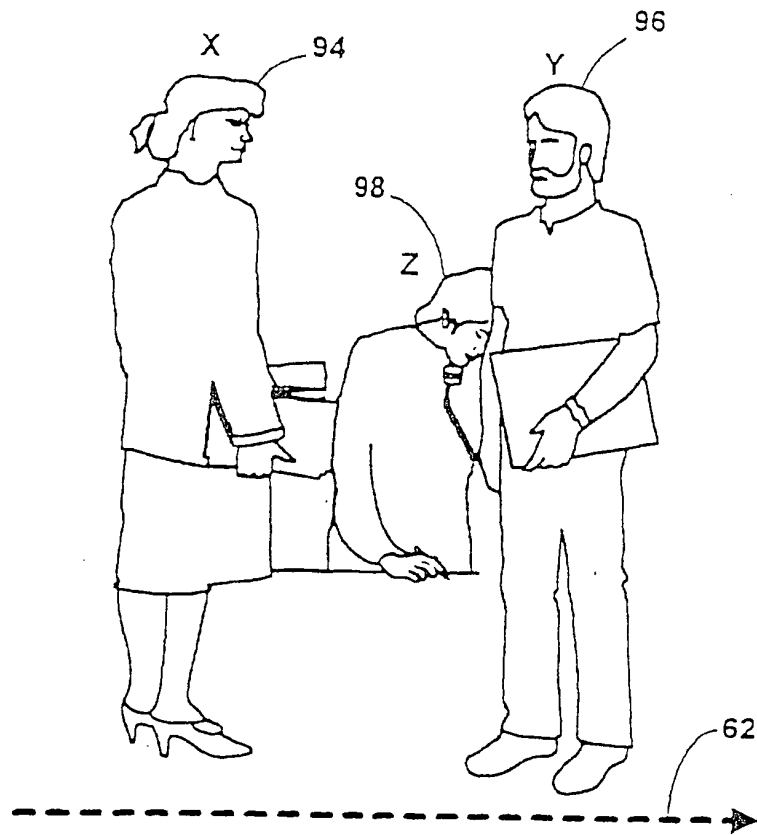


FIG. 7B

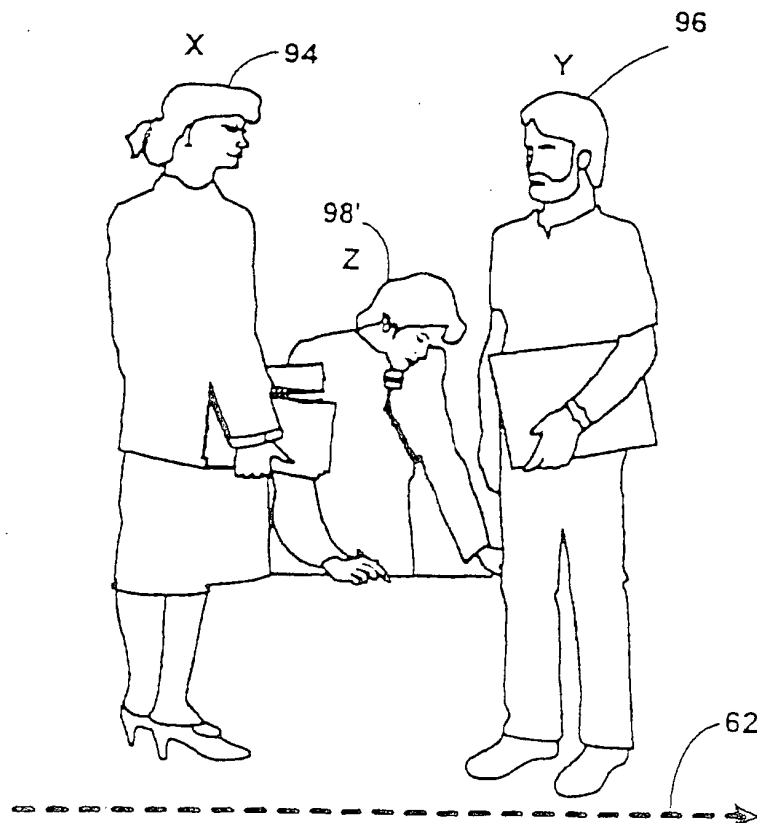


FIG. 8

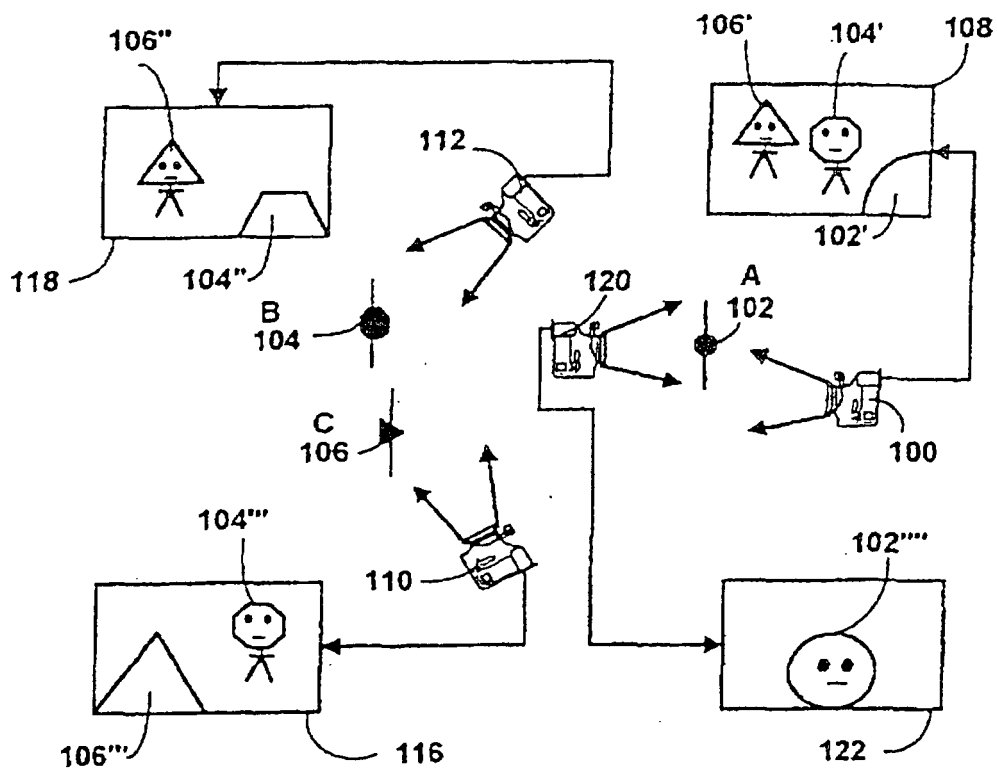


FIG. 9

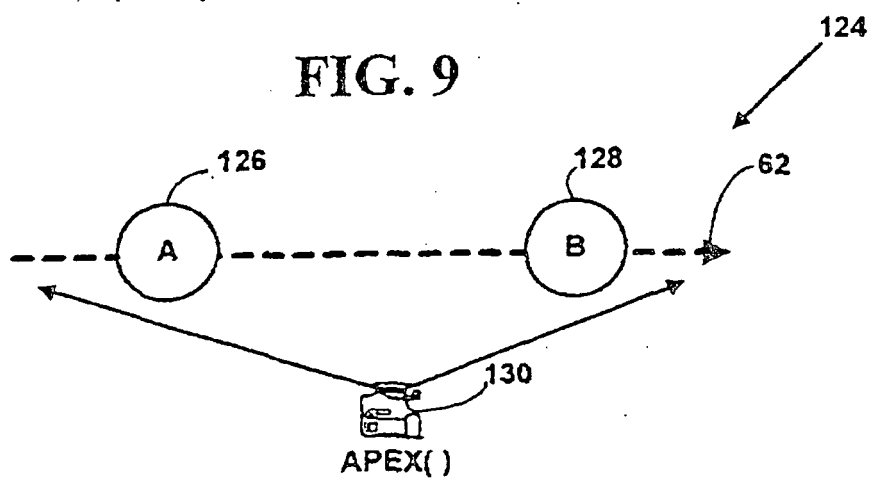


FIG. 10

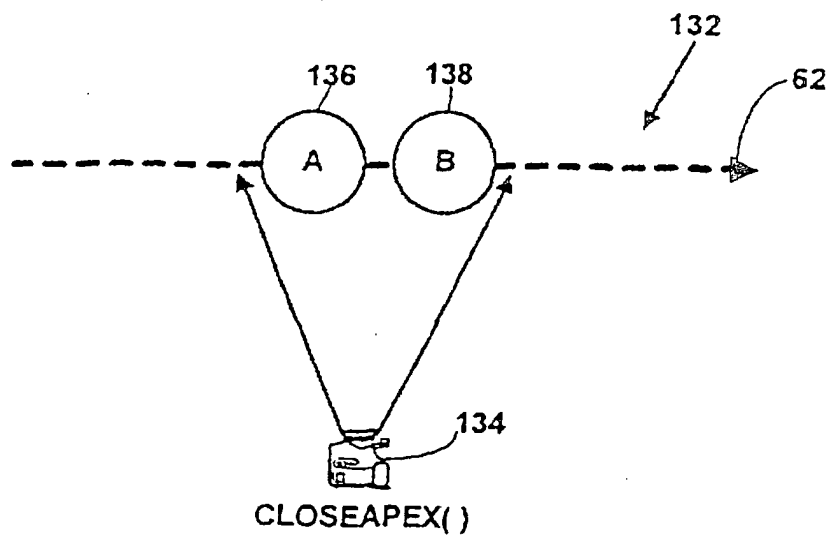


FIG. 11

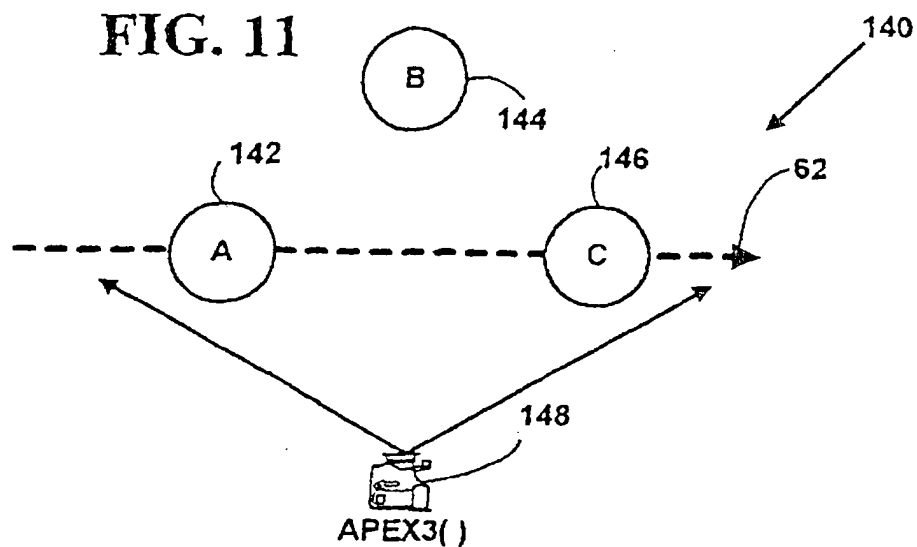


FIG. 12

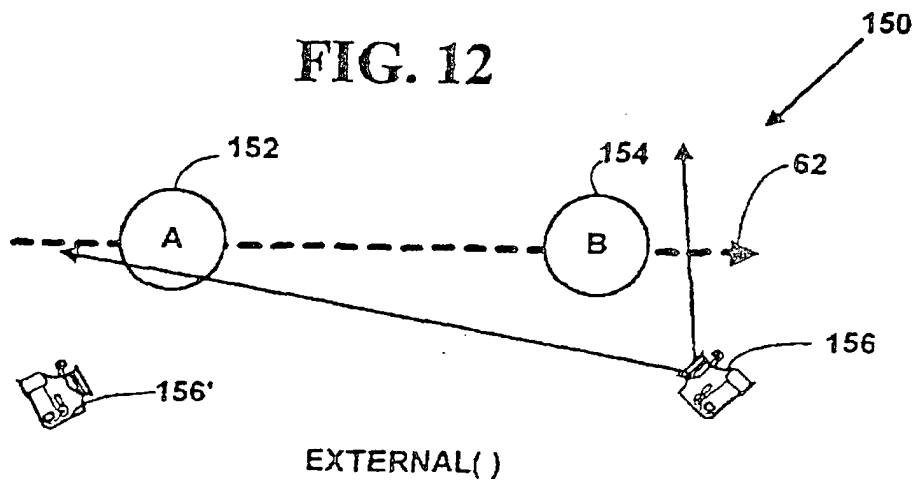




FIG. 13

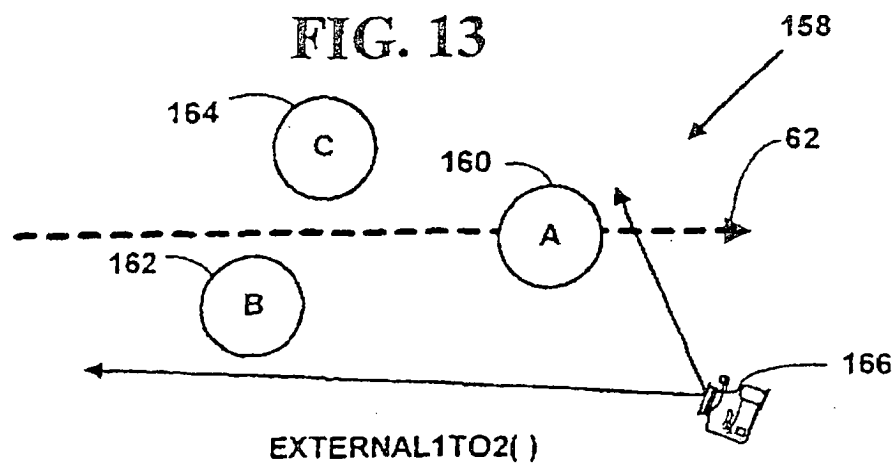


FIG. 14

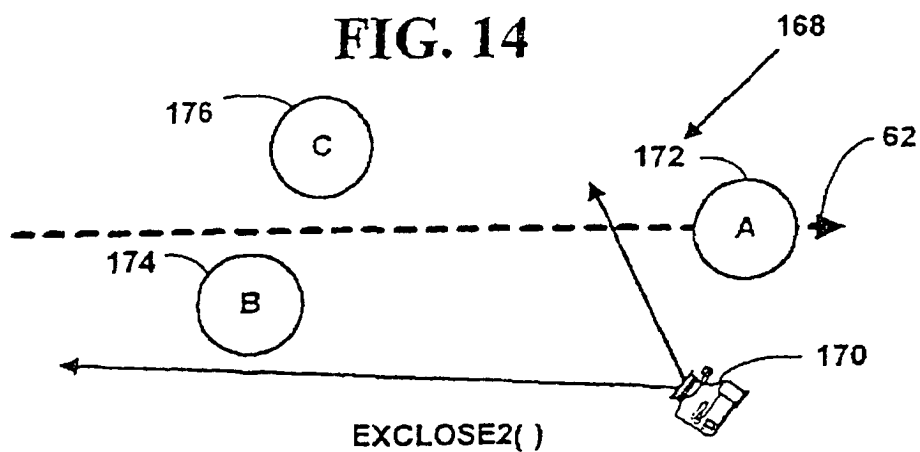


FIG. 15

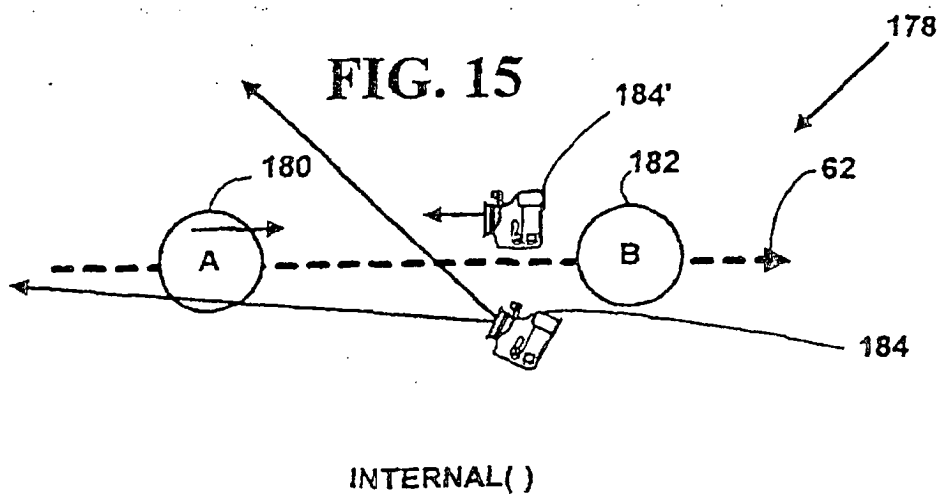


FIG. 16

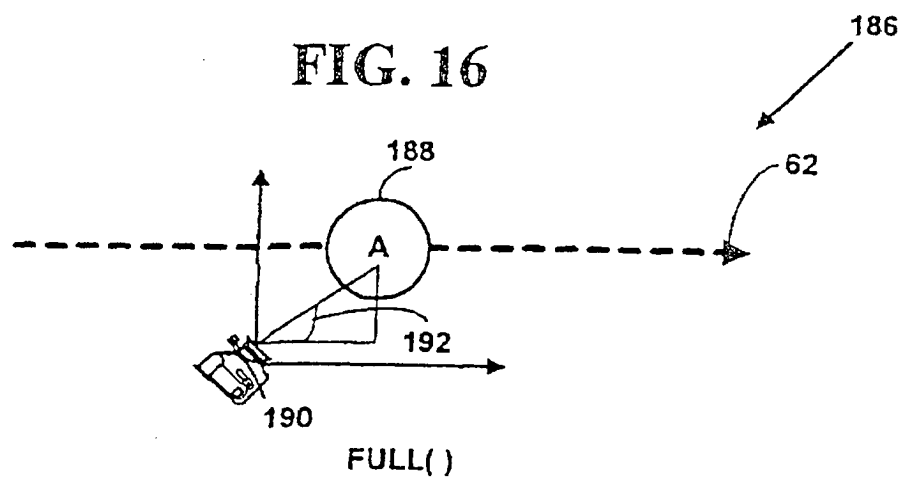


FIG. 17

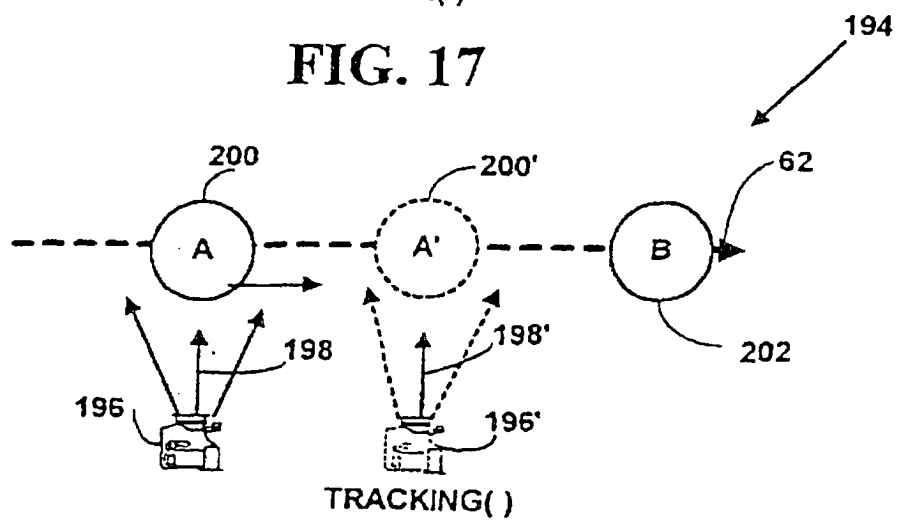


FIG. 18

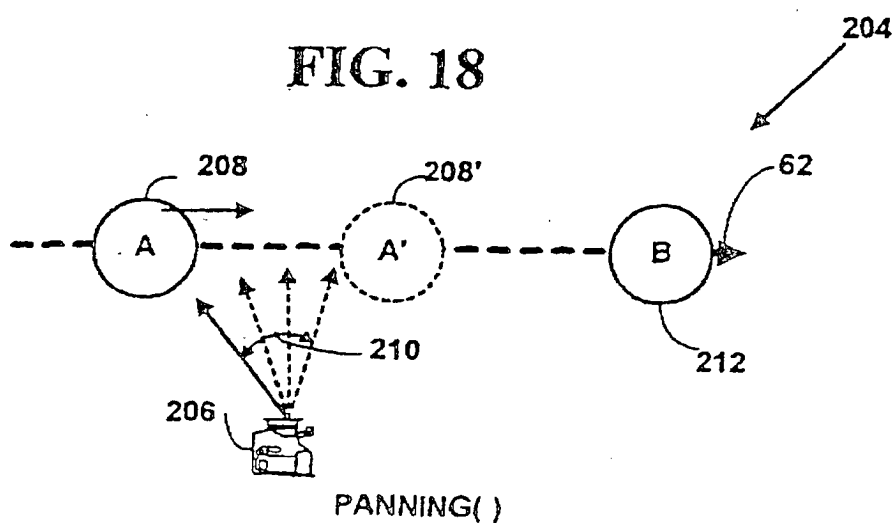


FIG. 19

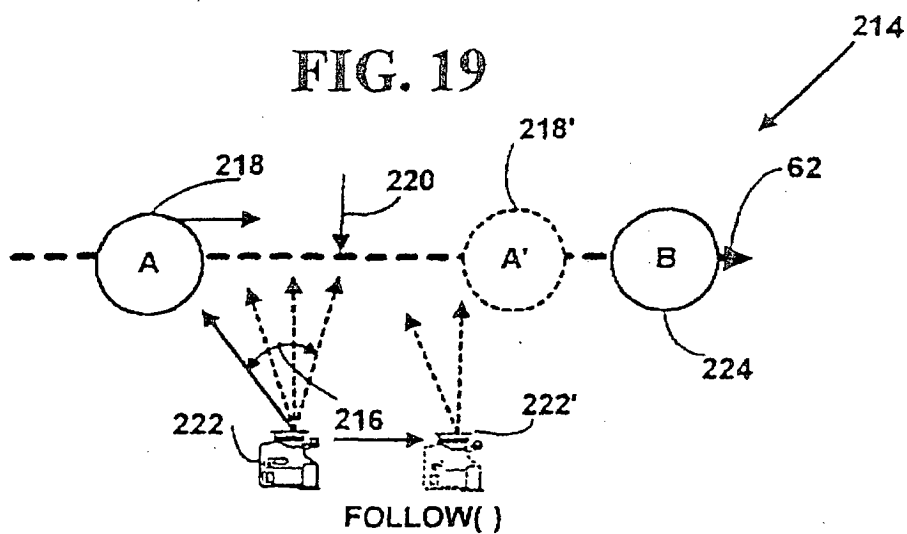


FIG. 20

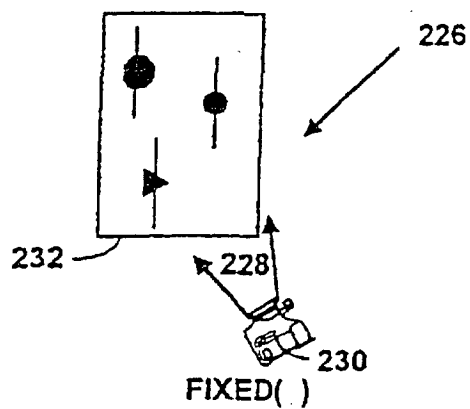
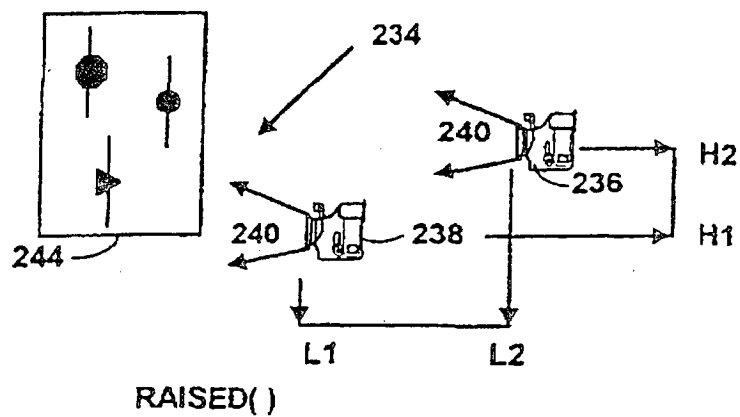


FIG. 21



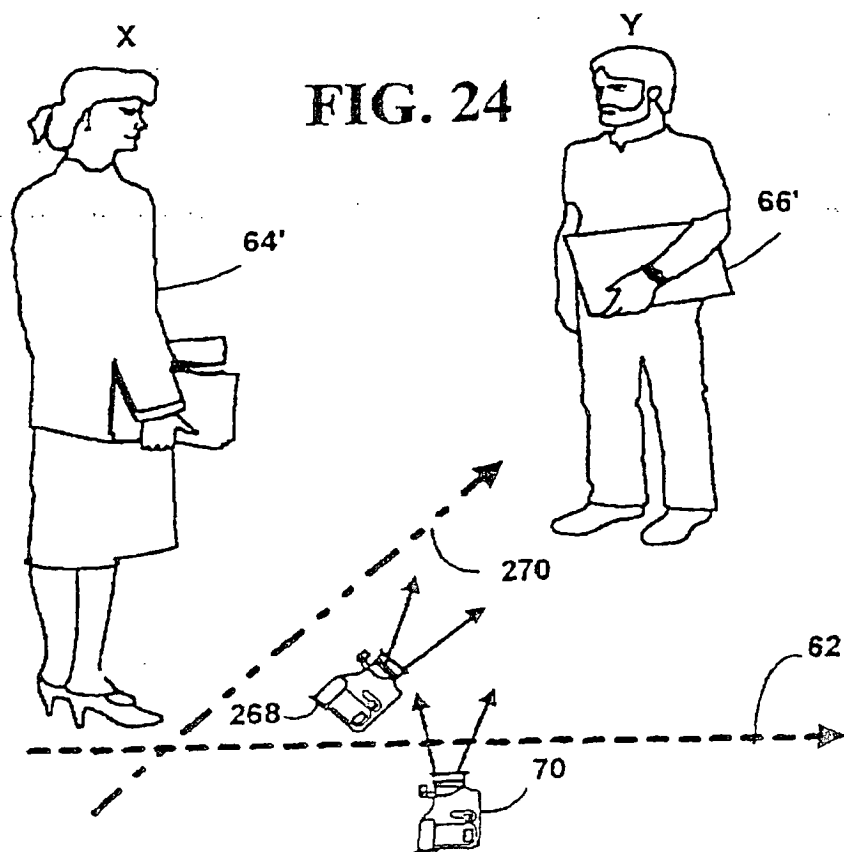
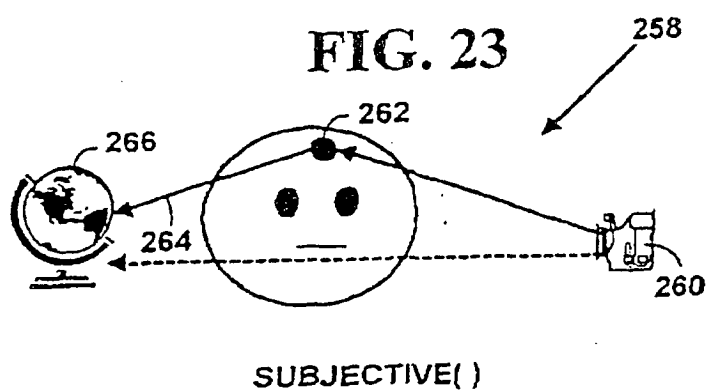
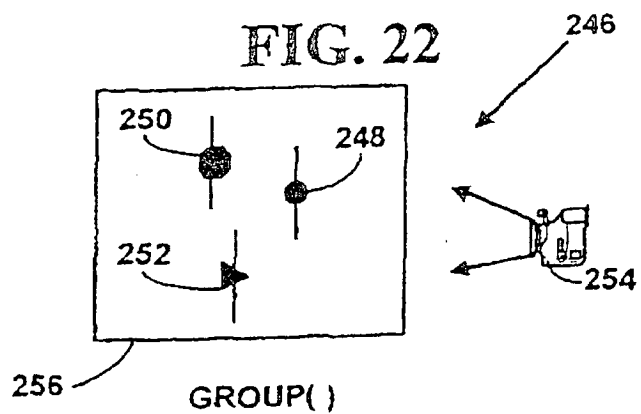


FIG. 25

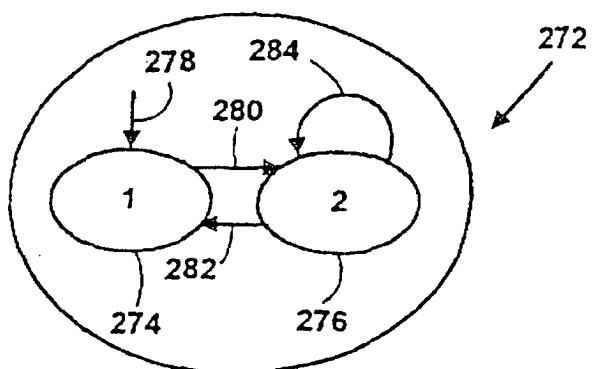
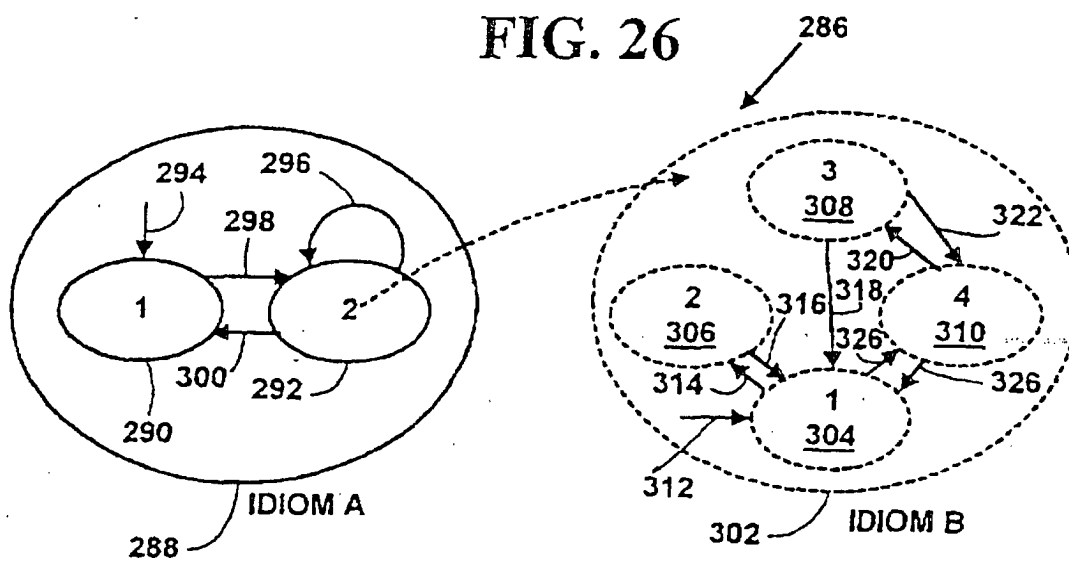
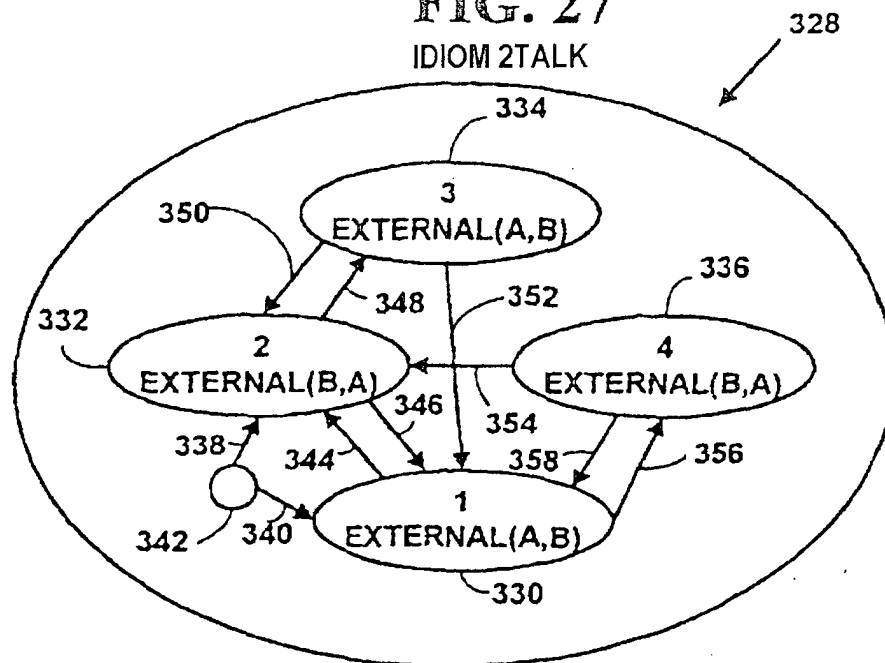


FIG. 26



**FIG. 27**

IDIOM 2TALK



**FIG. 28**

IDIOM 3TALK

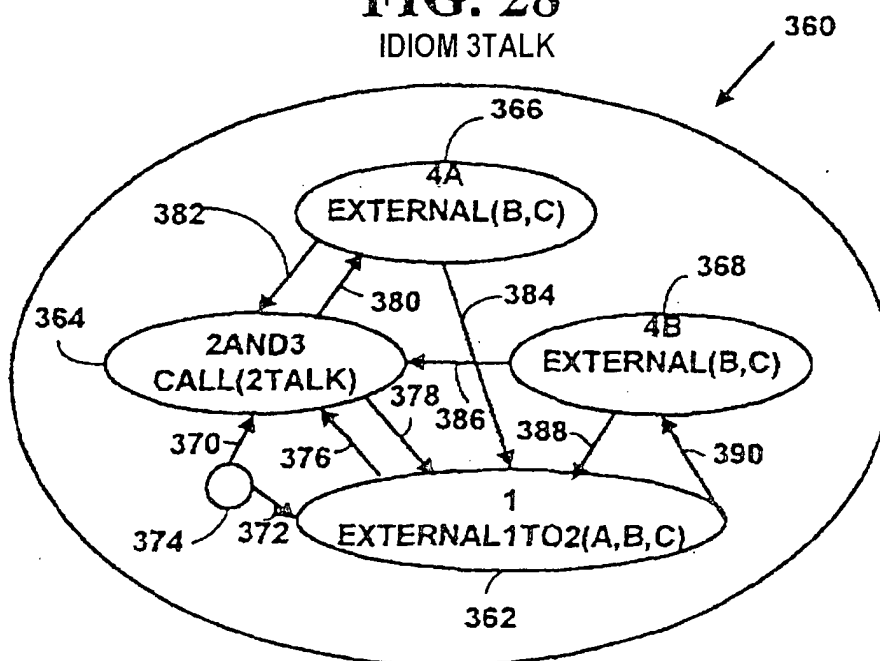


FIG. 29

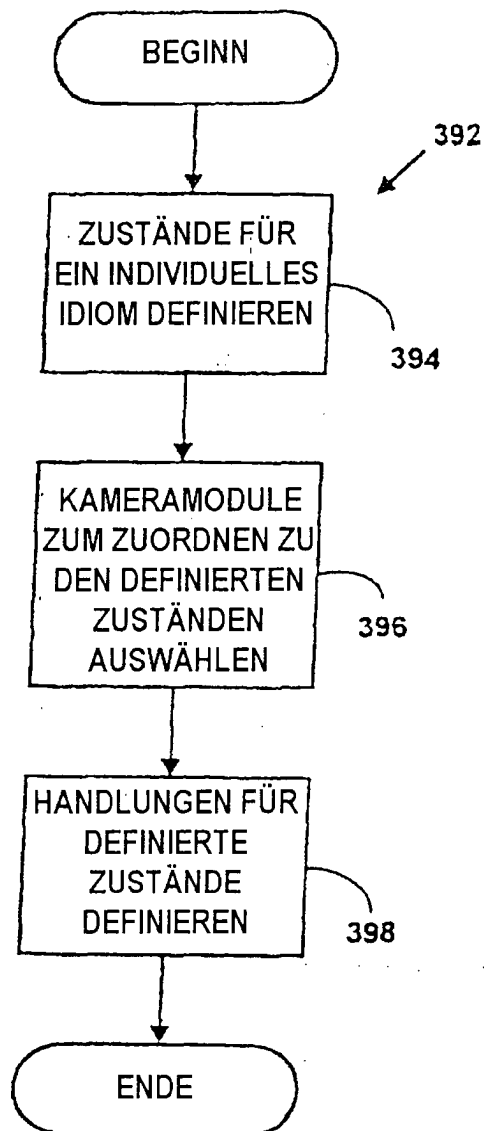


FIG. 30

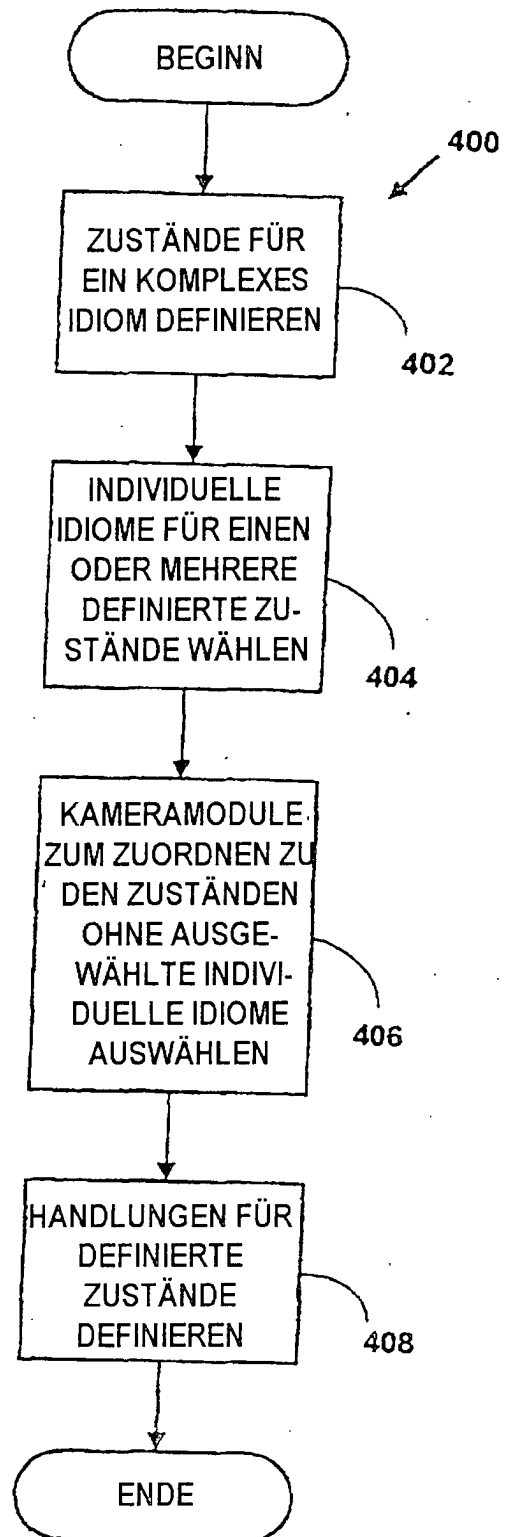


FIG. 31

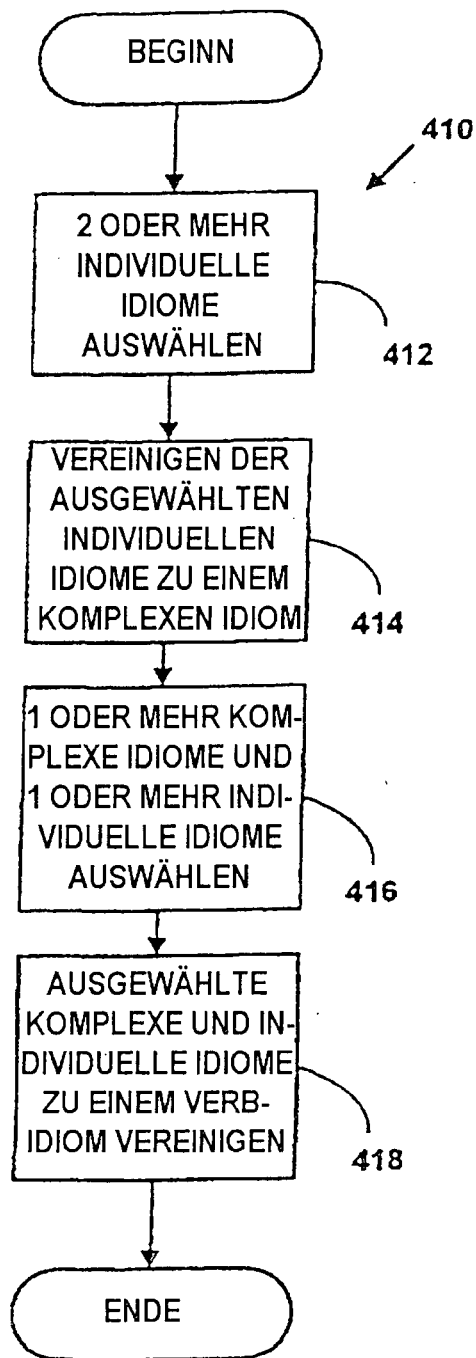


FIG. 32

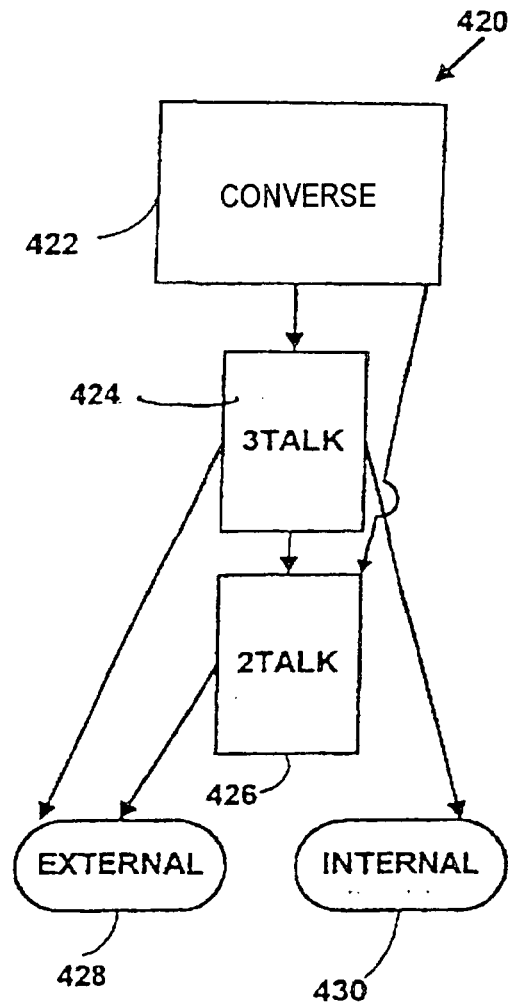




FIG. 33

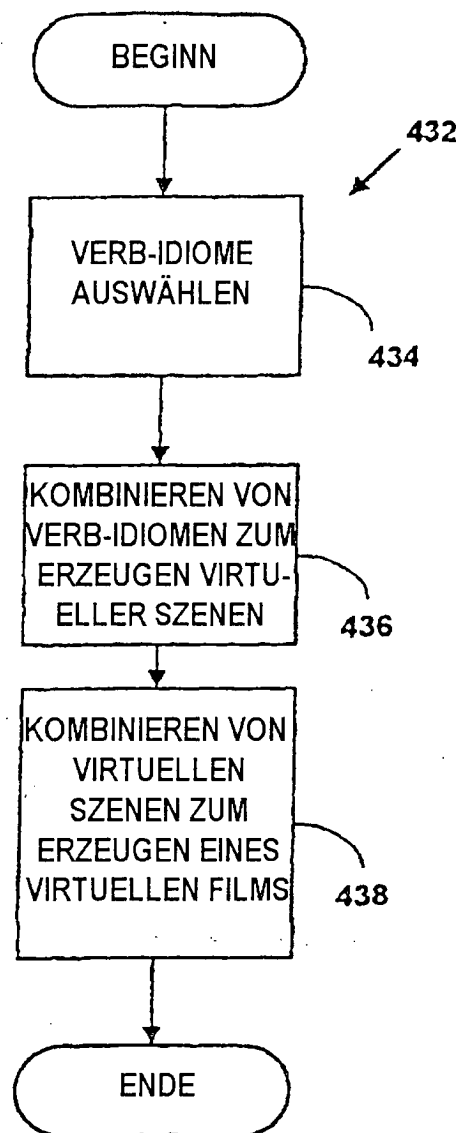


FIG. 34

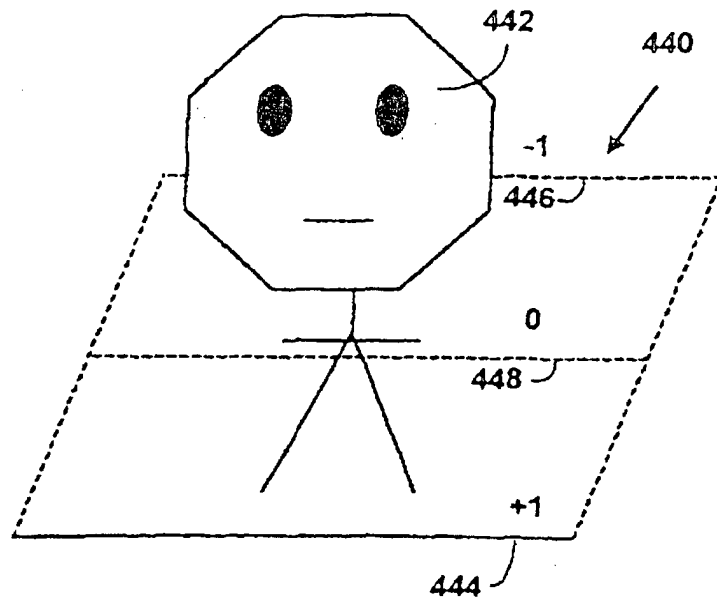


FIG. 35

