

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4607259号
(P4607259)

(45) 発行日 平成23年1月5日(2011.1.5)

(24) 登録日 平成22年10月15日(2010.10.15)

(51) Int.CI.

G06F 1/32 (2006.01)

F 1

G06F 1/00 332Z

請求項の数 31 (全 33 頁)

(21) 出願番号	特願平9-542729
(86) (22) 出願日	平成9年5月21日(1997.5.21)
(65) 公表番号	特表2001-503539 (P2001-503539A)
(43) 公表日	平成13年3月13日(2001.3.13)
(86) 國際出願番号	PCT/US1997/008702
(87) 國際公開番号	W01997/044737
(87) 國際公開日	平成9年11月27日(1997.11.27)
審査請求日	平成16年5月19日(2004.5.19)
審判番号	不服2008-2345 (P2008-2345/J1)
審判請求日	平成20年2月4日(2008.2.4)
(31) 優先権主張番号	60/018,405
(32) 優先日	平成8年5月22日(1996.5.22)
(33) 優先権主張国	米国(US)

(73) 特許権者	595020643 クアアルコム・インコーポレイテッド QUALCOMM INCORPORATED アメリカ合衆国、カリフォルニア州 92121-1714、サン・ディエゴ、モアハウス・ドライブ 5775
(74) 代理人	100108855 弁理士 蔵田 昌俊
(74) 代理人	100091351 弁理士 河野 哲
(74) 代理人	100088683 弁理士 中村 誠
(74) 代理人	100109830 弁理士 福原 淑弘

最終頁に続く

(54) 【発明の名称】感知された状況に応じて電子デバイスを制御する方法及び装置

(57) 【特許請求の範囲】

【請求項 1】

電子デバイスへの電力供給を制御するためのシステムにおいて、前記システムが、前記電子デバイスの地理空間位置を感知し、空間位置信号を生成する前記電子デバイスと接続した位置感知サブシステムと、

第一のパワーモードから第二のパワーモードへ前記電子デバイスの第一の部分を切り替えるための第一の制御信号を前記電子デバイスに与え、第一のパワーモードから第二のパワーモードへ前記電子デバイスの第二の部分を切り替える第二の制御信号を前記電子デバイスに与える制御サブシステムであり、前記第一のパワーモードが、パワーオフモードとパワーインモードの内の一方であり、前記第二のパワーモードが、パワーオフモードとパワーインモードの内の他方である、前記制御サブシステムと、

前記制御サブシステムと前記位置感知サブシステムとの間の少なくとも一つの通信接続手段と

を有し、

前記制御サブシステムが、前記位置感知サブシステムから少なくとも一つの空間位置信号を前記通信接続手段を介して受信するのに応じて、前記電子デバイスに前記第一及び第二の制御信号の少なくとも一方を与え、前記少なくとも一つの空間位置信号が前記電子デバイスの地理空間位置特性を表すシステム。

【請求項 2】

前記電子デバイスに結合されて、この電子デバイスの姿勢を感知して、姿勢信号を生成す

10

20

る姿勢感知デバイス、および

前記制御サブシステムと前記姿勢感知デバイスとの間の通信接続手段を更に含み、
前記制御サブシステムによって前記電子デバイスへ提供される制御信号の一つは前記姿勢
感知デバイスから前記制御サブシステムによって受信された信号にも応答する請求項1に
記載のシステム。

【請求項3】

前記第一のモードから前記第二のモードへ前記電子デバイスを切り替えるための前記制御
信号が、少なくとも前記電子デバイスの一部をパワーオフモードからパワーオンモードへ
切り替えるためのものである請求項1乃至2のいずれかに記載の前記システム。

【請求項4】

前記第一のモードから前記第二のモードへ前記電子デバイスを切り替えるための前記制御
信号が、少なくとも前記電子デバイスの一部をパワーオンモードからパワーオフモードへ
切り替えるためのものである請求項1乃至2のいずれかに記載の前記システム。

【請求項5】

請求項1乃至4のいずれかに記載の前記システムにおいて、
前記制御サブシステムが、前記位置感知サブシステムからの空間位置信号の受信に応じて
、前記電子デバイスに前記第一及び第二の両方の制御信号を与える前記システム。

【請求項6】

前記所定の姿勢特性がある範囲の姿勢値を含む請求項2乃至5のいずれかに記載の前記シ
ステム。

10

【請求項7】

前記所定の姿勢特性がある特定の姿勢値と前記特定の姿勢値周辺の所定の許容値とを含む
請求項2乃至5のいずれかに記載の前記システム。

【請求項8】

前記所定の姿勢特性が姿勢変化率を含み、前記姿勢変化率が所定の姿勢変化率閾値を満たす
請求項2乃至5のいずれかに記載の前記システム。

【請求項9】

前記空間位置がある範囲の位置を含む請求項1及び3乃至5のいずれかに記載の前記シス
テム。

【請求項10】

前記空間位置がある特定の位置と、前記特定の位置の周辺の所定の許容値とを含む請求項
1及び3乃至5のいずれかに記載の前記システム。

20

【請求項11】

前記電子デバイスの位置の変化率を感知して、位置変化率信号を生成する、前記電子デバ
イスに接続された位置変化率感知サブシステムを更に含み、前記制御サブシステムによ
って前記電子デバイスに与えられる前記制御信号の一つが前記位置変化率信号に応じる請求
項1及び3乃至5のいずれかに記載の前記システム。

【請求項12】

前記電子デバイスの加速度を感知して、加速度信号を発生する電子デバイスに接続された
加速度感知サブシステムを更に含み、前記制御サブシステムによって前記電子デバイスに
与えられる前記制御信号の一つが加速度信号に応じる請求項1及び3乃至5のいずれかに
記載の前記システム。

40

【請求項13】

ユーザー入力部をさらに有し、前記制御サブシステムが、前記ユーザー入力部を用いて前
記範囲を調節するオプション手段をユーザーに与える請求項6と9のいずれかに記載の前
記システム。

【請求項14】

ユーザー入力部をさらに有し、前記制御サブシステムが、前記特定の姿勢値と、前記特定
の姿勢値の周辺の前記所定の許容値を前記ユーザー入力部を用いて調節するオプション手
段をユーザーに与える請求項7に記載のシステム。

50

【請求項 15】

ユーザー入力部をさらに有し、前記制御サブシステムが、前記特定の位置と、前記特定の位置の周辺の前記所定の許容値とを前記ユーザー入力部を用いて調節するオプション手段をユーザーに与える請求項 10に記載のシステム。

【請求項 16】

ユーザー入力部をさらに有し、前記制御サブシステムが、前記姿勢変化率と前記所定の姿勢変化率閾値とを、前記ユーザー入力部を用いて調節するオプションをユーザーに与える請求項 8に記載のシステム。

【請求項 17】

ユーザー入力部をさらに有し、前記制御サブシステムが位置の変化率と前記所定の変化率閾値とを前記ユーザー入力部を用いて調節するオプションをユーザーに与える請求項 11に記載のシステム。 10

【請求項 18】

前記制御サブシステムが、前記空間位置信号から得た位置情報を受信する位置トリガを含む請求項 1 及び 3 乃至 5のいずれかに記載の前記システム。

【請求項 19】

前記位置トリガが前記位置情報に応じて前記制御信号を与える請求項 18に記載のシステム。

【請求項 20】

前記制御サブシステムが前記姿勢信号から得た姿勢情報を受信する姿勢トリガを含む請求項 2 乃至 5のいずれかに記載の前記システム。 20

【請求項 21】

前記姿勢トリガが前記姿勢情報に応じて前記制御信号を与える請求項 20に記載のシステム。

【請求項 22】

前記制御サブシステムが、振舞いのパターンを検出した時に、振舞いのパターンを記録し、且つ制御信号を提供する起動化プロファイルサブシステムを含む請求項 1 乃至 2 1のいずれかに記載の前記システム。

【請求項 23】

前記起動化プロファイルサブシステムが起動化インターバルトリガを有し、前記空間位置信号から得た空間位置情報が、観察された起動化インターバルトリガ値を含む請求項 2 2に記載のシステム。 30

【請求項 24】

前記観察された起動化インターバルトリガ値と記録された起動化インターバルトリガ値との比較に応じて前記起動化インターバルトリガが前記制御信号を与える請求項 2 3に記載のシステム。

【請求項 25】

前記起動化インターバルトリガが、複数の観察されたモード変更タイムインターバル値から前記起動化インターバルトリガ値を学習する請求項 2 3に記載のシステム。

【請求項 26】

前記起動化プロファイルサブシステムが反復アクショントリガを有し、前記姿勢信号が一連の観察された姿勢読み取り値を含む請求項 2 2に記載のシステム。 40

【請求項 27】

前記一連の観察された姿勢読み取り値と記録された反復アクショントリガ姿勢設定値との比較に応じて前記反復アクショントリガが前記制御信号を与える請求項 2 6に記載のシステム。

【請求項 28】

前記起動化プロファイルサブシステムが反復距離トリガを含み、前記空間位置信号から得た空間位置情報が、観察された距離値を含む請求項 2 2に記載のシステム。

【請求項 29】

前記観察された距離値と記録された反復距離値との比較に応じて前記反復距離トリガが前記制御信号を与える請求項28に記載のシステム。

【請求項30】

前記起動化プロファイルサブシステムが複数の起動化プロファイルにアクセスするようになっており、各々の起動化プロファイルが前記システムを使用することのできる別のユーザーと関連する請求項22に記載のシステム。

【請求項31】

前記起動化プロファイルの各々が、起動化インターバルトリガ、反復アクショントリガ及び反復距離トリガの中の少なくとも一つの設定値を有する請求項30に記載のシステム。

【発明の詳細な説明】

本出願は「電気デバイスの性能を改善するためのユーザーアクションを予想するシステム及び方法」(1996年5月22日出願、発明者John Ellenby、Peter Malcolm Ellenby及びThomas William Ellenby)という名称の仮特許出願No.60/018,405の利益を請求するものである。この仮特許出願No.60/018,405は本明細書に参考文献として組み入れられている。

発明の分野

本発明は電気デバイスの性能を改善する方法と装置に関し、特に様々な状況を感知し、その感知した状況に応じて電気デバイスを制御する方法と装置に関する。

発明の背景

電子デバイスは通常特定の機能を果たすように設計されている。スイッチが入った瞬間から電子デバイスは完全に機能し設計されたそのタスクを果たす。このアプローチは単純なデバイスについてはうまく機能することもある。しかし複雑なシステムにおいてはこのアプローチの不利な面が露呈する場合がある。例えば、複雑なシステムでは相対的に多量の電力が消費されることがある。複雑なシステムが常時完全に機能している場合、システムが実際には使用されていないときでも電力消費量は通常高いままの状態となっている。この相対的に多量の電力消費は、多量の電力消費が原因で動作寿命が短くなる電池駆動式システムについては特に重要な関心事となる。

従来型の電子デバイスのもう一つの特徴として、いくつかのオペレーションモードがあつて、スイッチが入るとそれらのモードの中の一つでオペレーションが開始するように設計できることがある。そのようなデバイスでは、例えば所定のボタンを押すというようなユーザーとの物理的対話処理に応じるだけでモードを切り替えることができる。そのような設計は単純なデバイスについてはうまく機能することもある。しかし複雑なシステムでは、一つのオペレーションモードからもう一つのオペレーションモードへの切り替えに相対的に長い時間がかかる場合がある。さらに、ユーザーが複数のモード間でシステムを手動で切り替えなければならないのは不便な場合もある。もし複雑なシステムのスイッチが入ったとき、いつも一つのモードでオペレーションが開始されたり、手動選択したときのみもう一つのモードに切り替えられるようにシステムが設計されているならば、その切り替え時間の長さと、ユーザーがその切り替えをシステムに開始させるのに要する時間とによってその電子システムの性能は妨げられることになろう。

従って、適切なタイミングでデバイスにその構成部品のパワーをアップさせたりパワーダウンさせることにより電気デバイスの電池寿命を伸ばす方法及び装置が望まれている。また、ユーザーが決定したある一定のモードでのデバイスの作動時刻と実際のモード切り替えとの間の呼出時間(すなわち遅延時間)を短くする方法及び装置も望まれている。呼出時間とは、ユーザーがデバイスのスイッチのオン・オフを決定する時刻とデバイスのスイッチの実際のオン・オフとの間の呼出時間を指す場合もある。

概要

本発明の一つの局面は、ユーザーがデバイスを現在使用中であるかあるいは今から使用しようとしているかどうかを感知することによって電気デバイスを制御する方法及び/又は装置を提供することである。この方法及び/又は装置は、電力が切れかかっている構成部品を必要にしそうなときだけ、電気デバイスに構成部品を起動させることができる。この方法及び/又は装置はまた、電力が切れかかっている構成部品を必要としないとき、電気

10

20

30

40

50

デバイスに構成部品を非起動にさせることもできる。

本発明の一つの局面は、ユーザーが電気デバイスの所望のオペレーションモードへいつ切り替えたがっているか、また、それに応じてデバイスのモードをいつ切り替えるかを感じすることによって電気デバイスを制御する方法及び／又は装置を提供することである。本発明の一つの局面は、構成部品を起動及び／又は非起動にしたり、例えば、デバイスの位置の変化、ターゲット物体または物体の位置への、あるいはそれらからのデバイスの距離、視覚によるイベントまたは反復イベントのような様々なイベント及び／又は状況に応じて電気デバイスのモード間の切り替えを行うことである。本発明の代替実施例では、デバイスの加速率または加速率の変化のような異なるタイプの条件が述べられている。発明の実施例では電気デバイスが構成部品を必要としそうな可能性がだんだん大きくなるにつれて本電気デバイスの構成部品を迅速に起動させることができる。とりわけ、感知された状況に基づいて、電気デバイスが所望のモードに入りそうで、かつ、その所望のモードで電気デバイスの特定の構成部品が必要になりそうに思われる場合には、電気デバイスが構成部品を必要としそうな可能性がだんだん大きくなるにつれて本発明の実施例は電気デバイスの個々の構成部品を様々な時点で適時起動させることができる。特定の構成部品が必要になりそうな兆候がより多く感知される（例えば、電子デバイスを特定のモードに入れようとするユーザーの意図の兆候が感知される）につれて、所望のモード専用のリソースなどを増加させるようにしてこの迅速な起動化を実現することができる。同様に、本発明の実施例では、構成部品がだんだん必要とされなくなるにつれて電気デバイスの構成部品の迅速な非起動化を可能にすることができる。10

【図面の簡単な説明】

本発明のこれらのまたその他の特徴と利点は、添付図面と関連して行う以下の詳細な説明から当業者に明らかになるであろう。

図1は本発明の一つの実施例であるシステム100を例示する。

図2は本発明の代替実施例であるシステム200を例示する。

図3はシステム100のオペレーションのメインフローを例示する。

図4はシステム200のオペレーションのメインフローを例示する。

図5は本発明の実施例で使用できるタイムトリガのオペレーションを例示する。

図6は本発明の実施例で使用できる起動化プロファイルサブシステムのオペレーションを例示する。30

図7は本発明の実施例で使用できる起動化インターバルトリガのオペレーションを例示する。

図8は本発明の実施例で使用できる反復アクショントリガのオペレーションを例示する。

図9は本発明の実施例で利用できる反復距離トリガのオペレーションを例示する。

図10は本発明の実施例で利用できる姿勢トリガのオペレーションを例示する。

図11は本発明の実施例で利用できる位置トリガのオペレーションを例示する。

図12は本発明の実施例で利用できるタイムトリガを例示する。

図13は本発明の実施例で利用できる姿勢トリガを例示する。

図14は本発明の実施例で利用できる位置トリガを例示する。

図15は本発明の代替実施例であるシステム1500を例示する。40

図16はシステム1500のオペレーションのメインフローを例示する。

図17と図18は、本発明の実施例で利用できるユニットモーションサブシステムに因るグラフィック制限部のオペレーションを例示する。

図19は、本発明の実施例で利用できる表示利用サブシステムのオペレーションを例示する。

図20は本発明の実施例で利用できるスリープサブシステムを例示する。

図21は本発明の実施例で利用できるユニットモーションサブシステムに因るグラフィック制限部を例示する。

図22は図2と図7の起動化インターバルトリガのハードウェアを例示する。

図23は図2と図8の反復アクショントリガのハードウェアを例示する。50

図24は図2と図9の反復距離トリガのハードウェアを例示する。

図25はユニットモーションサブシステムに因るグラフィック制限部のハードウェアを例示する。

図26は表示利用サブシステム1517のハードウェアを例示する。

詳細な説明

本発明の一つの実施例には、電子デバイスと関連する方法との性能を改善するための新規システムが含まれる。当業者が本発明を実施し利用できるように以下に説明を行う。具体的な応用例の説明はただ例示としてのみ行われるものである。推奨実施例に対するさまざまな変更例が当業者の眼には容易に明らかとなるであろう。本発明の精神と範囲から逸脱することなく本明細書に定義された一般原理を他の実施例や応用例に適用してもよい。従って、本発明は開示された実施例に限定されることを意図するものではなく、本明細書に開示された原理と特徴に矛盾しない最も広範な範囲に一致すべきものである。

図1は、本発明の第一の実施例であるシステム100のブロック図を例示する。このシステム100は、電子デバイス110の物理的特性をモニターしたり、電子デバイス110を起動させたり、あるいは他の方法でこのデバイスを制御するために他の所定の条件をモニターしたりする、デバイス使用検出システムである。デバイスを使用する意図の兆候や他の所定の条件をシステムが検出すると、システム100はデバイス110を起動したり制御したりする。システム100には、クロック101、ユーザー入力部102、システム出力部103、位置感知デバイス104、姿勢感知デバイス105、タイムトリガ106、及び予想/呼出時間 - 低減サブシステム107が含まれる。クロック101、タイムトリガ106、及び予想/呼出時間 - 低減サブシステム107によって制御サブシステムが形成される。代替実施例は、この特定の制御システムや、この特定の制御システムに構造的に同等の制御システムに限定されるものではない。ユーザー入力部102は、例えば、キーボード、マウス、スクロールキー及びグラフィカルユーザーインターフェースを無制限に含む任意の形の入力メカニズムにすることができるし、あるいは何らかの形の磁気式、光学式あるいは電子記憶装置にすることができる。システム出力部103は、例えば、システム100に外部のユーザーやもう一つのタイプの電子デバイスと通信させることができる任意の形の出力部とすることができます。従って、システム出力部103は、表示装置や、ネットワークポートのような何らかの形の通信ポートにすることもできる。例えば無線及び/又は赤外線伝送のような遠隔伝送技術を用いて入力部と出力部の双方をつくることができる。感知デバイスと制御サブシステムとの間の通信は、遠隔伝送を含む様々な技術を用いて行うこともできる。例えば慣性ナビゲーションシステムのような他の位置感知デバイスを利用することができるが、通常位置感知デバイス104はGPSやGLONASのような衛星ベースの位置決定システムである。傾角計やレーザーリングジャイロのような他の姿勢感知デバイスを使用することもできるが、通常姿勢感知デバイス105は、フラックスゲートコンパスや三軸磁力計のような磁束感知デバイス105である。予想/呼出時間 - 低減サブシステム107には、さらに位置トリガ108と姿勢トリガ109が含まれる。タイムトリガ106、姿勢トリガ109及び位置トリガ108の実施例をそれぞれ図12、13、14に示す。

図1に示すように、クロック101とユーザー入力部102の出力部はタイムトリガ106に接続している。ユーザー入力部102、タイムトリガ106の出力部、位置感知デバイス104及び姿勢感知デバイス105は、予想/呼出時間 - 低減サブシステム107、従って位置トリガ108と姿勢トリガ109に接続している。予想/呼出時間サブシステム107の出力部は電子デバイス110に接続している。

図3は、システム100の概略的オペレーションを示すフローチャート300である。ステップ301でユーザーはシステムモニターモードを起動する。このモードでは、システム100がタイムトリガ106をモニタし(ステップ302)、ある一定の定義された条件について予想/呼出時間サブシステム107を周期的にモニターする(ステップ303)。ある一定の定義された条件が満たされた場合には、システム100は電子デバイス110を起動する(ステップ304)。例えば、電子デバイスの感知された物理的条件に応じて電子デバイスを制御するために使用する信号を制御信号と呼ぶことにする。本発明の実施例では、そのような定義された

10

20

30

40

50

条件とは、ある一定の物体あるいは位置からの所定の距離の範囲内に電子デバイス110が入ったりその範囲から出たりすること、ユーザーが電子デバイス110の所定の近辺の範囲内に入ったりそこから出たりすること、電子デバイス110の振動あるいは電子デバイスの姿勢としてもよい。代替実施例では、電子デバイス110の加速または減速の変化などを無制限に含む電子デバイス110の他の物理的特性を感知するものであってもよい。電子デバイスの切り替えモードとして電子デバイスの起動化または非起動化を考えることができる。そのようなモード切り替えには、デバイス全体の起動化及び／又は非起動化あるいはデバイスのほんの一部の起動化及び／又は非起動化を含めてよい。

図5はタイムトリガ106のオペレーションを示すフローチャート500である。

このトリガ106は、システム100がその予想/呼出時間 - 低減サブシステムをモニターする期間の遅延を実行する。このトリガ106はまたタイム起動化ルーチンも実行する。タイム起動化ルーチンにおいてユーザーは、起動化ルーチンの開始時刻からある指定時間内にデバイス110を起動または非起動にしたい旨を示すこともできる。ユーザーはそのような起動化ルーチンを利用して10分毎にデバイス110を起動するようにトリガ106に指令することもできる。本発明の代替実施例では、タイム起動化ルーチンを利用して、ユーザー指定の回数でデバイス110の一部を迅速に起動することができる。デバイス110の異なる構成部品のスイッチを様々な回数でオンにすることが望ましい場合にそのようなシステムを利用することができます。従って、そのようなシステムを利用して、パワーアップするのに時間がかかる第一の構成部品のスイッチをオンにしたり、最も多くの電力を消費する最後の構成部品のスイッチをオンにしたりすることができる。そのようなシステムを利用して感知レベルを設けることもできる。特に、このシステムの感知デバイスを利用して、電子デバイスがターゲット位置のある一定の範囲内に入ってくるまでその位置を大まかに見積もることもできる。いったんその範囲内に入ると、システムは、（例えば第一の感知デバイスよりもっと正確で、もっと多くの電力を消費する）もう一つの感知デバイスをパワーアップすることができる。

ステップ501では、モニター限界値Wはトリガ106が使用するソフトウェアによって定義される。トリガ106によって格納される限界値Wは、システム100がサブシステム107をモニターするモニター期間中の遅延時間である。例えば、W=50msの場合、システム100はタイムトリガ106で50msの間作動する。50ms後、システム100は分岐して予想/呼出時間 - 低減サブシステム107をモニターする。

トリガ106が分岐してサブシステム107をモニターする前にどれくらいの期間ユーザーがタイムトリガ106を作動させたいかによって、この限界値Wを調節することができる。代替実施例では他のトリガでも同様の遅延時間を用いることができる。

図5のステップ502で、システム100はクロック101からタイムトリガ106へ現在時刻信号を送信する。以下に論じるトリガ108と109は、システム100が様々なサブシステム（トリガ108と109など）と感知デバイス（デバイス104と105など）をどのように利用して状況をモニターするかの例を示すものである。図5のステップ503～509はタイムトリガ106によって使われるタイム起動化ルーチンを処理し、電子デバイス110や電子デバイス110の一部をユーザー定義期間で起動する。ステップ503でトリガ106はそのような起動化ルーチンがアクティブになっているかどうかを判定する。アクティブになっていない場合にはフローチャート500はステップ510へ分岐する。そのようなルーチンがアクティブになっている場合には、フローチャート500はステップ504へ分岐する。ステップ504では、トリガ106はタイム起動化ルーチンが開始されているかどうかを判定する。開始されている場合には、トリガ106は現在時刻を“最後にチェックされた”時刻Yとして格納し（ステップ506）、所望の起動化タイムインターバルXを入力する（ステップ507）ようにユーザーに促し、Xが格納される（ステップ508）。このインターバルXは、トリガ106が、“最後にチェックされた”時刻Y（すなわちタイム起動化ルーチンが開始された時刻）から待機してデバイス110が起動されるまでの時間である。ステップ504で、タイム起動化ルーチンがすでにアクティブになっているとトリガ106が判定した場合には、トリガ106はステップ505へ分岐する。ステップ505で、トリガ106によって、最後にチェックされた時刻Yから現在時刻までの経

10

20

30

40

50

過時間が計算され、ステップ509でこの経過時間が起動化インターバルXと比較される。経過時間がX以上であれば、フローチャートは図3のステップ304へ分岐し、システム100はデバイス110を完全に起動する。ステップ509で経過時間がX未満の場合には、フローチャート500は図5のステップ510へ分岐する。ステップ510でトリガ106は新しい現在時刻信号を受信し、ステップ502で受信した最後の受信時刻信号とステップ510で受信した新しい現在時刻信号との間の時間差を計算する。トリガ106は、この計算された時間差を経過時間Zに加える。次いでトリガ106はステップ512で経過時間Zをモニター限界値Wと比較する。ZがWより大きいかまたは等しければ、トリガ106はZをゼロにセットし(ステップ514)、次いで図10のステップ1001へ進み、予想/呼出時間 - 低減サブシステム107をチェックする。このようにして、ステップ512は、システム100が分岐して予想/呼出時間 - 低減サブシステム107をモニターする前にタイムトリガ106が作動する総時間数を限定する。ステップ512でZがW未満ならば、トリガ106は、ユーザーがデバイス110のスイッチをオフにしているかどうかを調べるチェックを行い(ステップ513)、次いでステップ502へ戻って次の現在時刻信号を受信する。ステップ513を使って、システム100自身がデバイス110のスイッチをオフにしたかどうかを判定したり、他のいずれかのデバイスがデバイス110のスイッチをオフにしたかどうかを判定したりしてもよい。

図10は、姿勢トリガ109のオペレーションを示すフローチャート1000である。姿勢感知デバイスが感知する電子デバイスの物理的特性を姿勢特性と呼ぶことにする。この姿勢トリガはこれらの姿勢特性を表す姿勢情報を受信する。この姿勢情報は、姿勢感知デバイスから入来する姿勢信号から得られる。本実施例では、ソフトウェアを実行するハードウェアを用いて姿勢トリガ109が実行される。図13を参照しながらこのハードウェアを説明する。フローチャート1000は、姿勢トリガ109のソフトウェアのオペレーションを例示する。電子デバイス110の姿勢値が指定されている率より高く変化しつつあるかどうか、あるいは、電子デバイス110の姿勢が指定されている量以上に定常状態の姿勢値から変化してしまっているかどうかの判定チェックがこの姿勢トリガによって行われる。その姿勢値がこの指定されている率より高く変化しつつあれば、あるいは、指定されている量以上に変化してしまった場合には、システム100は電子デバイス110を起動する。

フローチャート1000のステップ1001によって、“角度/秒”起動化限界値C、“定常状態からの角度”起動化限界値D、及び“新しい定常状態Aの記録”限界値Eが定義される。ステップ1002で、トリガ109は姿勢感知デバイス105からの現在の姿勢信号を受信し、ステップ1003で、“定常状態”的姿勢値Aと“最後に受信した”姿勢値Bが以前に格納されているかどうかを調べるチェックが行われる。AとBの値が格納されていない場合、トリガ109は、AとBの両方として姿勢センサーから受信した現在の姿勢値をステップ1004で格納し、次いで、フローチャート1000は図11のステップ1101へ分岐して位置トリガ108をチェックする。AとBの値が以前に格納されているとステップ1003が判定した場合、フローチャート1000はステップ1005へ分岐し、そこで、トリガ109によって、現在の姿勢値と“最後に受信した”姿勢値Bとの間の差が計算される。次いでトリガ109は、この姿勢値の差をWで割り、電子デバイス110の姿勢の変化率を表す角度/秒値を算出する(ステップ1006)。次いで、トリガ109は、この計算された角度/秒を“角度/秒”起動化限界値Cと比較する(ステップ1006)。この計算された角度/秒値がCを超える場合、フローチャート1000は図3のステップ304へ分岐し、システム100がデバイス110を完全に起動する。この計算された角度/秒値がCを超えない場合、フローチャート1000はステップ1007へ分岐する。

ステップ1007でトリガ109は現在の姿勢値と最後に受信した姿勢値Bとの間の差を決定する。この差は、“新しい定常状態Aの記録”限界値Eと比較される。限界値Eとは姿勢変化閾値である。例えば、姿勢値の変化がこの閾値未満であれば定常状態の姿勢値Aと現在の姿勢値Bが更新されることはない。この閾値以上の姿勢値の変化によってこれらの姿勢値AとBは更新される。このようにしてこの限界値Eによって、電子デバイス110の小さな動きが新しい定常状態値Aとして記録されることが阻止される。言い換えれば、限界値Eとは、システム100や電子デバイス110のわずかなまたは些細な動きにより定常状態値Aがリセットされることを防ぐための値である。

10

20

30

40

50

従って、姿勢値の変化がEを超える場合、フローチャート1000はステップ1008へ分岐し、そこで、“定常状態”の姿勢値Aと“最後に受信した”姿勢値Bの双方として現在の姿勢読み取り値が格納される。姿勢値の変化がEを超えない場合、フローチャートはステップ1009へ分岐し、そこで、トリガ109によって現在の姿勢値と“定常状態”的姿勢値Aとの間の差が計算される。次いで、トリガ109は、この差を“定常状態の角度”起動化限界値Dと比較する（ステップ1010）。この差がDを超える場合、フローチャート1000は図3のステップ304へ分岐し、システム100はデバイス110を起動する。この差がDを超えない場合、フローチャートはステップ1011へ分岐し、Bとして現在の姿勢読み取り値が格納され、次いで図1のステップ1101へ分岐して位置トリガ108がチェックされる。従って、デバイス110が姿勢トリガ109によって起動されない場合、システム100は次に進み位置トリガ108をモニタ¹⁰する。

図11は、位置トリガ108のオペレーションを示すフローチャート1100である。位置感知デバイスが感知する電子デバイスの物理的特性を位置特性と呼ぶこととする。この位置トリガはこれらの位置特性を表す位置情報を受信する。この位置情報は位置感知デバイスから入来する位置信号から得られる。本実施例ではこのオペレーションはソフトウェアを実行するハードウェアを用いて実施される。図14を参照しながら位置トリガを実行するために使用されるハードウェアについて論じる。フローチャート1100はこのソフトウェアのオペレーションを例示する。この位置トリガは位置起動化ルーチンを実行する。位置トリガ108は、電子デバイス110の位置が、位置起動化ルーチンが起動されたときのデバイスの位置から指定されている量だけ変化したかどうかの判定チェックを行う。位置108はまた、ユーザーが指定する関心領域または地点への電子デバイス110の接近などをチェックする。ステップ1101で位置モニターモードが起動され、位置感知デバイス104が位置情報をトリガ108へ送信する。ステップ1102～1109は、ユーザー定義の運動距離やユーザー定義の位置で電子デバイス110または電子デバイス110の一部を起動する位置起動化ルーチンを処理する。例えば、ユーザーはシステム100をセットして電子デバイス110を50フィート毎に起動化することができる。ステップ1102では、トリガ108はそのような起動化ルーチンがアクティブであるかどうかを判定する。アクティブでない場合には、フローチャート1100はステップ1110へ分岐する。そのようなルーチンがアクティブな場合には、フローチャート1100はステップ1103へ分岐し、アクティブな位置起動化ルーチンが開始されたばかりかどうかを判定する。このルーチンが開始されたばかりであれば、トリガ108は、“最後のステップ”位置Gとしてステップ1101で受信したデバイス110の現在の位置を格納し（ステップ1104）、ユーザーに所望の起動化距離Fを入力するように促し（ステップ1105）、Fを格納する（ステップ1106）。ステップ1103によってそのような起動化ルーチンがすでにアクティブになっていたと判定された場合には、トリガ108はGから現在の位置までの距離を計算し（ステップ1107）、次いで、この計算された距離をユーザー指定の起動化距離Fと比較する（ステップ1108）。この計算された距離が距離F以上である場合、フローチャート100は図3のステップ304へ分岐し、システム100はデバイス110を完全に起動する。この計算された距離がF未満の場合、フローチャート1100はステップ1110へ分岐する。本発明の代替実施例ではいくつかのユーザー指定距離を処理する位置起動化ルーチンを用いて機能させることもできる。これらの距離を用いて例えばデバイス110の迅速なパワーアップやパワーダウンを行うこともできる。特に、さまざまな距離でデバイス110の特定の一部をパワーアップまたはダウンしてデバイス110の迅速なパワーアップやパワーダウンを達成することもできる。

ステップ1110～1112は、ユーザーが指定した関心領域または地点への電子デバイス110の接近を処理する。例えば、ユーザーは、デバイスが、指定した地点または領域のある一定の距離の範囲内に入ったとき自分がデバイス110またはデバイス110の一部を起動させたい旨を示してもよい。例えば、そのデバイスがオークランド（Auckland）港外側マーカーR2の半マイル以内にあるとき、ユーザーはデバイス110をアクティブにすることを望むことができる。ステップ1110で、システム100はユーザーが関心のある地点/領域を指定したかどうかを調べるチェックを行う。指定されていなかった場合には、フローチャート1100は

10

20

30

40

50

図5のステップ502へ分岐し、タイムトリガ106のモニターへ戻る。ユーザーがそのような関心のある地点/領域を指定していた場合、フローチャート1100はステップ1111へ分岐し、そこでトリガ108は、そのような各地点/領域までの、または、そこからの距離を決定する。ステップ1112で、トリガ108は、各地点/領域と関連するユーザー指定の起動化距離を決定された距離と比較する。この決定された距離のいずれかが関連する起動化距離未満の場合には、フローチャート1100は図3のステップ304へ分岐し、システム100はデバイス110を完全に起動する。決定された距離のいずれも起動化距離未満ではない場合、フローチャート1100は図5のステップ502へ分岐し、上記のようにシステム100はタイムトリガ106のモニターへ戻る。

図2は、本発明の第二の実施例であるシステム200のブロック図である。システム200には、システム100に関して説明した構成部品と同じように作動する構成部品が含まれる。類似する構成部品には同じような番号がつけられている（例えばタイムトリガ206はタイムトリガ106と同じように作動する）。そのような類似の構成部品を持っていることに加えて、システム200には起動化プロファイルサブシステム210が含まれる。このサブシステム210はシステム200の予想/呼出時間 - 低減サブシステム207の一部として含まれる。

ユーザーが行う反復アクションやデバイス110の起動化以前に生じた他の反復状況をシステム200が使用期間にわたって認識できるように、サブシステム210は設計されている。この認識に基づいて、サブシステム210は特定の反復アクションや特定の反復状況と関連する明確な起動化プロファイルを展開する。これらの起動化プロファイルによってシステム200は、差し迫った使用を示すものとして特定の反復アクションや状況を認識することが可能になる。それに応じて、システム200は、デバイス214の迅速な起動化を行ったり開始したりすることができる。この起動化プロファイルサブシステム210には、起動化インターバルトリガ（AIT）211、反復アクショントリガ（RAT）212、及び反復距離トリガ（RDT）213が含まれる。本発明の代替実施例では、本明細書に論じられている明確な反復状況以外の反復状況に応答するように設計することもできる。

図4は、システム200の概略的オペレーションを示すフローチャート400である。このオペレーションは、いくぶん変更はあるがシステム100のオペレーションとほぼ同じである。ステップ401はステップ301とほぼ同じである。ステップ402はステップ302とは異なる。特に、図5のフローチャート500は、図4のステップ402のオペレーションを実質的に説明するものである。しかしフローチャート500のステップ512において、もしZがW以上であれば、システム200のフローチャート500はステップ514を実行し、次いで図10のステップ1001へは分岐せずに図6のステップ601へ分岐する。この変更は、図3のフローチャートコネクタ25を図4のフローチャートコネクタ3と置き換えることによって示されている。図4のステップ403は、起動化プロファイルサブシステム210のオペレーションと姿勢トリガ208と位置トリガ209のオペレーションとを表す。サブシステム210のオペレーションを以下にもっと詳細に説明する。以下に論じる起動化プロファイルフローチャートの終端で判定することができるように、姿勢トリガ209及び位置トリガ208とは、それぞれ、これらのトリガが起動化プロファイルサブシステム210の後で作動するということを除いて上記に論じたようにトリガ109と108と同じように作動する。従って、システム200のオペレーションフローは、このフローの間電子デバイス214が起動されていないと仮定すると一般に図5から図6、図7、図8、図9、図10次いで図11となる。いったん図11のオペレーションが実行されると、電子デバイス214がまだシステム200によって起動されていないと仮定して、システム200のオペレーションは図11のコネクタ5を通って図5のステップ502ヘループして戻ってくる。従って、システム200は、デバイス214が起動される前にこれらのフローチャートを通ってマルチパスを行うこともできる。

図5のステップ514から図6のステップ601へ分岐後に、システム200は起動化プロファイルサブシステム210をチェックする。この起動化プロファイルサブシステム210は、ソフトウェアを実行する単複のプロセッサや他のタイプのハードウェアを使ってトリガ106、109と108などと同じように実施することができる。図6は、起動化プロファイルサブシステム21

10

20

30

40

50

0のソフトウェアの基本的オペレーションを示すフローチャート600である。ステップ601で、サブシステム210は起動化プロファイル(AP)がアクティブであるかどうかを調べるチェックを行う。すなわち、サブシステム210が特定の反復状況を検索するようにユーザーや他の何らかのデバイス(コンピュータなど)によって指示を受けているかどうかがステップ601によって判定される。もし指示を受けていればフローチャート600はステップ602へ分岐する。起動化プロファイルがアクティブでないとステップ601でサブシステム210が判定した場合、フローチャート600はステップ603へ分岐する。

ステップ602でサブシステム210は、現在アクティブであるAPとは異なるAPを使うようにシステム200が指示されているかどうかを確かめる。再言するが、そのような指示はシステム200のユーザーや他のなんらかのデバイスから来たものであってもよい。ステップ602で、異なるAPを使用するように指示されていたとシステム200が判定した場合、フローチャート600はステップ603へ分岐する。ステップ602で、異なるAPの使用が指示されていなかつたとシステム200が判定した場合、フローチャート600はフローチャート700のステップ701へ分岐する。フローチャート700はAIT211のオペレーションを例示する。

ステップ603で、システム200は、例えばユーザーが現在のAPを選択したかどうかを確かめる。APが選択されていればサブシステム210のソフトウェアはステップ607へ分岐する。ステップ607でシステム200は選択されている現在のAPを取り消し、現在のAP定義設定値をAPトリガ211、212及び213の各々へ与える。

ステップ607に例示されているように、起動化プロファイルはこれらのトリガの中のいくつかあるいはすべてを用いて状況を感知することもできる。以下に論じるように、これらのトリガの各々は異なるタイプの状況を感知する。代替実施例では他のタイプの状況を感じる代替トリガを用いることもできる。トリガ211、212及び213の中の適切なトリガにAP定義の設定値を入力後、フローチャート600は図7~9で展開されているステップ608へ分岐する。

ステップ603で、現在のAPが選択されていないとシステム200が判定した場合、フローチャート600はステップ604へ分岐する。ステップ604でシステム200は、展開時に新しいAPを保存すべきかどうかを確かめる。以下に説明するように、APサブシステム211、212及び213の中をシステム200が通るたびにリストに値を保存する本実施例は、どのようにAPを展開できるかという一例を示すものである。再言するが、新しいAPが展開するときそのAPを保存せよというシステム200への指令は、システム200のユーザーや他のなんらかのデバイスから来たものであってもよい。新しいAPが展開するときそのAPを保存せよという指令をシステム200が受けた場合、この新しいAPに名前をつけるようにシステム200はユーザーに促す(ステップ604)。この名前を受け取ると、システム200はデフォルトのAPをその名前で格納し(ステップ605)、次いでフローチャート600はコネクタ26を通じて図7へ分岐する。このデフォルトのAPはシステム200の特定のアプリケーションにとって適切な任意の方法で定義することができる。本実施例では、デフォルトAPにはトリガ211、212及び213について定義された設定値はない。ステップ608は、起動化プロファイルトリガ211、212及び213のオペレーションを表す。このオペレーションは図7から開始される。トリガ211、212及び213

のオペレーションを以下にもっと詳細に説明する。

図7は、起動化インターバルトリガ211のオペレーションを示すフローチャート700である。起動化インターバルトリガ211はハードウェアとソフトウェアを用いて実行される。以下の図22を参照しながらこのハードウェアについて論じる。本実施例ではフローチャート700はAITソフトウェアのオペレーションを表す。再言するが、この起動化インターバルトリガ211は起動化プロファイルサブシステムによって使用され反復状況が検出される。この特定のトリガ211は、システム210がステップ401でモニターモードに入った時刻からデバイス214が起動された“現在時刻”までの反復経過時間を検出するためのものである。従って、システム200は、システム200がモニターモードに入る時刻とほぼ同じ時刻(所定の許容値の範囲内で)に、電子デバイス214のスイッチが前もって繰り返しオンになる場合、特定の時刻に電子デバイス214のスイッチをオンにするようにシステム200がモニタ-

10

20

30

40

50

モードに入った時刻から“学習する”。本実施例では電子デバイスの“起動化”について論じられるが、代替実施例では非起動化処理を行うこともできる。従って、本明細書では、デバイス214がパワーアップされていようとパワーダウンされていようとAITトリガと起動化インターバル値はそのようなインターバル値を指すものとする。

ステップ701で、AIT211は、AIT211のAITインターバル値が格納されたかどうかを確かめる。インターバル値が格納されていれば、フローチャート700はステップ702へ分岐する。AITのインターバル値が格納されていなければ、フローチャート700はステップ701からステップ703へ分岐する。ステップ702で、AIT211は、システム200がモニターモードに入ったときから“現在時刻”までの経過時間を計算する。但し現在時刻とはステップ502で最も最近に読み出された時刻である。この経過時間を観察された起動化インターバル値と呼ぶことにする。再言するが、このモニターモードはステップ401で起動される。オペレーションはステップ702からステップ704へ分岐する。10

ステップ704で、AIT211は、ステップ702で計算された経過時間をAITインターバル値と比較する。この経過時間がAITインターバル値と“マッチ”すれば、フローチャート700は図4のステップ404へ分岐し、システム200は完全にデバイス214を起動する。経過時間がAITインターバル値と“マッチする”とは、経過時間がこのインターバル値と等しいか、このインターバル値の所定のある範囲内にある場合を指す。この経過時間がAITインターバル値とマッチしない場合には、フローチャート700はステップ703へ分岐する。以下に説明するように、このAITインターバル値は“学習”値であってもよい。

ステップ703でAIT211は、デバイス214が、（システム200自身、ユーザーあるいは何らかの他のデバイスによって）すでに起動されているかどうかを確かめる。デバイス214がまだ起動されていない場合には、フローチャート700によって表されるようなオペレーションは、図8のステップ801へ分岐し反復アクショントリガ212がチェックされる。ステップ703で、デバイス214がすでに起動されている場合には、フローチャート700はステップ705へ分岐する。ステップ705で、システム200がモニターモードに入った時刻から、電子デバイス214が起動されたことをトリガ212が検出した現在時刻までの経過時間であるモード変更時間インターバル（MCTI）がAIT211によって計算される。システム200において、電子デバイス214は最初システム200によってパワーオフされ、次いでパワーアップされる。代替実施例では、デバイス214を最初にパワーオンし、次いでシステム200がこのデバイスをパワーダウンしてもよい。従って、MCTIとは、システム200がモニターモードに入った時刻から電子デバイス214が非起動になるまでの経過時間と指すものであると言ってもよい。システム200がデバイス214をオフモードからオンモードへ切り替える途中であれば、MCTIをダウンタイムインターバル（DTI）と呼んでもよい。システム200がデバイス214をオンモードからオフモードへ切り替える途中であれば、MCTIをアップタイムインターバル（UTI）と呼んでもよい。システム200とAIT211をダウンタイムインターバル値という点から以下説明する。30

AIT211にはDTIのリストが保管される。再言するが、各々のDTIはAIT211の一つのパスによってフローチャート700の中を通って生成される。AIT211はリストを用いながらDTIの数を追跡する。新しく計算された各DTIはDTIのこのリストの最上部に置かれる。このリストへの入力回数は通常特定のアプリケーションの要求に応じて定義されている。例えば、アプリケーションが、飛行機のような高速運動環境用のものである場合には、DTI入力回数はきわめて大きなものになる場合もある。そのような状況下では、ユーザーはシステム200のスイッチをオンにし電子デバイス214を多數回起動することになる。DTIの入力回数は以下のように生成してもよい。システム200のスイッチがオンになり、システム200が電子デバイス214のスイッチをオンにした時刻後、ユーザー、システム200自身あるいは何らかの他の電子デバイスによって電子デバイス214のスイッチをオフにしてもよい。電子デバイス214のスイッチのオフに応じてシステム200のスイッチがオフになる。この時点の以後、ユーザーがシステム200を再びオンにしてもよいし、DTIリストに第二の入力を追加してフローチャート700の中でもう一回パスを行ってもよい。DTIリストがいっぱいになった場合、新しく計算された各DTIはリスト中に（例えばリストの最上部に）に置かれ、各々の追40

1020304050

加された新しいDTIと引き換えに最も古いDTIはリストから（例えばリストの最下部から）外される。代替アプローチを用いて関連のあるDTIを格納することもできる。

ステップ706でシステム200はDTIリストがいっぱいかどうかをチェックして調べる。リストがいっぱいではない場合、フローチャート700は図8に示されるフローチャート800のステップ801へ分岐する。ステップ801でシステム200は次へ進み、適切な場合には他のAPトリガ（すなわちRAT212とRDT213）を更新する。

ステップ706で、DTIリストがいっぱいであるとシステム200が判定した場合、フローチャート700はステップ707へ分岐する。ステップ707でシステム200はDTIリストへの入力の各々を比較し、DTIの大多数がお互いの所定の許容限界値の範囲内にあるかどうかを判定する（ステップ708）。お互いに相対的に近い（すなわちお互いの所定の許容値の範囲内の）DTIの大多数グループを特定する任意の方法でこの比較を達成することができる。再言するが、この所定の許容限界値は特定のアプリケーションについて適切なものとして選ぶことができる。大多数のDTIがお互いの所定の許容値の範囲内にあるかどうかを判定する一つのアプローチは、DTIリストへのすべての入力の平均値を決定することであろう。そうすればDTIの各々をこの平均値と比較することができる。大多数のDTIが平均値について所定の許容値の範囲内にある場合には、フローチャート700はステップ709へ分岐し、そこでこの大多数のDTIは平均化される。再言するが、もっと優れたアプローチを含む他のアプローチを用いて適切な大多数についてこの特定を行うこともできよう。この特定された大多数のDTIの平均値はAIT211についての新しいAITインターバル値として保存される。次いで、フローチャート700はステップ709から図8のステップ801へ分岐し、適切な場合には他のAPトリガの更新へ進む。ステップ707で、大多数のDTIがお互いの所定の許容値の範囲内にはないとシステム200が判定した場合には、フローチャート700は図8のステップ801へ分岐し、適切な場合には他のAPトリガの更新へ進む。10

図8は、反復アクショントリガ（RAT）212のオペレーションを示すフローチャート800である。本実施例では、ハードウェアを用いてソフトウェアを実行することによってこのオペレーションは実施される。フローチャート800はこのソフトウェアのオペレーションを表す。以下図23を参照しながらRAT212によって使用されるハードウェアを説明する。この反復アクショントリガは反復する一連の姿勢値を検索する。各々の姿勢読み取り値は（x,y,z）座標として表現してもよい。ここでxは縦軸周りの姿勢回転角度を表し、yは横軸周りの姿勢回転角度を表し、zはx軸とy軸の両方に垂直な軸周りの姿勢回転角度を表す。20

従って、例えば電子デバイス214は、デバイス214が起動する前に、一連の姿勢値

$[(0^\circ, 0^\circ, 0^\circ), (20^\circ, 20^\circ, 0^\circ), (45^\circ, 45^\circ, 0^\circ), (1^\circ, 90^\circ, 90^\circ, 0^\circ), (90^\circ, 120^\circ, 0^\circ), (90^\circ, 130^\circ, 0^\circ)]$

の中を動いてもよい。デバイス214が起動する前にこの一連の姿勢値の中を（所定の許容値の範囲内で）このデバイスが規則的に動く場合には、システム200はこの一連の姿勢値を学習することができる。次いで、システム200はこの学習された一連の姿勢値を検出してすぐにデバイス214を起動することができる。

ステップ801で、RAT212は姿勢感知デバイス205から姿勢読み取り値（例えばx,y,z座標）を記録し、姿勢読み取り値のリストの最上部にその記録された姿勢値を入れる。この姿勢値リストの姿勢読み取り値（x,y,z）の数値は、通常、特定のアプリケーションの要求条件によって再定義される。姿勢読み取り値はいっぱいになるまでこの姿勢値リストに追加される。再言するが、フローチャート800の中を通る各パスによって単一の（x,y,z）の姿勢読み取り値が生成されこのリストに追加される。いったんこの姿勢値リストがいっぱいになると、新しく記録された各姿勢読み取り値がリストの最上部に置かれ、追加される各々の新しい読み取り値と引き換えに、最も古い姿勢読み取り値がリストの最下部から外される。30

ステップ802でRAT212はデバイス214がすでに起動されているかどうかを確かめる。このデバイスは、ユーザー、システム200自身または何らかの他のデバイスによってすでに起動されていてもよい。デバイス214がすでに起動されている場合には、フローチャート800はステップ806へ分岐する。デバイス214がまだ起動されていない場合には、フローチャート40

800はステップ803へ分岐する。

ステップ803でRAT212は、反復アクショントリガ（RAT）212のためのRAT姿勢設定値がすでに格納されているかどうかをチェックして調べる。RAT姿勢設定値がすでに格納されている場合には、フローチャート800はステップ804へ分岐する。そのような設定値がまだ格納されていない場合には、フローチャート800は図9のステップ901へ分岐し反復距離トリガ213をチェックする。以下に論じるようにこのRAT姿勢設定値は学習された設定値であってもよい。RAT姿勢設定値は、実際には、システム200が検索する一連の姿勢読み取り値（上の（1）に示すような一連の値）である。

フローチャート800のステップ804でRAT212は、姿勢読み取り値（すなわち観察された姿勢読み取り値）のリスト中の入力の各々をRAT姿勢設定値中の対応する姿勢値と比較する。

10

例えば、RAT212は、第一の観察された姿勢値をRAT設定値中の第一の姿勢値と比較し、第二の観察された姿勢値をRAT設定値中の第二の姿勢値と比較する等である。このRAT姿勢設定値リストは一つ以上の姿勢値を含むものであってもよい。ステップ805でRAT212は、姿勢値リスト中の姿勢読み取り値の各々が所定の許容値の範囲内の対応するRAT姿勢設定値とマッチするかどうかを確かめる。再言するが、この観察された姿勢読み取り値が、RAT姿勢設定値リスト中で対応する姿勢値の所定の許容値の範囲内にあるときにマッチは生じる。再言するが、この所定の許容値は特定のアプリケーションの要求条件に従って決定することができる。この二つのリスト中の対応する姿勢値の各々がこの所定の許容値の範囲内でマッチすれば、RAT姿勢設定値によって定義されたような検索された一連の姿勢値の中を電子デバイス214が通ったとシステム200は判定してしまう。その結果、フローチャート800は図4のステップ404へ分岐し、システム200は完全にデバイス214を起動する。この二つのリストが所定の許容値の範囲内でマッチしない場合には、フローチャート800は図9のステップ901へ分岐し、システム200は反復距離トリガ213をチェックする。

20

ステップ802でRAT212がステップ806へ分岐した場合には、ステップ806でRAT212は、デバイス214のスイッチがオンになる前に生じた一連の姿勢値を“学習する”。特に、デバイス214のスイッチがオンになる前に生じたこれらの一連の姿勢値は起動化モーションルーチン（AMR）リストへ追加される。フローチャート800の中を複数回通ることによって複数の一連の姿勢値がこのAMRリストへ追加される。従って、起動化モーションルーチンリストとは“一連の姿勢値”的な単数リストまたは複数のリストのリストである。特に、AMRリストへの各々の入力はそれ自身姿勢値のリストである。RAT212は、この姿勢値リストをステップ801から起動化モーションルーチンとして格納する。次いでRAT212はこのAMRをAMRのリストの最上部に入れる。このAMRリストへの入力回数は通常特定のアプリケーションの要求条件に従って定義される。各々の新しいAMRがAMRリストの最上部に追加される。各々の新しいAMRが追加されたためにこのリストがいっぱいになった場合最も古いAMRがこのリストの最下部から外される。

30

ステップ807でRAT212は、AMRリストが実際にいっぱいかどうかを確かめるテストをする。いっぱいではない場合には、フローチャート800は図9のフローチャート900のステップ905へ分岐して適切な場合には反復距離トリガ設定値が更新される。AMRリストがいっぱいになっている場合には、フローチャート800はステップ808と809へ分岐する。ステップ808と809でRAT212はリスト中のAMRを比較し、大多数のAMRがお互いの所定の許容値の範囲内にあるかどうかを確かめる。これを行うための一つのアプローチは、AMRを構成するこれらのリストのすべての対応する姿勢読み取り値の平均値を決定することである。例えば、RAT212は、AMRを構成するこれらのリスト中の第一の姿勢読み取り値のすべての平均値を計算してもよい。次いで、AMRを構成するこれらのリスト中の第二の姿勢読み取り値のすべての平均値を計算してもよい。等々。そのような“平均姿勢値リスト”を計算して、RAT212は、AMRリストの各々の入力中のこれらの姿勢読み取り値の各々を平均姿勢値リスト中の対応する姿勢読み取り値と比較する。特定のAMR入力の姿勢読み取り値のすべては、平均姿勢値リストから得た対応する平均姿勢値の所定の許容値の範囲内にあり、その特定のAMRはあるグループの中に含まれる。大多数のAMRがこのグループの中に含まれる場合、フローチャート800はステップ809からステップ810へ分岐し、ここで、大多数のAMR入力の各々

40

50

の中の対応する姿勢読み取り値は平均化され、平均姿勢値を示すこのリストは新しいRAT姿勢設定値として保存される。他の技術を用いてAMRのいずれがお互いの所定の許容値の範囲内にあるかを判定することができる。ステップ810からフローチャートは図9のステップ905へ分岐し、必要な場合には反復距離トリガ設定値が更新される。ステップ809で、大多数のAMRがお互いの所定の許容値の範囲内にない場合、フローチャート800は図9のステップ905へ分岐し、適切な場合には反復距離トリガ設定値が更新される。

図9は、反復距離トリガ213のオペレーションを示すフローチャート900である。本実施例では、ハードウェアを用いてソフトウェアを実行することによってこのオペレーションは実施される。フローチャート900はこのソフトウェアのオペレーションを表す。以下図24を参照しながらRDT213が使用するハードウェアを説明する。反復距離トリガ213は、システム200がモニターモードに入ったときの電子デバイス214の位置から、電子デバイスのスイッチがオンになったときの電子デバイス214の位置までの反復距離をモニターする。システム200は、電子デバイス214がモニターモードに入ったときの電子デバイス214のこの位置からのこの同じ距離でスイッチをオンにすることを“学習する”。

フローチャート900のステップ901でRDT213は、反復距離トリガ(RDT)の反復距離値がすでにアプリケーションプロファイル(AP)に格納されているかどうかを確かめる。以下に説明するようにこの反復距離値は学習された値であってもよい。反復距離値がすでに格納されている場合には、フローチャート900はステップ902へ分岐する。反復距離値がまだ格納されていない場合には、フローチャート900はステップ903へ分岐する。ステップ902でRDT213は、電子デバイス214の現在の位置から、モニターモードが起動されたときの電子デバイス214の位置までの距離を計算する。この距離を観察された距離値と呼ぶことにする。次いで、フローチャート900はステップ902からステップ904へ分岐する。ステップ904でRDT213は、ステップ902で計算された距離をこの反復距離値と比較する。この計算された距離が反復距離値と“マッチする”場合には、フローチャート900はステップ904から図4のステップ404へ分岐しシステム200はデバイス214を完全に起動する。この計算された距離がこの反復距離値の指定範囲内に入った場合にマッチが生じる。この計算された距離がその反復距離値とマッチしない場合フローチャート900はステップ903へ分岐する。ステップ903でRDT213は、デバイス214がすでに起動されているかどうかを確かめる。デバイス214がまだ起動されていない場合には、フローチャート900は図10のステップ1001へ分岐し、姿勢トリガ209と位置トリガ208のチェックを続行する。図9から図6へのコネクタ27と図6から図10へのコネクタ25とによってこの分岐は示される。トリガ209と208のオペレーションは、それぞれ、システム100を参照しながら説明したトリガ109と108のオペレーションと同じである。

ステップ903でデバイス214がすでに起動されている場合には、フローチャート900はステップ903からステップ905へ分岐する。ステップ905でRDT213は、電子デバイス214の現在の位置からモニターモードが起動された電子デバイス214の位置までの距離を計算する。この距離をモード変更距離インターバル(MCDI)と呼ぶこととする。モード変更距離インターバル(MCDI)とは、システム200がモニターモードに入った時刻から、電子デバイス214がすでに起動されていることをトリガ213が検出した現在の時刻まで電子デバイス214が動いた距離のことである。AIT211のMCTIと同様、MCDIは、本実施例におけるような、パワーダウンモードからパワーアップモード(すなわち起動化)までのデバイス214の変化と呼ぶことができる。このようなMCDIはまたダウンディスタンスインターバル(DDI)と呼ぶこともできる。同様に、MCDIはパワーアップモードからパワーダウンモードへのデバイス214の変化と呼ぶことができる。このようなMCDIはまたアップディスタンスインターバル(UDI)と呼ぶこともできる。ダウンディスタンスインターバル値という点から以下システム200を説明する。

フローチャート900の中を通る各々のパスは单一のDDIを計算する。トリガ213はステップ905で計算されたDDIをDDIのリストの最上部に入れる。DDIのこのリストへの入力回数は各々の特定のアプリケーションの要求条件に従って定義することができる。DDIリストがいっぱいになった場合は、新しく計算されたDDIはリストの最上部に置かれ、最も古いDDIが

10

20

30

40

50

リストの最下部から外される。異なる技術を用いて関連のあるDDI値を格納することができる。トリガ213はステップ905からステップ906へ分岐する。ステップ906でシステム200はDDIリストがいっぱいかどうかを確かめる。いっぱいになればフローチャート900は図4のステップ404へ分岐し、システム200は完全にデバイス214を起動する。システム200によってデバイス214が起動された後、ユーザー、システム200自身または何らかの他の電子デバイスによってデバイス214のスイッチをオフにしてもよい。この時点以後システム200のスイッチを再びオンにすることによって、DDIのリストに追加されることになる第二のDDIが生成されることになる。このDDIリストがいっぱいの場合には、フローチャート900はステップ907へ分岐する。

ステップ907でRDT213はDDIリスト中の入力をお互いに比較しステップ908へ分岐する。ステップ908でRDT213は、大多数のDDIがお互いの所定の許容値の範囲内にあるかどうかを確かめる。お互いに相対的に近い（すなわちお互いの所定の許容値の範囲内の）大多数のDDIグループを特定する任意の方法でこの比較を達成することができる。再言するが、この所定の許容限界値は特定のアプリケーションにとって適切なものとして選ぶことができる。大多数のDDIがお互いの所定の許容値の範囲内にあるかどうかを判定する一つのアプローチは、DDIリストへのすべての入力の平均値を決定することであろう。そうすればDDIの各々をこの平均値と比較することができる。大多数のDDIが平均値の所定の許容値の範囲内にある場合には、フローチャート900はステップ909へ分岐し、そこでこの大多数のDDIは平均化される。再言するが、もっと優れたアプローチを含む他のアプローチを用いて適切な大多数についてこの特定を行うこともできよう。

大多数のDDIがお互いの所定の許容値の範囲内にある場合には、システム200はデバイス214起動化のパターンを特定したことになる。システム200がモニターモードに入ったとき、このパターンでデバイス214はデバイス214の位置からほぼ同じ距離で繰り返し起動されている。大多数のDDIがこの所定の許容値の範囲内にある場合には、フローチャート900はステップ909へ分岐し、そこで大多数のDDIは平均化される。この平均値は反復距離値を表す新しい値として保存される。このステップは、システム200が探し求めている反復距離を“学習する”ステップである。フローチャート900はステップ909から図4のステップ401へ分岐し、システム200は完全にデバイス214を起動する。ステップ908で、大多数のDDIが反復距離値から離れた値となり、所定の許容値の範囲内にない場合、フローチャート900は図4のステップ401へ分岐しシステム200は完全にデバイス214を起動する。

図15は、視覚システム1514に接続しているシステム1500を例示する。このシステム1500は、視覚システム1514を制御するために使用されている本発明の一つの実施例を例示する。この視覚システム1514はデバイス214のような電子デバイスの特別の例である。視覚システム1514は、ヘッドアップ・ディスプレイのような従来からある光コンビナ（combiner）タイプの器具であってもよいし、好適には、PCT公開公報出願番号W095/07526に開示されているようなタイプの視覚システムであってもよい。「位置と姿勢を利用する電気光学式視覚システム」（公開番号W095/07526、国際出願日1994年6月16日、出願者Criticom Corp.、発明者John Ellenby及びThomas William Ellenby、国際出願番号PCT/LTS94/06844）という名称のこの公開PCT出願は本明細書に参考文献として組み入れられている。システム1500と1514はまた、制御されている電気デバイスの動きが検出されたときのグラフィックの複雑さのレベルの低下に関する追加コンセプトを例示するために利用される。これらのシステムは、検出されたユーザーの接近に基づくシステム表示の起動化または非起動化に関するコンセプトと、他のシステム間でシステムの無活動が検出されたときの電力の維持に関するコンセプトとを例示する。

図15で、システム1500の構成部品は同様の番号がつけられている図2のシステム200の構成部品と同じように作動する。この視覚システム1514には、ユニットモーションサブシステム1516、表示利用サブシステム1517、スリープサブシステム1518及び、ビデオモニターやヘッドアップ・ディスプレイ（図示せず）のようなディスプレイに因るグラフィック制限部が含まれる。視覚システム1514にはまたピエゾ電気ジャイロシステム1515も含まれる。このジャイロシステム1515は視覚システム1514の画像安定化システム（図示せず）と関連

10

20

30

40

50

する。ピエゾ電子ジャイロを利用する変形可能プリズム画像安定化システムのような画像安定化の一例が、国際出願番号WO 95/07526（国際公開日1995年3月16日、出願者Criticom Corporation）に開示されている。図15のシステムを用いて、この国際出願番号WO 95/07526に記載されているシステムのようなシステムを実施することができる。

特に、例えば、グラフィックオブジェクトの実世界位置及び／又は姿勢に関する情報が何らかの方法で前もって格納されいるシステムで本実施例を用いてもよい。そのようなデータは、例えば、実世界の物体、位置あるいは領域のような実世界のものを表してもよい。しかし、このグラフィックオブジェクトは、これらの実世界の項目と関連するものであつてもよいし、そうでなくてもよい。次いで、この格納された情報はシステム1514へ与えられる。視覚システム1514の位置及び／又は姿勢に基づいて、また、その撮像デバイスの視野に基づいて、システム1514はその格納されたグラフィックオブジェクトを取り消し、ユーザーによって観察された特定の視界のリアルタイムの画像上の正しい場所にグラフィックオブジェクトを重ね合わせることができる。この撮像デバイスは適切なレンズを持つカメラ（例えばディジタル・カメラ）であつてもよい。10

クロック1501の出力部とユーザー入力部1502はタイムトリガ1506に接続している。タイムトリガ1506の出力部、ユーザー入力部1502、位置感知デバイス1504、及び姿勢感知デバイス1505は、予想/呼出時間 - 低減サブシステム1507の出力部に接続し、従って、位置トリガ1508、姿勢トリガ1509、起動化インターバルトリガ1511、反復アクショントリガ1512、及び反復距離トリガ1513に接続している。予想/呼出時間 - 低減サブシステム1507、クロック1501、ユーザー入力部1502、位置感知デバイス1504、姿勢感知デバイス1505、タイムトリガ1506、及びピエゾ電気ジャイロ1515は視覚システム1514に接続し、従って、ユニットモーションサブシステム1516、表示利用サブシステム1517、及びスリープサブシステム1518に因るグラフィック制限部に接続している。20

図16は、システム1500の概括的オペレーションを示すフローチャート1600である。ステップ1601で、ユーザーはモニターモードを起動し、システム1500に対してタイムトリガ1506と予想/呼出時間 - 低減サブシステム1507をモニターするように命じる。次いでフローチャート1600はステップ1602へ分岐し、そこでシステム1500はモニター限界値Wによって指定された時間についてタイムトリガ1506をモニターし、次いでステップ1603へ分岐する。ステップ1603でシステム1500は予想/呼出時間 - 低減サブシステム1507をモニターする。30

次いでフローチャート1600はステップ1604へ分岐し、そこで視覚システム1514が起動され次いでステップ1605へ分岐する。ステップ1605でシステム1500は、視覚システム1514のユニットモーションサブシステム1516に因るグラフィック制限部をモニターする。フローチャート1600はステップ1605からステップ1606へ分岐し、そこでシステム1500は視覚システム1514の表示利用サブシステム1517をモニターする。次いで、フローチャートはステップ1607へ分岐し、そこで、システム1500は視覚システム1514のスリープサブシステム1518をモニターする。

図17と図18は、ユニットモーションに因るグラフィック制限サブシステム1516のオペレーションを示す。図21は、ユニットモーションサブシステム1516に因るグラフィック制限を実施するために使用されるハードウェアの実施例のプロック図である。図17と図18は、ユニットモーションサブシステム1516に因るグラフィック制限を実施するために使用されるソフトウェアが、ピエゾ電気ジャイロ1515が自記する視覚システム1514の検出された振動に関連してどのように作動するかを示すフローチャート1700と1800を例示するものである。ステップ1701で、システム1516はアプリケーション特有の振動限界値Hを定義する。視覚システム1514は、その振動レベルが限界値H以上に上昇するとすべてのグラフィックの複雑さのレベルを落とし始める。この“振動レベル”は、ある期間にわたるいくつかの動きの変化（動きの方向など）によって通常測定される。従って、この“振動レベル”とは、例えば方向変化率となる場合もある“振動率”である。ステップ1702で、サブシステム1516は、ピエゾ電気ジャイロ1515からのモーション信号と、クロック1501からの時間信号とを受信し、その振動率を計算する。これらのジャイロはまた視覚システム1514の変形可能プリズム画像安定化システム（図示せず）とも通常関連する。もっとも、これらの4050

ジャイロは他のデバイス1514サブシステムから独立したものであってもよいが。ステップ1703でシステム1514はこの計算された振動率が振動限界値Hを超えるかどうかを確かめる。この計算された振動率がHを超えない場合にはフローチャート1700は、ユニットモーションサブシステム1516に因るグラフィック制限部の追加オペレーションを記述する図18のステップ1801へ分岐する。この計算された振動率が振動限界値Hを超える場合にはフローチャート1700はステップ1704へ分岐する。ステップ1704でシステム1514は、この計算された振動率が視覚システムによって表示された画像を安定化する安定化システムの能力を超えるかどうかを確かめる。この画像安定化システムは通常最大振動率を処理することができる仕様となっている。この計算された振動率が画像を安定化する視覚システム安定化システムの能力を超えない場合にはフローチャート1700はステップ1706へ分岐する。この計算された振動率が画像を安定化する視覚システム安定化システムの能力を超える場合にはフローチャート1700はステップ1705へ分岐する。

ステップ1705でシステム1516は、検出された振動の激しさに基づいて、すべての取り消されたグラフィックオブジェクトの“複雑さのレベル”を適当な数の“複雑さのレベル”だけ落とす。例えば、本実施例では、振動がその振動を補償する安定化システムの能力をレベル2以上超えているという判断に応じて複雑さのレベルを落としてもよい。本発明の実施例では、そのような状況下でその振動がもっと激しくなりそうなために安定化システムが振動を補償することがもはやできなくなったとき複雑さのレベルを適宜もっと大きく落としてもよい。

この安定化システムが振動を処理することができる場合であっても、本実施例では、ユーザーが振動しそうであるという理由でこの複雑さのレベルを落とす。

従って、ステップ1706でシステム1514は、表示されているグラフィックオブジェクトのすべての“複雑さのレベル”を1“レベル”だけ落とす。本実施例では、複雑さのレベルを落とすために、一つ以上の複雑さのレベルを定義して各々のグラフィックオブジェクトを表してもよい。従って、一つのグラフィックオブジェクトを一つの複雑さのレベルで表してもよい。一方、第二のグラフィックオブジェクトを複数の複雑さのレベルで表してもよい。ある特定のグラフィックオブジェクトと関連するこの様々な複雑さのレベルは各々視覚的にその特定のグラフィックオブジェクトを表すものではあるが、それを様々なレベルの複雑さで表すものである。これらのレベルは、非常に複雑なもの（例えばフルプローンラスター画像）からグラフィックオブジェクトの意味をユーザーに伝えるのに必要な最小限の複雑さのもの（例えば単純なベクトル画像）までその範囲を設定することができる。グラフィックオブジェクトが関連する実世界の物体の重要度に基づいて重要度を示す数をグラフィックオブジェクトに割り当てることなどによって、任意の特定の瞬間に特定のグラフィックオブジェクトを表すために使われるグラフィックの複雑さを決定してもよい。この重要度を示す数（IN）はアプリケーションにより定義したものであってもよい。例えば、海上航行用アプリケーションでは、航行マーカーと関連するグラフィックオブジェクトは相対的に高い重要度を示す数を持つものとしてもよい。しかし、観光用アプリケーションでは、同じ地理上の領域をカバーしていても、これらの航行マーカーは重要度の低いものになりそうである。従って、航行マーカーと関連するこのグラフィックオブジェクトは、航行用アプリケーションでは高い重要度を示す数を持つことになるかもしれないが、観光用アプリケーションでは低い重要度を示す数を持つことになろう。アプリケーションによって割り当てられるこの重要度を示す数は、アプリケーションが一つのオペレーションモードから別のモードへ切り替わるにつれて変化するものとしてもよい。上記の例を用いて、ある特定の地理上の地域で航行用と観光用の二つのオペレーションモードで作動するようにシステム1514を設計することもできよう。

複雑さのレベルを変更するためにシステム1514がどのグラフィックオブジェクトを表示するかについての制御を行ってもよい。例えば振動が起こったとき、システム1514がより重要なグラフィックオブジェクトだけを表示するようにしてもよい。あるいは、システム1514は、重要度に基づいて表示されるグラフィックオブジェクトのいくらか又はすべての解像度を迅速に落とすことによって複雑さのレベルを落とすこともできる。従って、例えば

10

20

30

40

50

、システム1514を前述の観光用という状況で使用していた場合、振動の結果としてマーカーの解像度が落とされてマーカーが大まかな幾何学的近似としてのみ表示されるようになる場合もある。あるいは、そのような状況でマーカーを全く表示しないこともある。これらの複雑さのレベルを定義するために各々の複雑さのレベルが“複雑さを示す数として”割り当てられる。本実施例では、この複雑さを示す数は、その特定の複雑さのレベルと関連するグラフィックオブジェクトを生成するために必要とされる計算回数である。グラフィック生成用としてシステム1514のリソースを割り当てる際にこれらの様々な複雑さのレベルはシステム1514によって利用される。

図18は、ユニットモーションに因るグラフィック制限サブシステム1516の第二の部分のオペレーションを示すフローチャート1800である。検出された状況に応じてオペレーションによってグラフィックオブジェクトの複雑さのレベルを落とすという意味で、このオペレーションは図17で説明したオペレーションと類似している。しかし、図18のオペレーションは振動ではなくて姿勢変化率を処理するものである。特に、図18のサブシステム1516は視覚システム1514の検出された姿勢値スルーレートに応じて作動する。本実施例では、視覚システム1514のスルーレートとはシステム1514の姿勢変化率を示す。本実施例では、サブシステム1516はソフトウェアを実行するハードウェアを用いて実施される。図25を参照しながらこのハードウェアを説明する。フローチャート1800はこのソフトウェアのオペレーションを表す。フローチャート1800のステップ1801ではサブシステム1516が所定の姿勢値スルーレート限界値Jを定義する。この限界値Jはアプリケーション特有値（すなわちアプリケーションの要求条件に従って決定された特有値）とすることができます。この限界値Jはシステム1516がグラフィックオブジェクトの複雑さのレベルを落とし始めるスルーレートである。10

ステップ1802でシステム1516は姿勢感知デバイス1505からの姿勢信号とクロック1501からのクロック信号とを受信する。システム1516はこれらの信号から視覚システム1514の実際の姿勢値スルーレートKを計算する。ステップ1803でシステム1516はこの計算された姿勢値スルーレートKが姿勢値スルーレート限界値Jを超えるかどうかを確かめる。KがJを超えない場合フローチャート1800は図19のステップ1901へ分岐し、表示利用サブシステム1517をチェックする。KがJを超える場合フローチャート1800はステップ1804へ分岐する。ステップ1804でシステム1514はすべてのグラフィックの複雑さのレベルをレベル1以上落とすが、その量はアプリケーション特有の複雑さレベル低下スルーレート閾値によって定義される。例えば、測定されたスルーレートKが第一の閾値を超える場合には、その第一の閾値の超過と関連してその複雑さをある複雑さのレベルまで落としてもよい。スルーレートKが第二の閾値を超える場合には、その第二の閾値の超過と関連してその複雑さをある複雑さのレベルまで落としてもよい。次いでフローチャート1800は図19のステップ1901へ分岐し表示利用サブシステム1517をチェックする。20

図19はシステム1514の表示利用サブシステム1519のオペレーションを示すフローチャート1900である。このサブシステム1519は、ユーザーが実際に視覚システム1514のディスプレー（図示せず）を見ているかどうかを検出し、適宜ディスプレーを起動したり非起動にしたりする。バックライティングをウォーミングアップするような、このディスプレーと関連する活動のいくつかは全く非起動にされることはないことに留意されたい。その代わりそのような活動は視覚システム1514が全体として完全に非起動になっている間もアクティブの状態のままであってもよい。ステップ1901でサブシステム1519はディスプレーがアクティブかどうかを確かめる。ディスプレーがアクティブであればフローチャートはステップ1902へ分岐する。ディスプレーがアクティブでなければフローチャートはステップ1903へ分岐する。30

ステップ1902でサブシステム1519は物理的物体がシステム1514のアプリケーション/ユーザー定義ディスプレー起動化範囲閾値の範囲内にあるかどうかを確かめる。本実施例ではサブシステム1519は、ユーザーの眼などの接近を検出するように設計されている。低パワー音波/光放射範囲ファインダや他の同様のデバイスを用いてそのような判断を行ってもよい。ユーザーがこの起動化閾値の変更を希望して、眼鏡着用者にもこのシステムを利用40

できるようにしてもよい。眼鏡は、ユーザーの眼よりディスプレーにより近い反射面を与えることによって範囲ファインダによる接近測定に影響を与える場合もある。好適なディスプレー起動化範囲閾値をユーザー使用プロファイルの一部とすることもできよう。このような使用プロファイルを用いて、システム1500及び／又はシステム1514に対してある特定のユーザーと関連する一定の属性について知らせることもできよう。

ステップ1902のシステム1519がこのディスプレー起動化範囲閾値の範囲内である物体を検出した場合には、フローチャート1900はステップ1905へ分岐し、そこでこのディスプレーは起動されたままの状態になる。次いでフローチャート1900は図20のステップ2001へ分岐し、スリープサブシステム1518をチェックする。ステップ1902でこのディスプレー起動化範囲閾値の範囲内で物体が検出されない場合には、フローチャート1900はステップ1902からステップ1904へ分岐する。ステップ1904でディスプレーは非起動となる。次いでフローチャート1900は図20のステップ2001へ分岐しスリープサブシステム1518をチェックする。フローチャート1900がステップ1901からステップ1903へ分岐した場合、ステップ1903でシステム1519は、ある物体がアプリケーション/ユーザー定義ディスプレー起動化範囲の範囲内にあるかどうかを確かめる。このディスプレー起動化範囲閾値の範囲内で、ある物体が検出された場合、フローチャート1900はステップ1903からステップ1906へ分岐しそこでディスプレーが起動される。次いでシステム1519は図20のステップ2001へ分岐しスリープサブシステム1518をチェックする。ステップ1903で物体がこのディスプレー起動化範囲の範囲内で検出されなかった場合、フローチャート1900は図20のステップ2001へ分岐しスリープサブシステム1518をチェックする。10

図20は、スリープサブシステム1518のオペレーションを示すフローチャート2000である。このフローチャート2000は例えばシステム200を例示したフローチャートと関連して読むことができる。ユーザーまたはアプリケーション定義期間にわたって視覚システム1514の位置または姿勢値が変化しない場合には、このサブシステム1518はコネクタ5を介してシステム1500をモニターモードへ戻す。このサブシステム1518は、本発明の一つの実施例によって制御されるデバイスの電力消費量をさらに減らすことのできる追加テクニックを例示する。20

スリープサブシステム1518は、ハードウェアを用いてソフトウェアを実行することによって実施される。図20のフローチャート2000によってこのソフトウェアのオペレーションは例示されている。スリープサブシステムを実施するためのハードウェアは当業で公知のいずれの方法でも設計することができる。30

図20に示すように、ステップ201でスリープサブシステム1518は、ユーザーがすでにスリープルーチンを起動しているかどうかを判定する。起動していないければサブシステム1518はステップ2014へ分岐する。起動しているれば、サブシステム1518はステップ2002へ分岐する。ステップ2002でサブシステム1518は定常状態での時間限界値Mを定義する。この時間限界値Mとは、システム1514をスリープ状態にすべきかどうかを判定するためにサブシステム1518が使う時間を指す。例えば、前回の時間Mの範囲内でシステム1514の姿勢と位置の変化がなかった場合には、サブシステム1518はシステム1514をスリープ状態にする。

サブシステム1518はステップ2002からステップ2003へ分岐する。ステップ2003でサブシステム1518は、スリープ状態起動化インターバルがユーザーまたは何らかの他のデバイスなどによって特定されていたかどうかを判定するためのテストを行う。このスリープ状態起動化インターバルは、時間限界値Mと同じ方法でサブシステム1518によって使用される。例えば、前回の時間Lの範囲内でシステム1514の姿勢と位置の変化がなかった場合、サブシステム1518はシステム1514をスリープ状態にする。MとLとの違いは、Mがサブシステム1518自身によって指定されているのに対して、Lはユーザーによって指定されているという点である。40

スリープ状態起動化インターバルLが特定されていた場合サブシステム1518はステップ2006へ分岐する。スリープ状態起動化インターバルが特定されていなかった場合にはサブシステム1518はステップ2004へ分岐する。

ステップ2006でサブシステム1518は、システムがLからゼロへカウントダウンし始めたか50

どうかを判定するテストを行う。このカウントダウンはデバイス1514の位置と姿勢のいずれも変化していないときはいつでも始まる。サブシステム1518がこのカウントダウンをすでに開始していた場合サブシステム1518はステップ2007へ分岐する。サブシステム1518がまだカウントダウンを開始していなかった場合サブシステム1518はステップ2009へ分岐する。

ステップ2009で、サブシステム1518は、視覚システム1514の位置又は姿勢が最近のM秒で変化したかどうかを判定するテストを行う。次いで位置と姿勢の少なくとも一方が変化していた場合サブシステム1518はステップ2014へ分岐する。位置と姿勢の一方が変化していなかった場合にはサブシステム1518はステップ2010へ分岐する。ステップ2010でサブシステム1518はLからゼロへカウントダウンし、次いでステップ2014へ分岐する。

サブシステム1518がステップ2004へ分岐した場合、システム1500はスリープ状態起動化インターバルLを定義するようにユーザーに促す。次いでサブシステム1518はステップ2005へ分岐する。ステップ2005でユーザーはシステム1500に対してタイムインターバルLを入力する。システム1500はこのタイムインターバルをサブシステム1518へ通信し、サブシステム1518はこのタイムインターバルをLとして格納する。次いでサブシステム1518はステップ2014へ分岐する。

サブシステム1518がステップ2007へ分岐した場合ステップ2007は、視覚システム1514の位置又は姿勢が最近のM秒で変化したかどうかを判定するテストを行う。位置と姿勢の一方が変化していた場合にはサブシステム1518はステップ2008へ分岐する。ステップ2008でサブシステム1518はLからゼロへのカウントダウンを停止し、カウントダウンを行っていたカウンターをLへリセットする。サブシステム1518はステップ2008からステップ2014へ分岐する。ステップ2007で位置と姿勢の一方が変化していないとサブシステム1518が判定した場合、サブシステム1518はステップ2011へ分岐する。ステップ2011でサブシステム1518はカウントダウンがゼロに等しいかどうかを判定するテストを行う。カウントダウンがゼロに等しくない場合サブシステム1518はステップ2014へ分岐する。カウントダウンがゼロに等しい場合サブシステム1518はステップ2011からステップ2012へ分岐する。ステップ2012でサブシステム1518は、（例えばユーザーや何らかの他のデバイスによって）デバイス1514が非起動になっているかどうかを判定するテストを行う。デバイス1514が非起動になつていればサブシステム1518はステップ2015へ分岐し、そこでシステム1500のスイッチをオフにする。デバイス1514が非起動になつていなければサブシステム1518はステップ2012からステップ2013へ分岐する。

サブシステム1518がステップ2014へ分岐した場合ステップ2014でサブシステム1518は、ユーザーや何らかの他のデバイスが手動または他のデバイスのアクションのいずれかによってシステム1514をスリープ状態にしたかどうかを判定するテストを行う。システム1514がスリープ状態にされている場合、サブシステム1518はステップ2013へ分岐する。ステップ2013でシステム1500はディスプレーを非起動にし図5のステップ502へ戻る。システム1514がスリープ状態にされていなかった場合サブシステム1518はステップ2016へ分岐する。ステップ2016でサブシステム1518はシステム1514が非起動になっているかどうかを判定するテストを行う。システム1514が非起動になっている場合システム1500はスイッチをオフにする。システム1514が非起動になつていなければサブシステム1518は図16のステップ1604へ分岐する。

タイムトリガ106と206を実施するハードウェアは図12に例示されている。タイムトリガ106と206には四つのRISCプロセッサ1202、1204、1206及び1208が含まれる。これらのプロセッサは図5に関して説明したオペレーションの一部をそれぞれ行うようにプログラムされている。RISCプロセッサ1202はモニター限界値Wを格納し、経過時間読み取り値ZをWとを比較する。プロセッサ1204は経過時間読み取り値Zを追跡する。プロセッサ1208は“最後にチェックされた”時刻Yを格納し、時刻Yから現在の時刻までの経過時間を計算する。プロセッサ1206は起動化インターバルXを格納し、起動化インターバルXを時刻Yから現在の時刻までの経過時間と比較する。マルチプロセッサを使用する必要はない。特に、代替実施例では、本発明の一つの実施例を実施するために適切な機能を果たすものであれば、様

10

20

30

40

50

々な数のプロセッサ、単一のRISC又はCISCプロセッサあるいは他のタイプのハードウェア及び／又はそれらの組合せでも利用することができる。

姿勢トリガ109と209を実施するハードウェアは図13に例示されている。姿勢トリガ109と209には5つのRISCプロセッサ1302、1304、1306、1308及び1310が含まれる。これらのプロセッサは図10に関して説明したオペレーションの一部を行なうようにプログラムされている。プロセッサ1308は“最後に受信した”姿勢値Bを格納し、姿勢値Bが格納された時刻から新しい姿勢値が格納される時刻までの姿勢変化の角度/秒レートを計算する。プロセッサ1302は“角度/秒”起動化限界値Cを格納し、この限界値Cをプロセッサ1308によって計算された角度/秒値と比較する。プロセッサ1304は定常状態姿勢読み取り値Aを格納し、最後に受信した姿勢値とAとの差を計算する。プロセッサ1306は“新しい定常状態Aの記録”限界値Eを格納し、プロセッサ1308が計算した角度/秒値を限界値Eと比較する。プロセッサ1310は“定常状態からの角度”起動化限界値Eを格納し、この限界値Eをプロセッサ1304が計算した差と比較する。マルチプロセッサを使用する必要はない。特に、代替実施例では、本発明の一つの実施例を実施するために適切な機能を果たすものであれば、様々な数のプロセッサ、単一のRISC又はCISCプロセッサあるいは他のタイプのハードウェア及び／又はそれらの組合せでも利用することができる。10

位置トリガ108と208を実行するハードウェアは図14に例示されている。位置トリガ108と208には三つのRISCプロセッサ1402と1404グラフィックコントローラ1406が含まれる。これらのプロセッサは、それぞれ図11に関して説明したオペレーションの一部を行なうようにプログラムされている。グラフィックコントローラはまた、図11に関して説明した機能の中のいくつかを行う。プロセッサ1402は“最後のストップ”位置Gを格納し、現在の位置から最後のストップ位置Gまでの範囲を計算する。プロセッサ1404はその設定距離Fを格納し、プロセッサ1402によって計算された範囲を距離Fと比較する。グラフィックコントローラ1406は、ユーザーが関心を持つものとして指定した全ての領域、並びに、範囲起動化と関連したすべての領域に対する範囲を計算する。マルチプロセッサを使用する必要はない。特に、代替実施例では、本発明の一つの実施例を実施するために適切な機能を果たすものであれば、様々な数のプロセッサ、単一のRISC又はCISCプロセッサあるいは他のタイプのハードウェア及び／又はそれらの組合せでも利用することができる。20

図21は、ユニットモーションによるグラフィック制限サブシステム1516を実施するために使用されるこのハードウェアを示す一つの推奨実施例のブロック図である。サブシステム1516には三つのRISCプロセッサ2102、2104及び2106が含まれる。RISCプロセッサの各々は、図17と図18に関して説明したオペレーションの一部を行なうようにプログラムされている。例えば、RISCプロセッサ2102は振動限界値Hを格納し、システム1514の現在の測定振動値をHと比較するようにプログラムされている。RISCプロセッサ2104は、安定化システムの振動限界値を格納し、サブシステム1514の現在の測定振動値をシステム1514の安定化サブシステムの振動限界値と比較するようにプログラムされている。RISCプロセッサ2106はスルーレート限界値Jを格納し、システム1514の測定スルーレートKとJを比較する。30

図22は図2のAIT211のハードウェアを例示する。図に示すように、AIT211は7個のRISCプロセッサ2202、2204、2206、2208、2210、2212、及び2214を使用する。RISCプロセッサ2202はAITインターバル値を格納する。プロセッサ2204はシステム200のモニターモードが起動された時刻を格納する。この時刻は図5のステップ502で最初に読み取られた時刻である。プロセッサ2206は、プロセッサ2204によって格納された時刻から現在の時刻までの経過時間を計算する。ただし現在時刻とは、ステップ502でシステム200によって最も最近に読み取られた時刻である。プロセッサ2208は、プロセッサ2206によって計算された経過時間(DTI)をプロセッサ2202に格納されたAITインターバル値と比較する。プロセッサ2210はユーザーがすでにデバイス214を起動しているかどうかを判定する。起動していれば、プロセッサ2206で計算された経過時間がダウンタイムインターバル(DTI)となる。プロセッサ2210はDTIのリストにこのDTIを格納する。このリストがいっぱいであれば最も古いDTIがリストから外される。プロセッサ2212は、プロセッサ2210によって作成されたリスト中のDTIの各々をこのリスト中のその他のDTIの各々と比較し、リスト中の大多数のDTIがお40

互いの所定の許容値の範囲内にあるかどうかを判定する。プロセッサ2214はお互いの所定の許容値の範囲内にあるDTIがもしあればそれを平均する。この平均値は、AITインターバル値の新しい値としてプロセッサ2202によって格納される。マルチプロセッサを使用する必要はない。代替実施例では、本発明の一つの実施例を実施するために適切な機能を果たすものであれば、様々な数のプロセッサ、単一のRISC又はCISCプロセッサあるいは他のタイプのハードウェア及び／又はそれらの組合せでも利用することができる。

図23は図2のRAT212のハードウェアを例示する。図に示すように、RAT212は6個のRISCプロセッサ2302、2304、2306、2308、2310、及び2312を使用する。RISCプロセッサ2302は姿勢読み取り値のリストにデバイス214の現在の姿勢値を入れる。リストがいっぱいの場合には最も古い姿勢読み取り値がリストから外される。プロセッサ2304はその反復アクショントリガ設定値を格納する。プロセッサ2306は、プロセッサ2302が生成したリストの各々の値をプロセッサ2304が格納したRAT設定値と比較する。プロセッサ2308は、ユーザーが電子デバイス214を起動しているかどうかを判定する。起動されていれば、プロセッサ2302が生成した姿勢読み取り値のリストはモーションルーチンのリストの最上部に格納される。

モーションルーチンのリストがいっぱいの場合には最も古いモーションルーチンがモーションルーチンのリストから外される。プロセッサ2310はこのリストのモーションルーチンの各々をお互いに比較し、大多数のモーションルーチンがリスト中の任意の他のモーションルーチンの所定の許容値の範囲内にあるかどうかを判定する。プロセッサ2312はお互いの所定の許容値の範囲内にあるモーションルーチンを平均する。プロセッサ2304はRAT設定値の新しい値としてこの平均値を格納する。マルチプロセッサを使用する必要はない。代替実施例では、本発明の一つの実施例を実施するために適切な機能を果たすものであれば、様々な数のプロセッサ、単一のRISC又はCISCプロセッサあるいは他のタイプのハードウェア及び／又はそれらの組合せでも利用することができる。特に、代替実施例では、本発明の一つの実施例を実施するために適切な機能を果たすものであれば、様々な数のプロセッサ、単一のRISC又はCISCプロセッサあるいは他のタイプのハードウェア及び／又はそれらの組合せでも利用することができる。

図24は図2のRDT213のハードウェアを例示する。図に示すように、RDT213は7個のRISCプロセッサ2402、2404、2406、2408、2410、2412及び2414を使用する。

RISCプロセッサ2402はRDT反復距離値を格納する。システム200のモニター モードが最初に起動されたときプロセッサ2404はシステム200の位置を格納する。この位置は、図11に例示されているように位置トリガオペレーションの中を通る第一のバスの間最初に読み取られた位置である。プロセッサ2406は、現在の位置とプロセッサ2404によって格納された位置との間の距離を計算する。プロセッサ2408は、プロセッサ2406が計算した距離をプロセッサ2402が格納したRDT反復距離値と比較する。プロセッサ2410はユーザーがデバイス214を起動したかどうかを判定する。起動されていれば、プロセッサ2406で計算された距離がダウンディスタンスインターバル (DDI) となる。プロセッサ2410はこのDDIをDDIのリストに格納する。リストがいっぱいの場合には最も古いDDIがリストから外される。プロセッサ2412は、プロセッサ2410がつくったリスト中のDDIの各々をこのリスト中の他のDDIの各々と比較し、このリスト中の大多数のDDIがお互いの所定の許容値の範囲内にあるかどうかを判定する。プロセッサ2214はお互いの所定の許容値の範囲内にDDIがあればそれらを平均する。この平均値はプロセッサ2402によってRDT反復距離値を表す新しい値として格納される。マルチプロセッサを使用する必要はない。代替実施例では、本発明の一つの実施例を実施するために適切な機能を果たすものであれば、様々な数のプロセッサ、単一のRISC又はCISCプロセッサあるいは他のタイプのハードウェア及び／又はそれらの組合せでも利用することができる。

図25はユニットモーションによるグラフィック制限 (GLDUM) サブシステム1516のハードウェアを例示する。図に示すように、GLDUMサブシステム1516は8個のRISCプロセッサ2502、2504、2506、2508、2510、2512、2514及び2516を使用する。RISCプロセッサ2502～2508は電子デバイス214の振動を処理する。RISCプロセッサ2510～2516は電子デバイス214

10

20

30

40

50

の姿勢変化を処理する。RISCプロセッサ2502は振動限界値Hを格納する。プロセッサ2504は視覚システム1514の振動率を計算する。プロセッサ2506は、プロセッサ2504が計算した振動率をプロセッサ2506が格納した振動限界値と比較する。プロセッサ2508は、プロセッサ2506による比較の結果に従って、表示されたグラフィックオブジェクトの複雑さのレベルを落とすようにシステム1514に指示する。プロセッサ2510はシステム1514の姿勢値スルーレート限界値Jを格納する。プロセッサ2512はシステム1514の姿勢値の実際のスルーレートを計算する。プロセッサ2514は、プロセッサ2514が計算したこのスルーレートをプロセッサ2510が格納したスルーレート限界値Jと比較する。マルチプロセッサを使用する必要はない。代替実施例では、本発明の一つの実施例を実施するために適切な機能を果たすものであれば、様々な数のプロセッサ、単一のRISC又はCISCプロセッサあるいは他のタイプのハードウェア及び／又はそれらの組合せでも利用することができる。図26は表示利用サブシステム1517を例示する。図に示すように、表示利用サブシステム1517は、範囲ファインダ2602、3個のRISCプロセッサ2604、2606及び2608並びにディスプレー平面2610を使用する。範囲ファインダ2602は、デバイス1514の平面ディスプレー2610に接近する物体に対する範囲を検出するようになっている赤外線範囲ファインダであってもよい。本実施例では、この範囲はこのディスプレーから離れる方向及びこのディスプレーに垂直な方向でテストされる。平面ディスプレー2610は、ユーザーに対して画像を提示するビデオモニターやヘッドアップ・ディスプレイのようなディスプレイに対応する。RISCプロセッサ2604はこのディスプレー起動化範囲閾値を格納する。プロセッサ2606は、赤外範囲ファインダ2602から得た範囲データを、プロセッサ2604が格納したこのディスプレー起動化閾値と比較する。プロセッサ2606によるこの比較によってある物体がこのディスプレー起動化範囲閾値の範囲内にあることが示された場合、プロセッサ2608はディスプレーを起動する。あるいはすでにディスプレーが起動されていればディスプレーは起動された状態のままになる。プロセッサ2606によるこの比較によって物体がこのディスプレー起動化範囲閾値の範囲内にないことが示されれば、プロセッサ2608はディスプレーを非起動にする。あるいはすでに非起動であればディスプレーは非起動の状態のままになる。マルチプロセッサを使用する必要はない。代替実施例では、本発明の一つの実施例を実施するために適切な機能を果たすものであれば、様々な数のプロセッサ、単一のRISC又はCISCプロセッサあるいは他のタイプのハードウェア及び／又はそれらの組合せでも利用することができる。

出願者は本発明を具体的な実施例という点から説明してきたが、本発明はこれらの開示された実施例によって限定されるものではない。出願者の発明は、本明細書中で例として述べられた特定のシステムの範囲を越えて適用してもよい。様々な回路を本明細書で説明したが、本発明の実施例において本明細書で説明した特定の回路のすべてを使用する必要はない。さらに、代替実施例では、これらの回路のいくつかあるいはすべては代替の回路を使用することもできよう。例えば、図12～14のRISCプロセッサを、説明した諸機能を果たすCISCプロセッサ、単一のRISCプロセッサまたは異なる回路構成によって置き換えることもできよう。さらに、これらの実施例の一部をソフトウェアとして開示したが、代替実施例ではソフトウェア機能のいくつか又はすべてをハードウェアで実行することもできよう。本発明の代替実施例ではまた、代替の感知方法乃至デバイスを利用して本明細書に開示された目的を達成することもできよう。上限（または下限）限界値として本明細書で表現された限界値及び／又は閾値も、代替実施例では下限（又は上限）限界値として実施してもよい。これらの実施例では電子デバイスの起動化について論じた部分を、代替実施例では、同様の方法で電子デバイスの非起動化のために利用してもよい。同様に、本実施例のフローチャートは“デバイスの起動化”という観点からのものであるが、代替実施例では、デバイスの一部を起動化（または非起動化）して、例えば迅速な起動化あるいは迅速な非起動化が行えるように設計してもよい。

10

20

30

40

【図1】

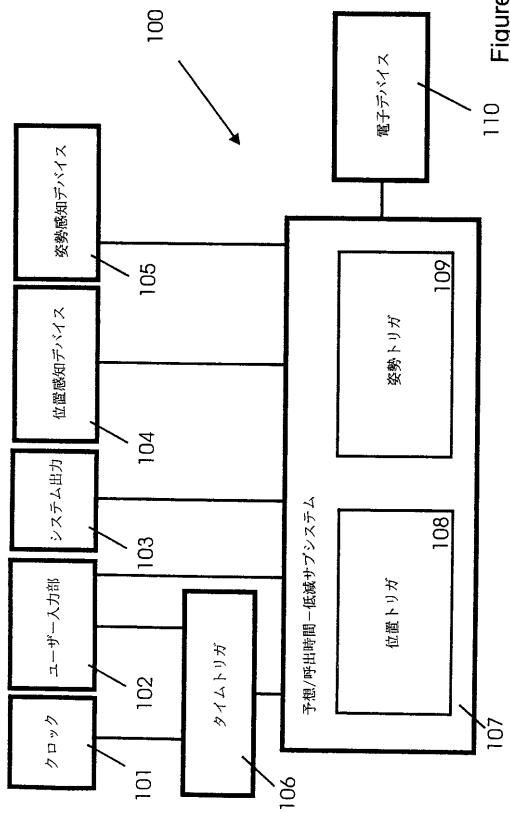


Figure 1

【図2】

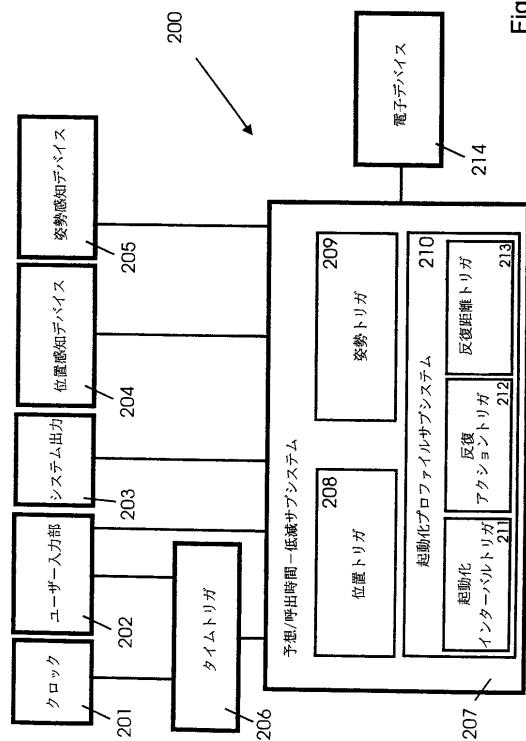


Figure 2

【図3】

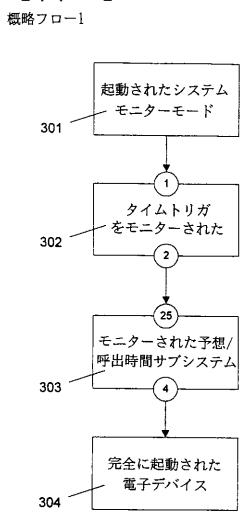


Figure 3

【図4】

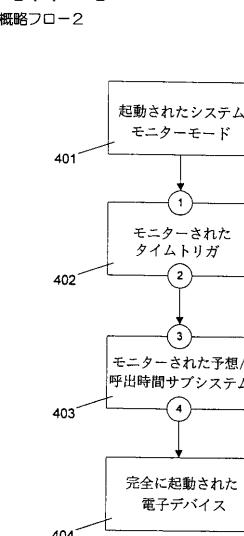


Figure 4

【図5】

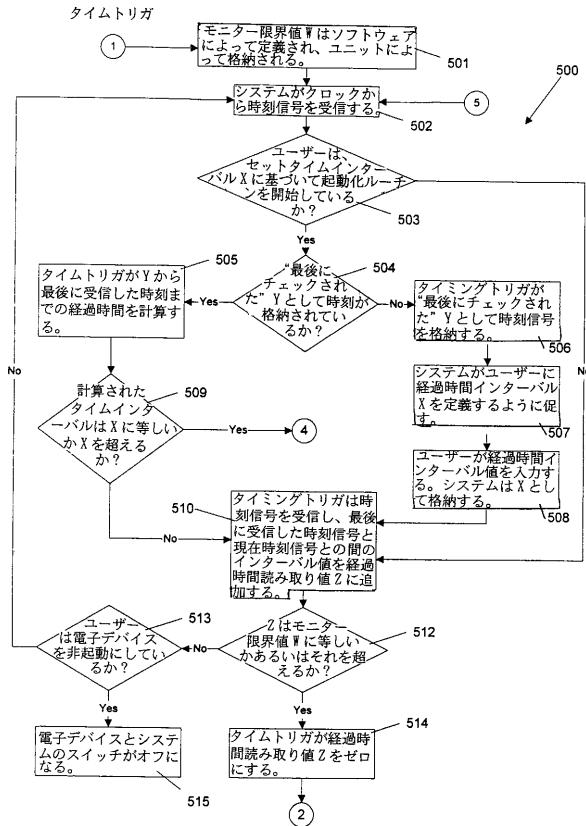


Figure 5

【図6】

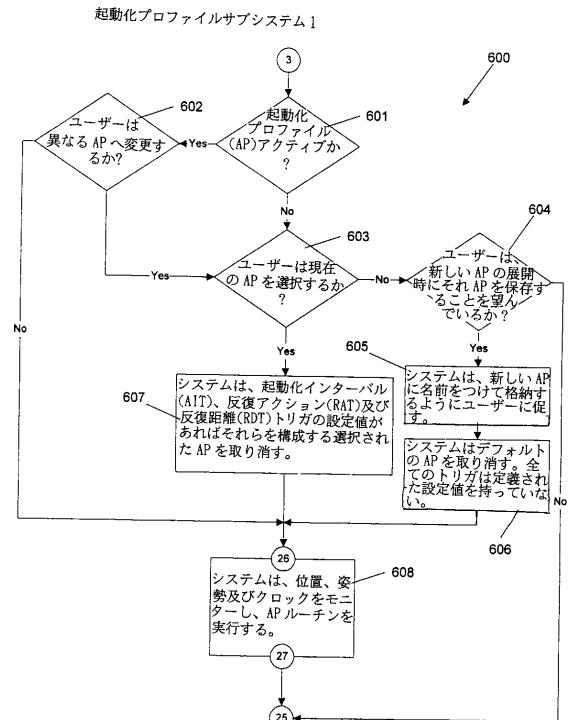


Figure 6

【図7】

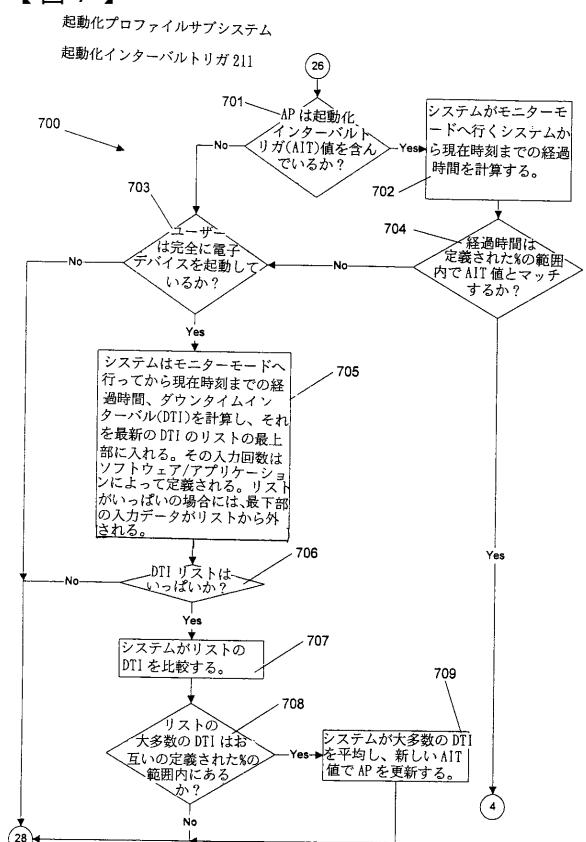


Figure 7

【図8】

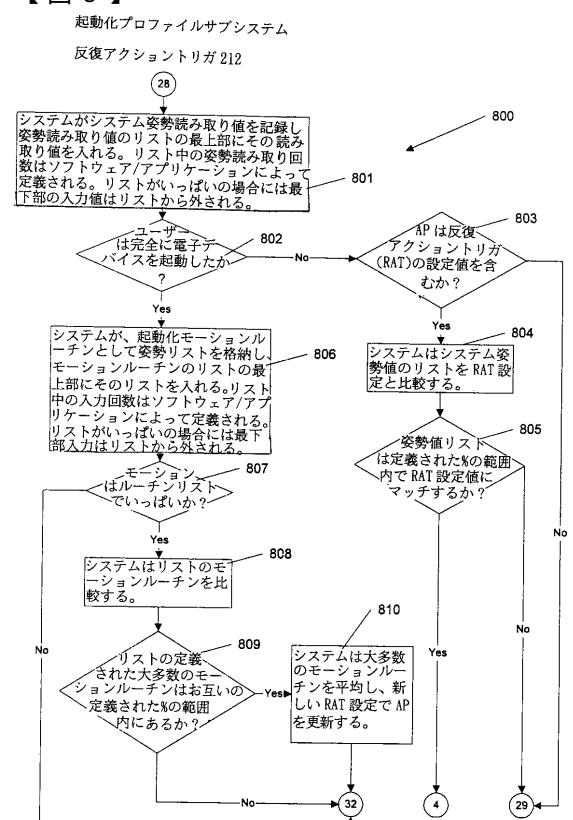
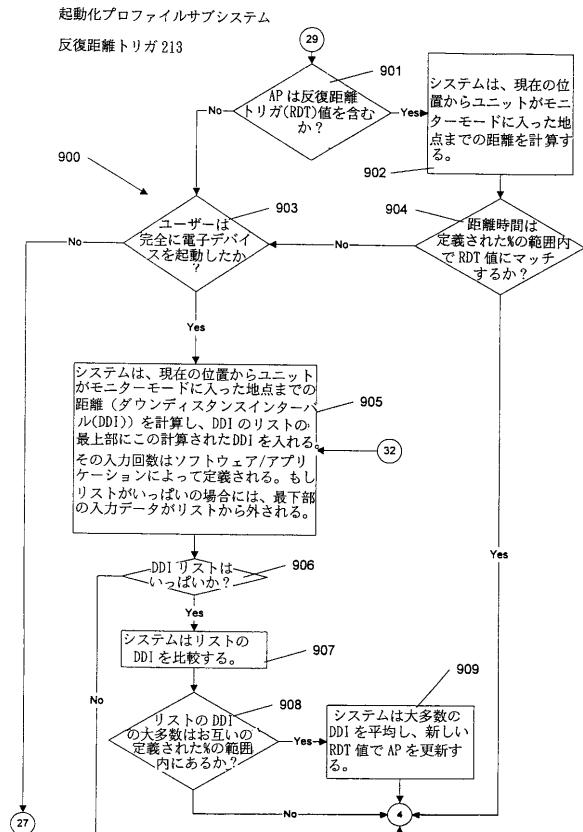
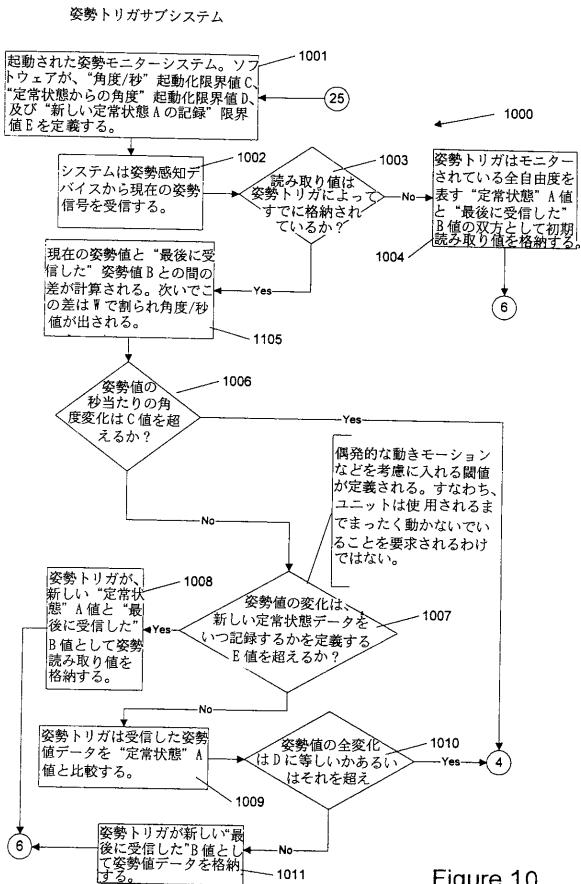


Figure 8

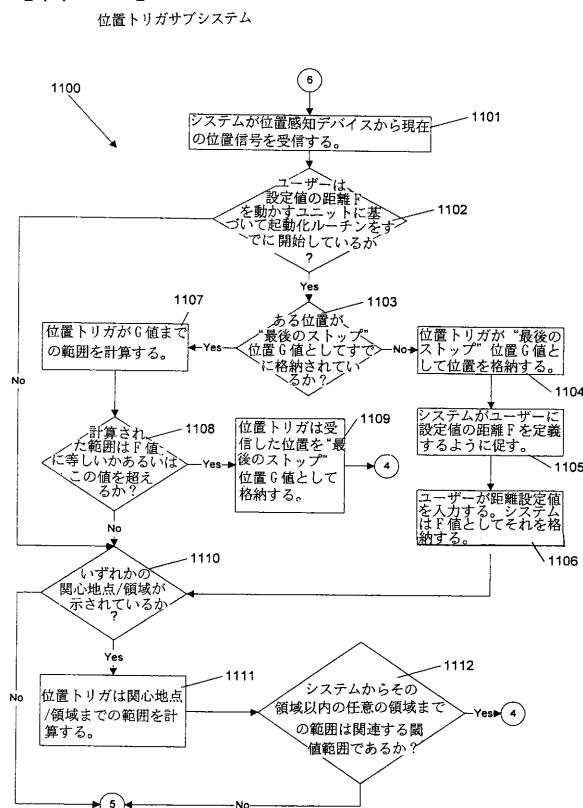
【図9】



【図10】



【図11】



【図12】

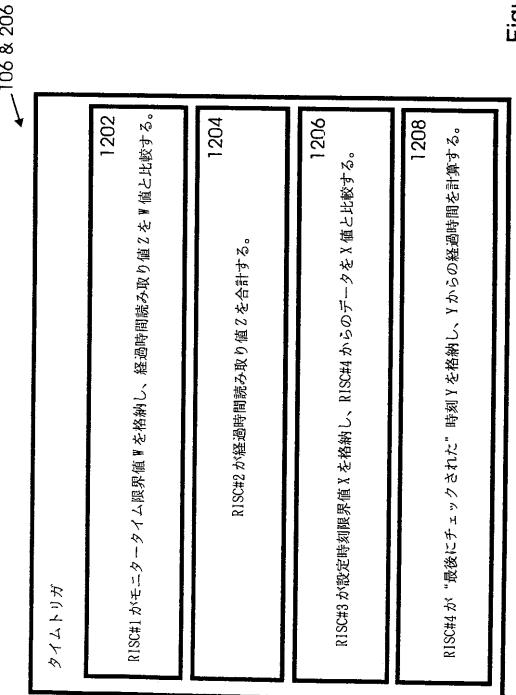


Figure 12

【図 1 3】

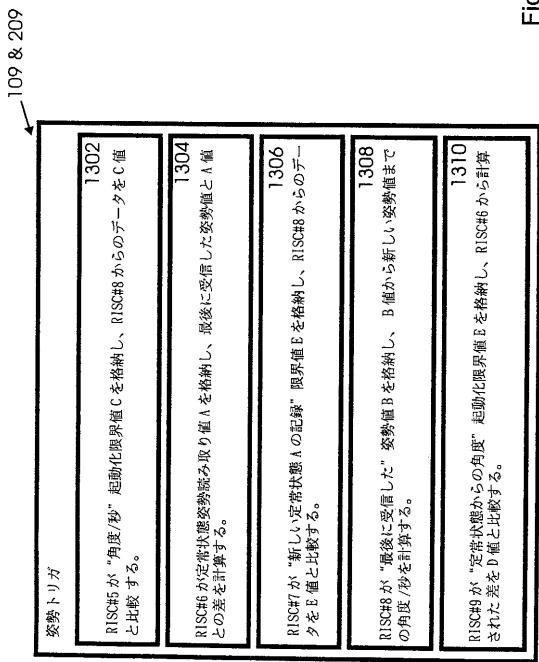


Figure 13

【図 1 4】

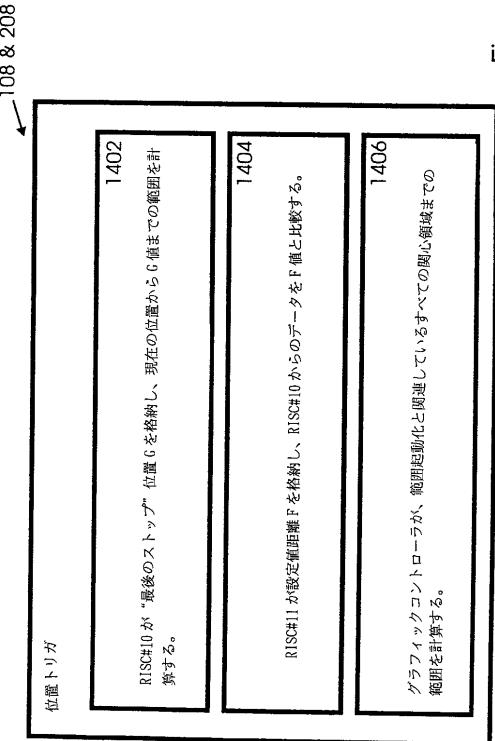


Figure 14

【図 1 5】

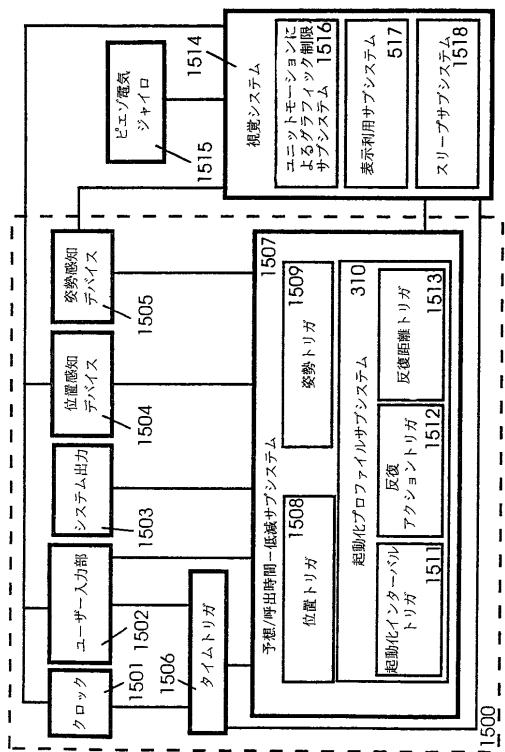


Figure 15

【図 1 6】

経路フロー3

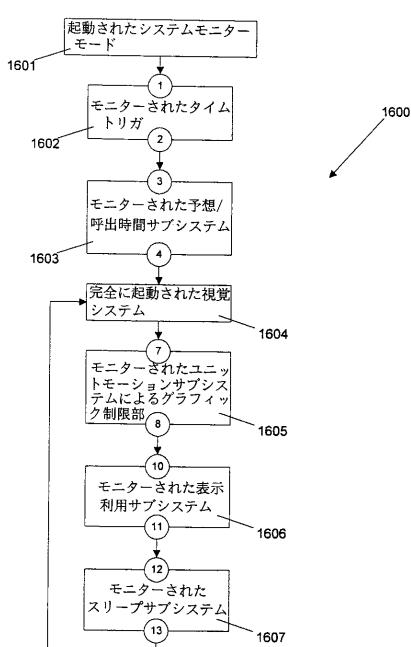
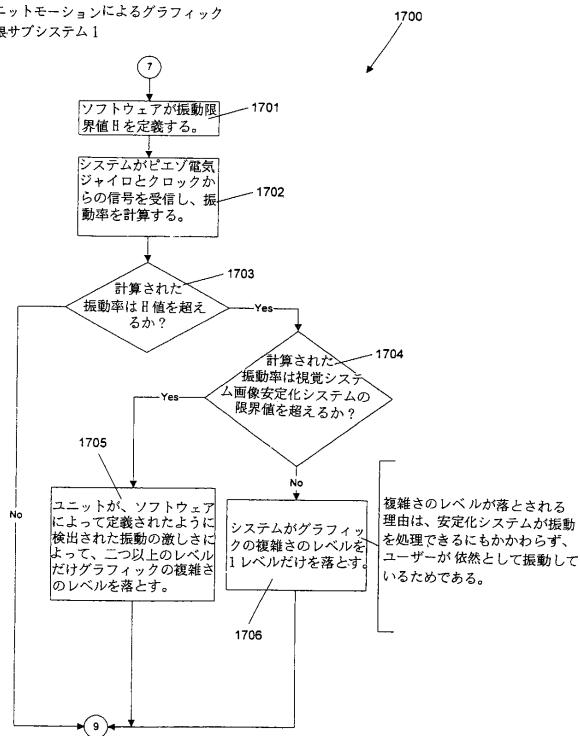


Figure 16

【図17】
ユニットモーションによるグラフィック制限サブシステム1



【図18】
ユニットモーションによるグラフィック制限サブシステム2

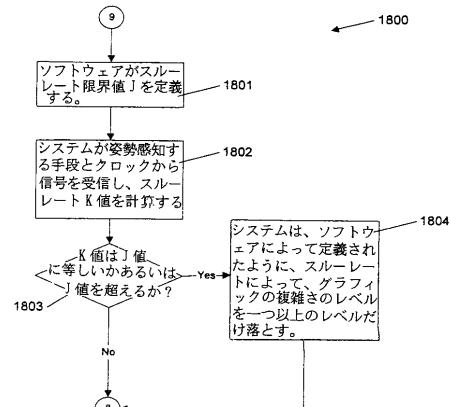


Figure 17

Figure 18

【図19】
表示利用サブシステム

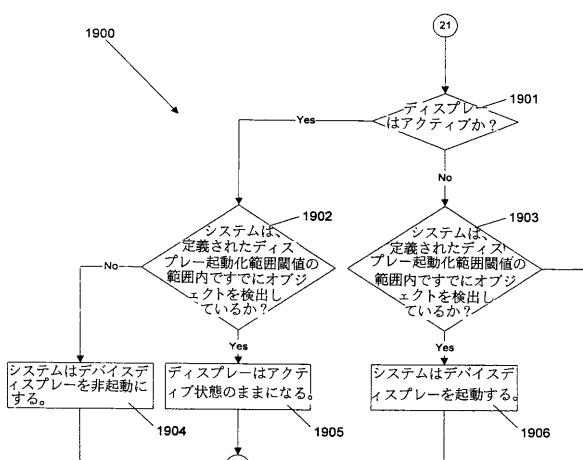


Figure 19

【図20】

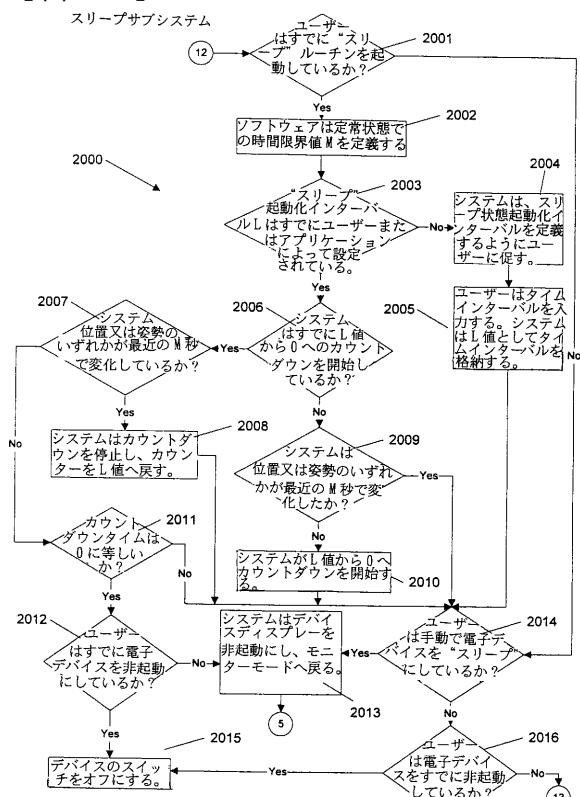


Figure 20

【図 2 1】

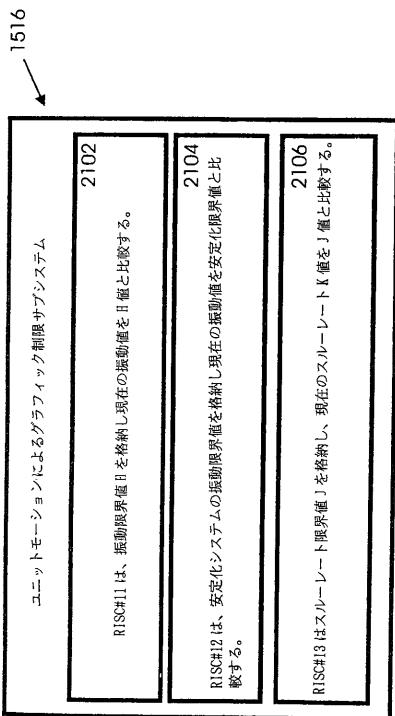


Figure 21

【図 2 2】

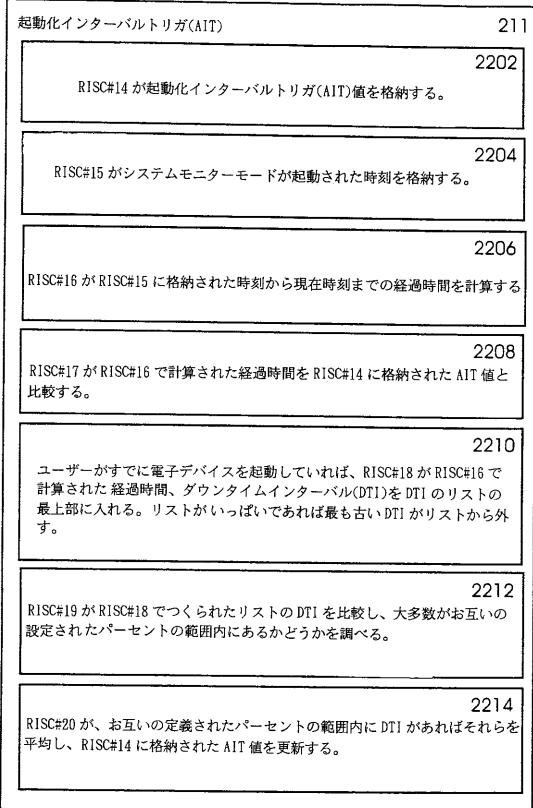


Figure 22

【図 2 3】

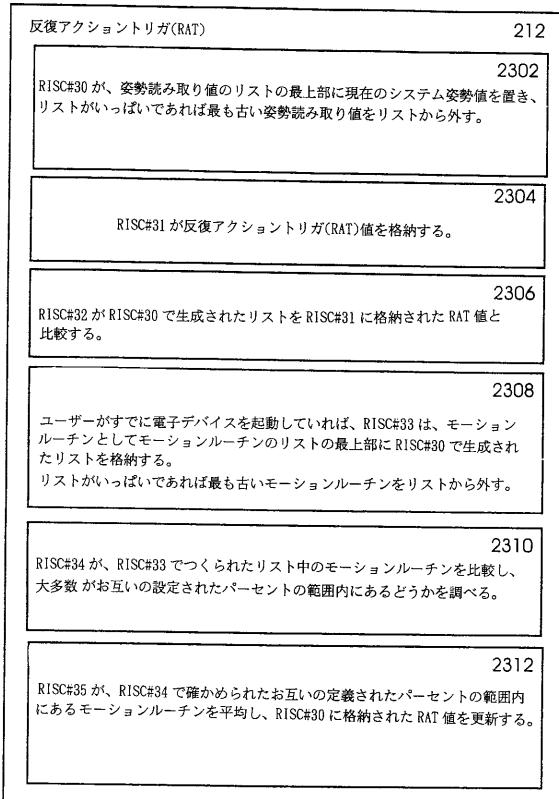


Figure 23

【図 2 4】

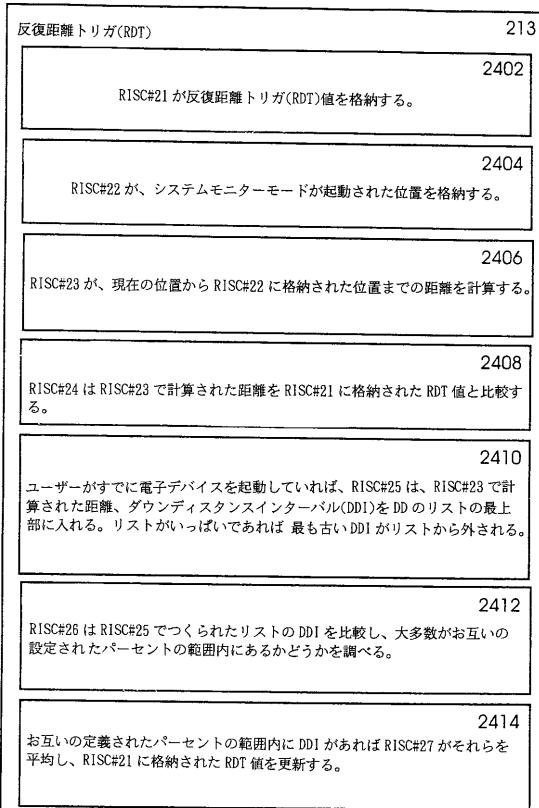
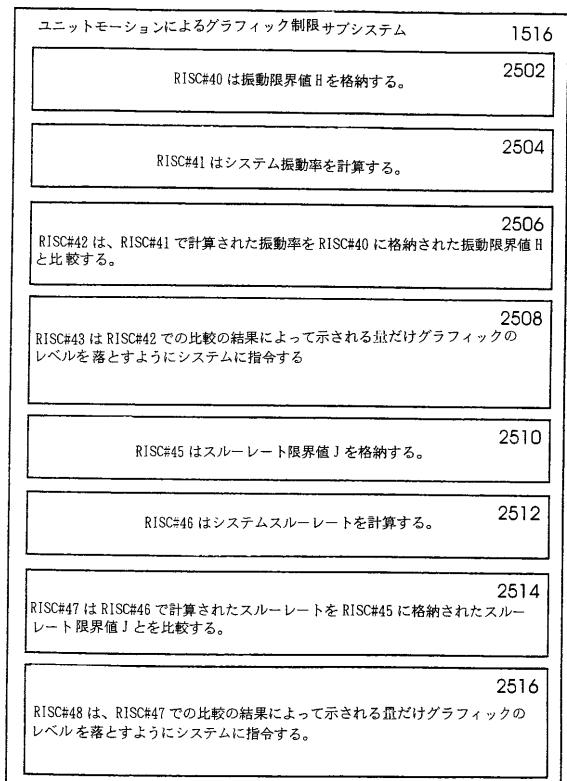


Figure 24

【図 25】



【図 26】

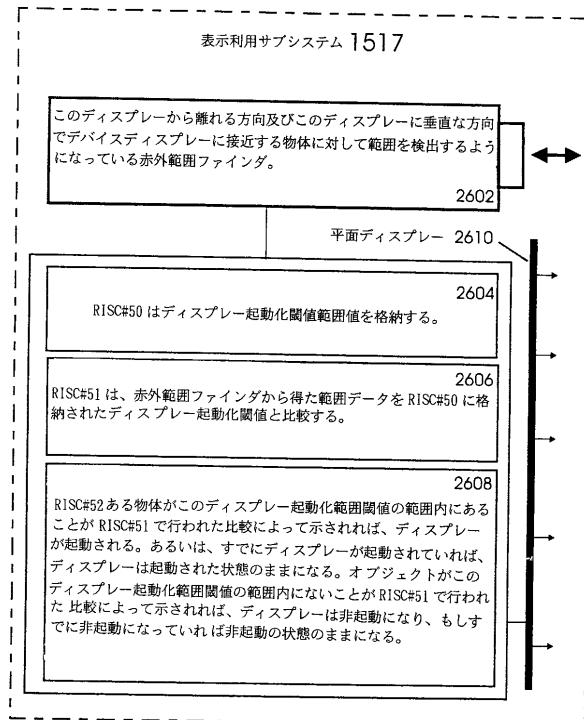


Figure 26

Figure 25

フロントページの続き

(74)代理人 100075672
弁理士 峰 隆司

(74)代理人 100095441
弁理士 白根 俊郎

(74)代理人 100084618
弁理士 村松 貞男

(74)代理人 100103034
弁理士 野河 信久

(74)代理人 100119976
弁理士 幸長 保次郎

(74)代理人 100153051
弁理士 河野 直樹

(74)代理人 100140176
弁理士 砂川 克

(74)代理人 100100952
弁理士 風間 鉄也

(74)代理人 100101812
弁理士 勝村 紘

(74)代理人 100070437
弁理士 河井 将次

(74)代理人 100124394
弁理士 佐藤 立志

(74)代理人 100112807
弁理士 岡田 貴志

(74)代理人 100111073
弁理士 堀内 美保子

(74)代理人 100134290
弁理士 竹内 将訓

(74)代理人 100127144
弁理士 市原 卓三

(74)代理人 100141933
弁理士 山下 元

(72)発明者 エレンピー ジョン
アメリカ合衆国 カリフォルニア州 94107 サン フランシスコ ミネソタ ストリート
601 ユニット 212

(72)発明者 エレンピー ペーター エム
アメリカ合衆国 カリフォルニア州 94107 サン フランシスコ ミネソタ ストリート
601 ユニット 109

(72)発明者 エレンピー トーマス ダブリュー
アメリカ合衆国 カリフォルニア州 94107 サン フランシスコ トウェンティフォース
ストリート #320 2250

合議体

審判長 江嶋 清仁

審判官 近藤 聰

審判官 青木 健

特開平5 - 35355 (JP, A)
特開平7 - 156721 (JP, A)
特開平5 - 244074 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 1/00