

(19) 日本国特許庁 (JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2011-170857

(P2011-170857A)

(43) 公開日 平成23年9月1日(2011.9.1)

(51) Int.Cl.		F I			テーマコード (参考)
G06F 3/01	(2006.01)	G06F 3/01	310A		5B050
G06T 19/00	(2011.01)	G06T 17/40	D		
A63B 69/00	(2006.01)	A63B 69/00	A		

審査請求 未請求 請求項の数 20 O L (全 45 頁)

(21) 出願番号	特願2011-32475 (P2011-32475)	(71) 出願人	510113531
(22) 出願日	平成23年2月17日 (2011.2.17)		アイライブ インコーポレイテッド
(31) 優先権主張番号	12/709, 523		アメリカ合衆国, カリフォルニア州 94
(32) 優先日	平成22年2月22日 (2010.2.22)		041, マウンテン ビュー, 1200
(33) 優先権主張国	米国 (US)		ヴィラ ストリート
(31) 優先権主張番号	12/875, 146	(74) 代理人	110001139
(32) 優先日	平成22年9月3日 (2010.9.3)		S K 特許業務法人
(33) 優先権主張国	米国 (US)	(74) 代理人	100130328
			弁理士 奥野 彰彦
		(74) 代理人	100130672
			弁理士 伊藤 寛之
		(72) 発明者	シャオユエン トウ
			アメリカ合衆国, カリフォルニア州 94
			087, サニーヴェイル, 1162 クラ
			ンダノ コート

最終頁に続く

(54) 【発明の名称】 最小のディレイでモーション認識を行うシステム及び方法

(57) 【要約】 (修正有)

【課題】最小のディレイでモーション認識を行う技術を提供する。

【解決手段】処理ユニットが、少なくとも一つモーション検出デバイスからモーション信号を受け取るために備えられる。このモーション信号は、ユーザーによってなされたモーションを表す。処理ユニットは、モーションレコグナイザーに含まれるプロトタイプの集合にアクセスして、そのモーションレコグナイザー中のプロトタイプの1又は複数を全く考慮することなく、そのモーションレコグナイザーに応答して前記モーション信号から対応する認識信号を生成するように構成されている。バーチャルインタラクティブ環境中の物体の少なくとも一つの動きは、認識信号に応答する。物体の一つを制御するための、前記モーションからのフィードバックは、どれほど多くのモーション信号が受信された場合でも即座であり、実質的に正確である。

【選択図】 図1A

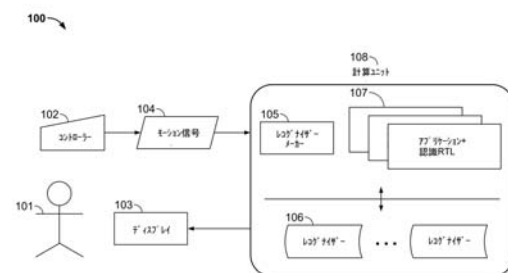


FIG. 1A

【特許請求の範囲】**【請求項 1】**

バーチャルインタラクティブ環境で物体を制御するシステムであって、
少なくとも一つモーション検出デバイスからのモーション信号を受け取り、モーションレ
コグナイザーに含まれるプロトタイプの集合にアクセスして、そのモーションレコグナイ
ザー中のプロトタイプの 1 又は複数を全く考慮することなく、そのモーションレコグナイ
ザーに応答して前記モーション信号から対応する認識信号を生成するように構成されてい
る処理ユニットを備え、前記モーション検出デバイスは、ユーザーによってなされたモー
ションに応答するモーション信号を送り、前記認識信号は、モーション検出デバイスを用
いてユーザーがモーションを行っている間に連続的に生成され、

10

前記処理ユニットは、ディスプレイにバーチャルインタラクティブ環境を表示させ、バー
チャルインタラクティブ環境中の物体の少なくとも一つの動きは、前記認識信号に応答し

、
前記物体の一つを制御するための前記モーションからのフィードバックは、どれほど多く
のモーション信号が受信された場合でも即座であり、

前記フィードバックは、前記モーションからの現在の最良推測を通じてなされた進捗の尺
度、及び/又は現在の最良推測とともに提供された確かさの尺度に応答する、システム。

【請求項 2】

請求項 1 に記載のシステムであって、前記処理ユニット、前記ディスプレイ及び前記モー
ション検出デバイスは、全て、前記モーションに応答する単一の手持ち式デバイスにある
、システム。

20

【請求項 3】

請求項 1 に記載のシステムであって、前記モーションに応答する前記モーション信号は、
前記モーション検出デバイス中にある慣性センサー、及び(i)モーション検出デバイス上
のボタン、dパッド、スティック及びトリガー、(ii) 1 又は複数のタッチスクリーンセン
サー、(iii)身体部分の位置及び方向情報を表すビデオストリームからのトラッキングさ
れた特徴、及び(iv)マイクロフォン、のうちの 1 又は複数からの成分出力を含む、システ
ム。

【請求項 4】

請求項 1 に記載のシステムであって、前記処理ユニットは、モーション信号と各プロトタ
イプの間の早期カットテスト計算を行う認識ランタイムライブラリを実行するように構成
されており、前記各プロトタイプは、早期カットテストが成功すると、前記モーション信
号について、さらなる検討から除外される、システム。

30

【請求項 5】

請求項 1 に記載のシステムであって、前記処理ユニットは、パラメーターのランク付けさ
れたリストを計算するように構成されており、前記パラメーターのそれぞれは、前記モー
ション信号と比較されるプロトタイプの一つの現在の進捗、前記モーション信号に対する
、プロトタイプの一つの予測での現在の確かさ、及び前記モーション信号に対する、プロ
トタイプの一つの現在の走行距離を含む、システム。

【請求項 6】

請求項 6 に記載のシステムであって、最終の認識信号は、パラメーターのランク付けされ
たリストから決定されたプロトタイプの一つに対応する、システム。

40

【請求項 7】

請求項 1 に記載のシステムであって、前記処理ユニットは、ある時点において互いに頻繁
に混同されるムーブの集合のコレクションを体系化する曖昧性除去ツリーを使用して、モ
ーション信号のどの程度がこれまでに取り込まれているのかを考慮して現在互いに混同さ
れているムーブの集合を表す、曖昧性除去ツリー中の現在のノードを同定するように構成
されている、システム。

【請求項 8】

請求項 8 に記載のシステムであって、前記処理ユニットは、バーチャルインタラクティブ

50

環境中の物体の少なくとも一つが、現在互いに混同されているムーブの集合の共通の予めデザインされたプリアンブルに応答して動くことを可能にするように構成されている、システム。

【請求項 9】

請求項 1 に記載のシステムであって、前記プロトタイプのそれぞれは、一連のデータフレームによって表され、前記認識信号の一つは、物体の一つがユーザーによってなされたモーションに対してほぼ即座に応答するということをユーザーが感じるように前記一連のデータフレームの最初の部分のみから生成される、システム。

【請求項 10】

バーチャルインタラクティブ環境中の物体を制御する方法であって、
少なくとも一つモーション検出デバイスからモーション信号を受け取り、
許されたモーションの予め定義された集合を参照せず且つモーションをどのように実行するかについて制限なくエンドユーザーによって生成されたトレーニングデータの集合から前もって作成されたモーションレコグナイザーを参照して、モーション信号からモーション認識信号を処理ユニットにおいて計算し、
バーチャルインタラクティブ環境中の物体の一つを、ユーザーがモーション検出デバイスを操っている間に連続的に生成される認識信号に応答させ、
前記物体の一つを制御するための、モーションからのフィードバックは、どれほど多くのモーション信号が受信された場合でも即座であり、
前記フィードバックは、推定の進捗の尺度、及び / 又はユーザーの現在の最良推測モーションでの確かさに応答する、方法。

10

20

【請求項 11】

請求項 10 に記載の方法であって、前記モーションに応答する前記モーション信号は、モーション検出デバイス中の慣性センサー、及び (i) モーション検出デバイス上のボタン、d パッド、スティック及びトリガー、(ii) 1 又は複数のタッチスクリーンセンサー、(iii) 身体部分の位置及び方向情報を表すビデオストリームからのトラッキングされた特徴、及び (iv) マイクロフォン、のうちの 1 又は複数からの成分出力を含む、方法

【請求項 12】

請求項 10 に記載の方法であって、少なくともモーションレコグナイザーをロードすること及び前記モーション信号から認識信号を生成することは、前記モーション検出デバイスで起こり、前記モーション検出デバイスは、バーチャルインタラクティブ環境を表示するディスプレイスクリーンを含む、方法。

30

【請求項 13】

請求項 10 に記載の方法であって、パラメーターのランク付けされたリストを計算することをさらに含み、前記パラメーターのそれぞれは、モーション信号と比較されるプロトタイプの一つの現在の進捗、前記モーション信号に対する、プロトタイプの一つの予測での現在の確かさ、及びモーション信号に対する、プロトタイプの一つの現在の走行距離を含む、方法。

【請求項 14】

請求項 10 に記載の方法であって、前記モーションレコグナイザーは、ある時点において互いに頻繁に混同されるムーブの集合のコレクションを体系化するために提供される曖昧性除去ツリーと共に組み込まれ、
この方法は、モーション信号のどの程度がこれまでに取り込まれているのかを考慮して現在互いに混同されているムーブの集合を表す、曖昧性除去ツリー中の現在のノードを同定し、
モーション信号に対して最終の認識信号が決定される前のどの時点でも、現在混同されているムーブの集合をアプリケーションにリターンする、方法。

40

【請求項 15】

請求項 14 に記載の方法であって、
バーチャルインタラクティブ環境中の物体の少なくとも一つが、現在互いに混同されてい

50

るムーブの集合の共通の予めデザインされたプリアンブルに応答して動くようにする、方法。

【請求項 16】

請求項 10 に記載の方法であって、モーション信号と各プロトタイプの間の早期カットテスト計算を行う認識ランタイムライブラリを実行することをさらに含み、前記各プロトタイプは、早期カットテストが成功すると、前記モーション信号について、さらなる検討から除外される、方法。

【請求項 17】

請求項 10 に記載の方法であって、パラメーターのランク付けされたりリストを計算することをさらに含み、前記パラメーターのそれぞれは、前記モーション信号と比較されるプロトタイプの一つの現在の進捗、前記モーション信号に対する、プロトタイプの一つの予測での現在の確かさ、及び前記モーション信号に対する、プロトタイプの一つの現在の走行距離を含む、方法。

10

【請求項 18】

請求項 17 に記載の方法であって、前記認識信号は、パラメーターのランク付けされたりリストから決定されたプロトタイプの一つに対応する、方法。

【請求項 19】

バーチャルインタラクティブ環境で物体を制御するシステムであって、ユーザーによってなされたモーションに応答するモーション信号を送り出すモーション検出デバイスとを備え、前記モーション信号は、処理ユニットに無線で取り込まれ、この処理ユニットは、モーションレコグナイザーに含まれるプロトタイプの集合にアクセスして、そのモーションレコグナイザー中のプロトタイプの 1 又は複数を全く考慮することなく、そのモーションレコグナイザーに応答して前記モーション信号から対応する認識信号を生成するように構成されており、前記認識信号は、モーション検出デバイスを用いてユーザーがモーションを行っている間に連続的に生成され、

20

前記処理ユニットは、ディスプレイにバーチャルインタラクティブ環境を表示させ、バーチャルインタラクティブ環境中の物体の少なくとも一つの動きは、前記認識信号に応答し、

前記物体の一つを制御するための前記モーションからのフィードバックは、どれほど多くのモーション信号が受信された場合でも即座であり、

30

前記フィードバックは、前記モーションからの現在の最良推測を通じてなされた進捗の尺度、及び/又は現在の最良推測とともに提供された確かさの尺度に応答する、システム。

【請求項 20】

請求項 19 に記載のシステムであって、前記処理ユニットは、モーション信号と各プロトタイプの間の早期カットテスト計算を行う認識ランタイムライブラリを実行するように構成されており、前記各プロトタイプは、早期カットテストが成功すると、前記モーション信号について、さらなる検討から除外される、システム。

【発明の詳細な説明】

【技術分野】

【0001】

40

本発明は、一般に、人工知能の領域に関し、さらに詳しくは、特に、例となるモーションからモーションレコグナイザーを生成するという文脈での機械学習に関する。いくつかの実施形態では、レコグナイザーメーカーは、エンドユーザーアプリケーションに組み込むか、これと共に用いることが可能であり、エンドユーザーは、エンドユーザーアプリケーションで用いられる、アドホックパーソナライズ化モーションレコグナイザーを生成可能である。

【背景技術】

【0002】

ソフトウェアアプリケーションのフリーフォームヒューマンモーションコントロールの約束を実現させる我々の能力は、所定の人間がしようとすることを検出及び認識する能力

50

によって厳しく制限される。大まかに言って、最も興味深いモーションコントロールの可能性は、次の人間「デバイス」(指、手、表情、及び頭の運動、肩、胴、及び脚)を解釈することから来る。人間は、他の人間のジェスチャー及び表情を解釈するのが非常に上手である。しかし、それと同じようなレベルで機能可能な機械又はコードを作成することはできない。

【0003】

提供されたモーションが既知のモーションクラス中にある集合の例であるかどうかを認識するプログラムコードを書くことは、難しい。その理由は、一つには、特殊化したモーションデータのソースが多く、それぞれが、このようなデータを処理する実用的な方法についての比較的小さい集団の公の知識であり、それぞれが、意味解釈及び作動範囲が異なり、能力のある人間なら誰でも取り上げることができる人類学的情報を何れも反映していない。結果として生じるモーションデータは、複雑で直観に反している場合が多い。例えば、3D加速度計出力対時間の単純なグラフが提示されると、当業者は、時系列のデータが対応するジェスチャーを決定しようと努力する。どのモーショングラフが同じジェスチャーに属するかを選択するというシンプルなタスクでさえ、その問題が提示された大部分のエキスパートを悩ませる。この問題は、センサー雑音、デバイス間の違い、及び体型及び筋肉組織が異なる人が演じた場合や、同じ人であっても異なる機会に演じた場合には、同じジェスチャーについてのデータが異なって見えるという事実があるので、なおさら、たちが悪い。これらの条件下では、当業者が効果的なモーションレコグナイザーを構築することは、難しい挑戦である。

【0004】

チャレンジングなソースデータと共に、データが時間に対して静的でなく動的であるという事実は、克服するのが非常に難しい。フリーフォームヒューマンモーションは、一般の意味では、経時的な動きによって特徴付けられ、それに続くモーション認識は、時系列データに対する計算によって特徴付けなければならない。ワンステップインタイムに基づいて数多くの静的特徴を計算して、識別ベースの認識を実行するという典型的なパターン認識又はジェスチャー認識アプローチは、この発明に関連しない。

【0005】

自動モーション認識システムに対する顕著な挑戦を提示するフリーフォームヒューマンモーションの第三の特性は、全ての個人ユーザーが自身の「アドホック」(つまり、予め定義されていない)モーションレコグナイザーを生成してパーソナライズすることを可能にする要望がある。先行技術は、当該分野のエキスパートがジェスチャーの予め定義された特定の集合に対して静的認識を適用可能なアルゴリズムの多くの例を含む。ジェスチャーの予め定義された集合を使う能力は、非常に多くの実用的なコーナーがカット可能であることを意味する。例えば、クラシファイアー構築回数は、数日又は数週間にすることができる。トレーニングデータは、何百万もの例を含みうる。バイアスは、3~5の互いに異なるクラスに対してうまく働くように組み込むことができるが、その外側ではうまく行かない。

クラスの予め定義された集合に特有の特性は、アルゴリズム及び対応するパラメーター中にコードすることが難しい。大雑把に言うと、数の予め定義されたクラスに対して分類を行う能力は、アドホックモーション認識を行う能力とはほとんど又は全く関係がない。我々の知識では、アドホックモーションレコグナイザーをエンドユーザーが作成することを教示する従来技術は全くない。

【0006】

以前の仕事(例: Kjeldson[3])では、手の静止イメージのコレクションを取得し、そのイメージを表す静的特徴の大きなコレクションを構成し、ツール(例: その後の静止イメージを認識可能なニューラルネットワーク)を用いてクラシファイアーを構築するシステム及び方法が説明されている。この仕事は、アドホックモーションレコグナイザーを構築することには関連していない。まず、Kjeldsonの入力データは、静止イメージデータである。時間成分がなく、混合モードの入力もない。静止分類問題に役立つ技術は、フリーフ

フォームヒューマンモーションコントロールには適用されない。さらに、Kjeldson[3]は、予め考えられた静止イメージのコレクション間の違いを識別するクラシファイアーを当業者が構築する技術にフォーカスしている。しかし、当業者でない者が予め考えられていないアドホックなジェスチャーの集合を認識するクラシファイアーを作成できるようにすることが非常に望ましい。

【0007】

以前の仕事(例:Kwon[4])では、システム及び方法は、トレーナー/トレーニーセッションを作成するように説明されている。このシステム及び方法では、トレーナーモーションを表し、受信トレーニーモーションを認識するために使用される隠れマルコフモデルが作成される。このアプローチは、トレーニーに対して40-60%のエラー率が許容可能であることに依存している。しかし、大部分のアプリケーション(例:コンピュータービデオゲーム)は、95%以上の成功率を要求する。さらに、Kwon[4]に記載された方法は、トレーニング信号中の三成分(スタート位置、モーション、及びエンド位置)を必要とする。このアプローチは、フリーフォームモーションコントロールを提供したいアプリケーションではうまくいかない。なぜなら、スタート及びエンド位置が予め定義されておらず、妥当なトレーニングセットを構築することなく、仮説に基づいて定量化することが事実上できないからである。

【0008】

本発明中の教示は、種々のアプリケーション中で使用されるアドホックパーソナライズ化レコグナイザーを作成する能力をスキルのないエンドユーザーに与えるという、前例のないステップを有する。受信データは、経時的なモーション信号が種々混合されたものである。予め定義されたジェスチャーがなく、それをどのように実行するかについての制約がなく、予め定義されたスタートポーズ又は停止ポーズもない。モーションレコグナイザーを構築するのにコーディングが必要ない。エンドユーザーは、単純に例を与えることによって、彼らが選択するどんなモーションレコグナイザーでも選択することができる。本発明の目的、特徴、及び利点は、次の詳細な説明を精査して明らかになるであろう。

【0009】

ここで十分に記載されたかのように参照によって取り込まれる参考文献の詳細は、以下のものを含む。

- [1]. E. Keogh and M. Pazzani, Derivative Dynamic Time Warping, in First SIAM International Conference on Data Mining, (Chicago, IL, 2001);
- [2]. Lawrence R. Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE, 77 (2), p. 257-286, February 1989;
- [3]. R. Kjeldson and J. Kender, Towards the Use of Gesture in Traditional User Interfaces, Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition) 1996; and
- [4]. D. Kwon and M. Gross, Combining Body Sensors and Visual Sensors for Motion Training, ACM SIGCHI ACE 2005.

【発明の概要】

【0010】

このセクションは、本発明のいくつかの観点を要約し、いくつかの好ましい実施形態を簡潔に紹介する。このセクション中の単純化又は省略は、このセクション及び要約又は発明の名称の目的を曖昧にすることを避けるために行われる。このような単純化又は省略は、本発明の範囲を限定することを意図しない。

【0011】

一般的に言えば、本発明は、エンドユーザーが例モーション(example motions)からモーションレコグナイザー(motion recognizers)を生成し、使用することに関する。本発明の一観点によれば、ホスト計算ユニット上で動作するアプリケーションに組み込むことができるアドホックパーソナライズ化モーションレコグナイザー(ad-hoc personalized mot

10

20

30

40

50

ion recognizers)を、当業者ではないエンドユーザーが作成する技術が提供される。次に、そのホスト計算ユニットは、組み込まれたモーションレコグナイザーに基づいて、モーションを直ちに認識するように構成されている。さらに、モーションレコグナイザーは、より多くのモーションを認識する能力を広げるために、常にアップデート、調整、又は改良されている。

【0012】

本発明の別の観点によれば、エンドユーザーによって作成されたモーションレコグナイザーは、様々なモーション検出デバイスの1又は複数からの、エンドユーザーモーションを特徴づけるモーション信号について、モーション認識を行うように構成されている。そのモーション認識からの認識信号は、表示に関して、アプリケーションと情報をやりとりするために使用される。モーション信号は、新たなモーションレコグナイザーを作成するために使用され、また、既に作成されたモーションレコグナイザーをアップデート又は調整するためにも使用可能である。

【0013】

本発明は、種々の形態（システム又はシステムの一部、コンピューター読み取り可能媒体中のソフトウェアモジュールを含む。）で実装可能である。一実施形態によれば、本発明は、モーションレコグナイザーを作成及び使用する方法である。この方法は、以下のものを含む。

許されたモーションの予め定義された集合を参照することなくエンドユーザーによって作成されたモーション信号のトレーニングセットを受け取り、前記トレーニングセットは、経時的に実行された少なくとも一つのタイプのモーション(at least one type of motion executed over some period of time)を特徴づけるモーション信号の第1集合を含み、モーションレコグナイザーの少なくとも一つを、前記トレーニングセットから自動的に構築し、

(1) アドホックであり且つモーション認識を行うモーションレコグナイザーの作成に必要とされる実質的に全てのパラメーターは、自動的に決定され、

(2) どのムーブが認識可能であるかに影響を与える手段は、モーションの新たな例を追加するか又はモーション信号のいくつかをトレーニングセットから取り除くことであり、モーション信号の第2集合を分類することによって、モーションレコグナイザーの少なくとも一つを用いてモーション認識を行い、一方、モーション信号の第2集合のいくつか又は全ては、モーションレコグナイザーをアップデートするか又は追加のモーションレコグナイザーを作成するために使用される。

【0014】

別の実施形態によれば、本発明は、モーションレコグナイザーを作成及び使用するシステムである。このシステムは、以下のものを含む。

モーション信号の第1集合を生成する少なくとも一つの手持ち式モーション検出デバイスと、

アドホックである少なくとも一つのモーションレコグナイザーと、許されたモーションの予め定義された集合を参照することなくエンドユーザーによって作成された少なくとも一つのトレーニングセットを記憶するためのメモリースペースとを備え、このトレーニングセットは、経時的に実行された少なくとも一つのモーションを特徴づけるモーション信号の第2集合を含み、

少なくとも一つのトレーニングセットから、少なくとも一つのモーションレコグナイザーを自動的に生成するように構成されているレコグナイザーメーカー(a recognizer maker)を有する第1の処理ユニットと、

少なくとも一つの手持ち式モーション検出デバイスからモーション信号を受け取り、少なくとも一つのモーションレコグナイザーに応答する認識ランタイムライブラリを実行し、そのモーション信号に対するモーションラベルを計算する第2の処理ユニットを備える。

【0015】

さらに別の実施形態によれば、本発明は、モーションレコグナイザーを作成する方法で

あり、この方法は、以下のものを含む。

経時的に実行された少なくとも一つのタイプのモーションを特徴づけるモーション信号の第1集合のトレーニングセットを受け取り、

このトレーニングセットから少なくとも一つのモーションレコグナイザーを自動的に構築し、このモーションレコグナイザーは、認識ランタイムライブラリによって使用されたときに、モーション信号の第2集合についてのモーション認識をサポートし、

以下に示すものの少なくとも一つをトレーニングセットから自動的に計算することを含む。

(1) (i)全体の分類率、(ii)クラス毎の分類率(per-class classification rates)における差異、又は(iii)所望の「未決定」分類率、の関数として、トレーニングセットからモーション信号を追加又は削除することなくモーションレコグナイザーのクラス毎の分類許容値(per-class classification tolerances)を制御するために使用されるスラックパラメーター(slack parameters)の集合、

(2) (i)モーションレコグナイザーのクラス数、(ii)各クラスの必要な分類率、又は(iii)所望の「未決定」分類率の関数として、モーションレコグナイザーの認識キャパシティを制御するために使用されるキャパシティパラメーター、

トレーニングセット中のどのモーションクラスが新たなモーション信号でアップデートするか完全に再デザインする必要があるのかを示すことによって、レコグナイザーメーカーをインタラクティブに使用するように導くために使用される混同マトリックス(a confusion matrix)。

【0016】

さらに別の実施形態によれば、本発明は、モーションレコグナイザーを作成するシステムであり、このシステムは、以下のものを含む。

モーション信号の第1集合を生成する少なくとも一つのモーション検出デバイスと、少なくとも一つのモーションレコグナイザー、及び経時的に実行された少なくとも一つのモーションを特徴づけるモーション信号の第2集合を含む少なくとも一つのトレーニングセットを記憶するためのメモリースペースと、

少なくとも一つのモーション検出デバイスからのモーション信号の第1集合を受け取り、少なくとも一つのモーションレコグナイザーに応答する認識ランタイムライブラリを実行し、モーション信号の第1集合に対するモーションラベルを計算する第1の処理ユニットと、

少なくとも一つのトレーニングセットから少なくとも一つのモーションレコグナイザーを自動的に作成し、さらに、以下に示すものの少なくとも一つをトレーニングセットから自動的に計算するように構成されているレコグナイザーメーカーを有する第2の処理ユニット。

(1) (i)全体の分類率、(ii)クラス毎の分類率における差異、又は(iii)所望の「未決定」分類率、の関数として、トレーニングセットからモーション信号の第1集合の何れも追加又は削除することなくモーションレコグナイザーのクラス毎の分類許容値を制御するために使用されるスラックパラメーターの集合、

(2) (i)モーションレコグナイザーのクラス数、(ii)各クラスの必要な分類率、又は(iii)所望の「未決定」分類率、の関数として、モーションレコグナイザーの認識キャパシティを制御するために使用されるキャパシティパラメーター、

トレーニングセット中のどのモーションクラスが新たなモーション信号でアップデートするか完全に再デザインする必要があるのかを示すことによって、レコグナイザーメーカーをインタラクティブに使用するように導くために使用される混同マトリックス。

【0017】

さらに別の実施形態によれば、本発明は、モーションレコグナイザーを作成する方法であり、この方法は、以下のものを含む。

1又は複数のモーション検出デバイスからのデータのトレーニングセットとしてモーション信号を受け取り、モーション信号のそれぞれは、経時的に実行された少なくとも一つの

タイプのモーションを特徴づけ、
モーション信号のそれぞれにおいて特徴付けられる、モーションのスタートの前又はエンドの後のデータを含むモーション信号のそれぞれについてのデータのエンベロープ(an envelope of data)を記録及び保持し、

トレーニングセット中のモーション信号のそれぞれのスタートの前、間及び後のモーション信号アクティビティ中の差異を含む特徴に基づいて、モーションのスタートを予測するモーションスタートクラシファイアー(classifier)を作成するためにモーション信号のそれぞれを解析し、

モーションが開始されたことをモーションスタートクラシファイアーを示した場合には受信モーション信号ストリームを自動的にモーションスタートとラベルづけする。

10

【0018】

さらに別の実施形態によれば、本発明は、モーションレコグナイザーを作成する方法であり、この方法は、以下のものを含む。

モーション検出デバイスを用いて経時的に実行された少なくとも一つのタイプのモーションを特徴づけるモーション信号の第1集合で構成されるトレーニングセットから作成されたモーションレコグナイザーを受け取り、このモーション信号は、モーション検出デバイスの経時的な位置及び方向を計算するのに十分な情報を含み、

第2モーション検出デバイスの経時的な位置及び方向を計算するのに十分な情報を提供する、第2モーション検出デバイスからのモーション信号の第2集合を受け取り、

モーション認識を行って、モーション信号の第2集合中の第2の例に最も応答する、トレーニングセット中の第1の例モーション信号を決定し、

20

第1の例モーション信号の第1の3D追跡、及び第2の例モーション信号の第2の3D追跡をいつでも計算し、

2つのモーションの間の少なくとも最初の主要な相違点を強調して第1及び第2の3D追跡を並べて描画する。

【0019】

多くの目的、特徴、恩恵及び利点は、上記のものと共に、次の説明において、本発明の実施において達成され、添付図面で示した実施形態が得られる。

【図面の簡単な説明】

【0020】

30

本発明のこれら及び他の特徴、観点、及び利点は、次の説明、付属の請求項及び添付図面に関連して、よりよく理解されるであろう。

【0021】

図1Aは、本発明の一実施形態による構成を示し、この構成では、ディスプレイ、コントローラー、及び計算ユニットが3つの別々のデバイスである。

【0022】

図1Bは、モーションを行っているプレーヤーから生じる例示的モーション信号を示す。

【0023】

図2は、本発明の別の実施形態による、コントローラーとして働く単一のデバイスを示す。このデバイスは、ディスプレイ及び計算ユニットを含む。

40

【0024】

図3は、例示的構成を示す。この構成では、モーション信号は、エンドユーザーの種々の動き及びアクションを測定する多くの異なる入力で構成され、そのデータに応答するモーションレコグナイザーを構築可能なレコグナイザーメーカーに与えられる。

【0025】

図4は、本発明の一実施形態による、パーソナライズ化アドホックモーションレコグナイザーを作成するプロセスを示す。

【0026】

図5は、本発明の一実施形態による、モーション認識を行うプロセスを示す。

【0027】

50

図6は、アドホックパーソナライズ化モーションレコグナイザーを作成するとともに、その同じレコグナイザーを使用するモーション検出アプリケーションと情報をやりとりして、モーションコントロールを提供するプロセスを示す。

【発明を実施するための形態】

【0028】

本発明の詳細な説明は、大きくは、手順、ステップ、論理ブロック、及び処理の観点と、及びデータ処理デバイスの動作に直接又は間接的に類似した他のシンボル表現の観点で行う。これらのプロセス説明及び表現は、典型的に、当業者が他の当業者に対して自分の仕事の内容を最も効果的に伝えるために用いられる。

【0029】

本発明の完全な理解のために、多くの具体的な詳細を述べている。しかし、これらの具体的な詳細がなくても、本発明が実施可能であることは、当業者にとって自明になるであろう。別の例では、周知の方法、手順、成分、及び回路は、本発明の観点を不必要にばやかすことを避けるために詳細には述べていない。

【0030】

本明細書において、「一実施形態」又は「1実施形態」への言及は、その実施形態に関連して説明される特定の特徴、構造、又は特性が本発明の少なくとも一実施形態に含まれ得ることを意味する。明細書中の種々の場所において現れるフレーズ「一実施形態において」は、必ずしも全てが同じ実施形態を参照することではなく、他の実施形態と相互に排他的な別々の又は二者択一の実施形態を参照することでもない。さらに、本発明の1又は複数の実施形態を表すプロセスフローチャート又はダイアグラムでのブロックの順序は、本質的に、特定の順序を示すものではなく、本発明を限定することを意味するものでもない。

【0031】

便宜上、いくつかの用語の定義を以下に示す。この定義は、一実施形態による、本発明の理解及び説明を容易にするためのものであることに注意すべきである。この定義は、その実施形態に関する限定を含むように見えるかも知れないが、その用語の実際の意味は、そのような実施形態を超えて適用可能である。

1 定義

【0032】

アドホックモーションレコグナイザー：許容可能なモーションの概念を予め定義することなく、且つこれらのモーションを実行する許容可能な方法の概念を予め定義することなく構築されたモーションレコグナイザー。

【0033】

キャパシティ：所定のモーションレコグナイザーにおいて許されるプロトタイプの数値を制御するパラメーター。キャパシティは、所定のモーションレコグナイザーの予測されるメモリ及びCPUコストの代用物(proxy)として働く。

【0034】

分類(Classification)：ラベルのないモーション信号に対して、クラスラベル又はモーションラベルの割り当てを行うプロセス。割り当てられたクラスラベルは、「未知の」又は「未決定」となる可能性もある。分類は、おそらく追加のファクターに応じて、ラベルのない例が、可能性のある各クラスの例である確率の割り当てを追加的に行ってもよい。この場合、割り当てられたラベルは、見込みが最大のクラスである(Classification might additionally assign probabilities, possibly in response to additional factors, that an unlabelled example is an example of each possible class, in which case the assigned label is the class with greatest likelihood.)。

【0035】

分類距離：分類距離は、特定のモーションクラスでの所定のモーションプロトタイプに特有の閾値である。この範囲内では、プロトタイプは、モーション信号が「クラス内」で

10

20

30

40

50

あると分類し、この範囲外では、プロトタイプは、モーション信号に無関係である。

【0036】

分類率：統計的尺度(statistical measures) (例：偽陽性及び偽陰性の数)の集合に応じたモーションレコグナイザーパフォーマンスの尺度(A measure of motion recognizer performance)。

【0037】

クラシファイアー：本明細書においては、この用語は、分類を行う計算デバイスが解釈することができるソフトウェアインストラクションを指す。この用語は、モーションレコグナイザーと互換的に使用される。

【0038】

開発者：アプリケーションの作成に関わる人。本明細書においては、これは、ゲームプログラマー、AIプログラマー、プロデューサー、レベルデザイナー、テスター、雇われた請負人などを含むがこれに限定されない。

【0039】

エンドユーザー：これは、アプリケーションが対象としているユーザーである。例えば、ビデオゲームアプリケーションについてゲームプレーヤー、携帯電話については携帯電話ユーザーである。

【0040】

モーション：位置を変えるアクション又はプロセス。これは、意図的な及び意味のあるモーション(例：手紙を書いたり、手首を捻ってスクリュードライバーの使用をシミュレートしたりすること)を含む。また、意図的でないモーション(例：退屈又は緊張しているときに、そわそわすること)も含む。

【0041】

モーションプロトタイプ：モーションプロトタイプは、モーションレコグナイザー中のモーション信号の、あるクラスについての代表的モーションの集合のメンバーであると選ばれた、(生の又は処理された)モーション信号である。

【0042】

モーションレコグナイザー：モーション分類を行う計算デバイスが解釈することができるソフトウェアインストラクション。本明細書において、「プリディクター」("predictor")という用語は、モーションレコグナイザーと互換的に使用される。

【0043】

モーション信号：モーション信号は、ある期間にわたるモーション(例として、図1bを参照)を表現する情報(例：時系列データ)である。このデータは、多くの形態を取りうる。例えば、物体の経時的位置、物体の経時的方向、物体が経験する経時的加速度、物体が経験する経時的力、周波数領域で表現されたデータ、パラメーター化された領域で表現されたデータ(例：R3又はR4など)である。モーション信号は、モーションと呼ぶこともある。本明細書において、モーション信号は、処理されたモーション信号又は生のモーション信号を指す可能性がある。生のモーション信号は、モーション検出デバイスのデバイスドライバーから直接得られるデータを表す。処理されたモーション信号は、モーション検出デバイスからのデータがさらに処理又は変換され、もはや「生の」状態ではなくなっているものを表す。

【0044】

スラック(Slack)：プロトタイプ分類距離に関して非線形乗数として働くパラメーター。スラックが大きいほど、関連するプロトタイプが所定の例モーションを分類する可能性が高まる。同様に、スラックが小さいほど、プロトタイプが例モーションを分類する可能性が低下する。一実施形態では、スラックは、モーションレコグナイザーにおける、所定のクラスの分類許容値(classification tolerance)の効率的な説明である。

【0045】

トレーニングセット：モーションレコグナイザーを生成するために用いられる(生の又は処理された)モーション信号の集合。トレーニングセットが取ることができる考えられ

10

20

30

40

50

る形態には、様々なものがある。本明細書においては、トレーニングセットは、モーションのサブセットのコレクションである（所定のサブセットの全てのメンバーが同じ明示的又は黙示的ラベルを共有する。）。例えば、明示的クラスラベルは、「フォアハンド」、「バックハンド」及び「サーブ」にしてもよい。明示的ラベルが得られない場合には、その代わりに、そのモーションが属するサブセットに基づいて、黙示的ラベルが導き出される。例えば、モーション信号のラベルのないサブセットがトレーニングセットに5つ存在している場合、モーションの各サブ集合に対する黙示的ラベルは、それぞれ、「サブセット1」、...、「サブセット5」にしてもよい。

2 実施形態の詳細な説明

10

【0046】

図1Aは、本発明100の実施形態を示す。この実施形態では、ディスプレイ103、コントローラー102、及び計算ユニット108が3つの別々のデバイスである。この配置は、この発明の一実施形態のホストとして使用される典型的なコンピュータビデオゲームシステム（例：Nintendo Wii又はSony PS3）を反映する。

【0047】

既に定義したように、エンドユーザー101は、コントローラー102を動かすことによって、計算ユニット108にある種々のアプリケーションに供給されるモーション信号を生成している、典型的なコンシューマー又はエンドユーザーである。この発明での特徴、利点及び恩恵の一つは、エンドユーザーに新たな機能を提供することである。それは、計算ユ

20

【0048】

この実施形態でのコントローラー102は、1又は複数の内蔵の慣性検出デバイス（例：加速度計、ジャイロスコープ及び磁力計）を含むモーション検出デバイスである。エンドユーザー101がこれを動かすと、モーション信号104のストリームが生成される。モーション信号104のストリームは、計算ユニット108に送られる。

【0049】

モーション信号104は、コントローラー102の出力であり、計算ユニット108にロバストで効率的に（例：有線又は無線で）送信を行うようにまとめられる。図1Bは、投げ縄をス

30

【0050】

レコグナイザーメーカー105は、計算ユニット108にあるモジュールである。レコグナイザーメーカー105は、エンドユーザーのために、アドホックパーソナライズ化モーションレコグナイザーを作成する。レコグナイザーメーカー105は、モーション信号104を入力として受け取り、新たなレコグナイザー106をアップデート又は作成し、次に、ディスプレイ103をアップデートし、レコグナイザー作成プロセスについてエンドユーザー101にフィードバックを提供するように構成されている。レコグナイザーメーカーは、この及び他の実施形態では、エンドユーザー向けに作成されており、開発者向けではない。レコグナイザーメーカーは、どのムーブを含めるか、及びそのムーブがどのように実行されるかについて、エンドユーザーに完全な自由度を与えるものでなければならない。

40

【0051】

50

一実施形態によれば、アプリケーション・プラス・認識ランタイムライブラリ107は、計算ユニット108上のアプリケーション（それぞれが独立してモーション認識ランタイムライブラリを取り込むように構成されている。）（例：ビデオゲーム）のコレクションである。各アプリケーションは、入力の一部としてモーション信号104を取り込み、1又は複数のレコグナイザー106に対して応答する。アプリケーション107は、エンドユーザー101のモーションに応答するように、ディスプレイ103と内部状態をアップデートする。一般に、ビデオゲームのようなアプリケーションに対しては、モーションレコグナイザー106は、全ての年代の数百万の異なるプレイヤーを対象としなければならない。従って、モーションレコグナイザー106は、身体構造の違い、モーション全体の力及び長さのバリエーション、コントローラーのグリップの違い、スタート及びエンド方向の変化による、モーション信号データのバリエーションに対してロバスト(robust)でなければならない。これらのバリエーションの全てが基礎となるモーションデータに対して驚くほどのインパクトを与える。

10

【0052】

計算ユニット108は、コントローラー102からの入力を受信し、レコグナイザーメーカー105、アプリケーション107及びレコグナイザー106をロードして実行し、ディスプレイ103をアップデートする手段を提供する役割を有する。

【0053】

図2は、この発明の実施形態200を示す。この実施形態では、ディスプレイ、コントローラー及び計算ユニットは、全て、単一のデバイスとして一体化されている。この配置は、この発明のホストとして使用される典型的なモバイルシステム（例：Apple iPhone又はSonyPSP）を反映する。本発明での特徴、利点及び恩恵の一つは、計算ユニット208中の1又は複数のアプリケーション206に対する、独自のパーソナライズ化モーションコントロールインターフェースをユーザーが作成する機能を提供することである。

20

【0054】

エンドユーザー201は、モーションセンサー202（例：内蔵の慣性センサ）の集合を含む計算ユニット208を動かすことによって、モーション信号を生成する。このモーション信号は、作成されたモーションレコグナイザーを用いてモーション認識を実行するように構成されている認識ランタイムライブラリ(RTL)203に与えられる。モーションセンサー202は、計算ユニット208をあちこち移動させたときにモーション信号を生成する内蔵のセンサーであり、この信号は、認識ランタイムライブラリ203に与えられる。

30

【0055】

認識ランタイムライブラリ203は、計算ユニット208にある1又は複数のアプリケーション206によって共有される。RTL203、アプリケーション206及びレコグナイザーメーカー207の間には、モーションコントロールサービスレイヤー205が介在して共有が可能になっている。認識RTL203は、モーションセンサー202からのモーション信号の一定のストリームを受信し、1又は複数のレコグナイザー204に応答して、アプリケーション206及びレコグナイザーメーカー207に対してモーション認識信号及びフィードバックを提供する。システムフィードバックは、計算デバイス208によって、エンドユーザー201に表示される。

40

【0056】

レコグナイザーメーカー207は、計算ユニット208にあるモジュールである。レコグナイザーメーカー207の主な役割は、エンドユーザー用のアドホックパーソナライズ化モーションレコグナイザーを作成することである。レコグナイザーメーカー207は、処理されたモーション信号をRTL203から受け取り、次に、モーションセンサー202から連続的に受け取る入力及び/又は新たなモーション信号に基づいて、新たなレコグナイザー204をアップデート又は作成し、次に、計算ユニット208にあるディスプレイをアップデートして、レコグナイザー作成プロセスについてエンドユーザー201にフィードバックを提供する。レコグナイザーメーカーは、この及び他の実施形態では、エンドユーザー向けに作成されており、開発者向けではない。レコグナイザーメーカーは、エンドユーザーの手元にある計算ユニットにおいて動作可能でなければならない。さらに、レコグナイザーメーカーは、

50

どのムーブを含めるか、及びそのムーブがどのように実行されるかについて、エンドユーザーに完全な自由度を与えるものでなければならない。

【 0 0 5 7 】

モーションコントロールサービスレイヤー205は、センサー202、RTL203及びレコグナイザー204の組み合わせによって提供される共有のモーションコントロールサービスを、アプリケーション206が探し、これに結合し、これを利用する手段を、計算ユニット208上で動作する全てのアプリケーションに提供する。アプリケーション206に提供されるサービスは、モーション分類及び他の関連した信号、モーション認識チューニング、及びレコグナイザーメーカー207によって利用可能になった新たなモーションコントロールインターフェースをセーブ及びロードするに対して能力を提供することを含む。

10

【 0 0 5 8 】

この発明は、100又は200に記載された特定のハードウェア構成に限定されない。例えば、計算ユニット108及びコントローラー102は、スマートフォンであってもよい。スマートフォンは、ディスプレイデバイス103(例：テレビジョン又はプロジェクター)の制御に使用可能である。同様に、計算ユニット108は、ディスプレイデバイス103としてのモニター又はテレビジョンに接続され且つコントローラー102として働くペンシル及びモーション信号104を提供するwebカメラトラッキングアプリケーションを有する標準ラップトップPCであってもよい。一実施形態では、計算デバイス108及びwebカメラは、ぬいぐるみ又は他のおもちゃに組み込まれ、コントローラーは、子供がTeddyで遊ぶときの子供の手である。他のアプリケーションは、脳卒中リハビリ用の医療アプリケーションを含んでもよい。この場合、フィジカルトレーナーは、在宅患者に対して、その特定のニーズに対してパーソナライズ化された新たなモーションコントロール療法を構築することができる。

20

【 0 0 5 9 】

図3は、この発明の実施形態による機能ブロック図300を示す。モーション信号304は、ゼロ又はそれより多くのモーション検出デバイス302を握っているエンドユーザー301の動き及びアクションを測定する互いに異なる多くの信号で構成される。信号304は、モーション検出デバイス302に応答する汎用モーションレコグナイザー306を構築可能なレコグナイザーメーカー305に送られ、モーション検出アプリケーション308及び認識RTL307にも送られる。この実施形態の特徴、恩恵及び利点の一つは、エンドユーザーが、互いに異なる多くのタイプのモーション(手の大きな動きを含むモーションだけではない。)の認識に利用可能な非常に汎用のパーソナライズ化アドホックモーションレコグナイザーを作成する機能を提供することである。

30

【 0 0 6 0 】

モーション検出デバイス302は、エンドユーザー301の様々なアクティビティをキャプチャーする種々のタイプの複数のデバイスを含んでもよい。モーション検出デバイス302からの生のモーション信号は、種々の方法(後述する。)で生の信号を処理して、処理されたモーション信号304を作成するアダプター310を通過する。この実施形態では、種々のタイプのモーション信号ストリーム304を検出するモーションレコグナイザー306を構築するレコグナイザーメーカー305が、このような信号を生成するハードウェアとともに重要である。

40

【 0 0 6 1 】

アプリケーション308は、計算ユニット309上のサービスとして全てのアプリケーションが利用可能な外部認識RTL307と直接的に情報のやりとりを行ってもよく、認識RTLを直接組み込んでもよい。

【 0 0 6 2 】

この実施形態でのモーション信号のソースの例は、1又は複数のエンドユーザー301のそれぞれの手にある、ボタンプレス又はジョイスティック移動を含む出力を有する慣性検出コントローラーを含む。ボタンプレス又はジョイスティック移動は、実空間でのフィジカルモーションと同期していてもよく、モーション信号ストリーム304の一部を形成する。

50

【 0 0 6 3 】

例は、エンドユーザーの頭又は肩又は胴の、イメージプレーンに対する追跡位置及び方向(the tracked positions and orientations relative to the image plane of the head or shoulders or torso of the end user)を、いくらか処理して出力し、モーション信号304の一部を構成するwebカメラを含む。

【 0 0 6 4 】

他の例は、タッチ検出スクリーン上のトレースを含む。このようなトレースは、モーション信号304の一部を形成する。他の例も可能であり、この発明の範囲内であると考えべきである。この実施形態は、データタイプの様々な集合で構成されるモーション信号304のコレクションの全ての成分に応答するアドホックパーソナライズ化モーションレコグナイザー306を作成可能なレコグナイザーメーカー305を使用するエンドユーザー301にかかっている。ディスプレイ303、認識RTL307、アプリケーション308、及び計算ユニット309の説明は、実施形態100及び実施形態200での対応するものに対する説明に似ている。レコグナイザーメーカー305は、図4に記載されたプロセスに類似したプロセスを実行する。認識RTL307は、図5に記載されたプロセスに類似したプロセスを実行する。

【 0 0 6 5 】

図4は、この発明の実施形態による、アドホックモーションレコグナイザーを作成するプロセス400のフローチャートを示す。プロセス400は、ソフトウェア(例：図1の105のようなレコグナイザーメーカーモジュール)、ハードウェア又はその両方の組み合わせで実装可能である。プロセス400の特徴、恩恵又は利点の一つは、エンドユーザーがオンラインで(例：エンドユーザーが待っている間に)ホスト計算プラットフォーム上にロバストなアドホックモーションレコグナイザーを作成する機能を提供することである。

【 0 0 6 6 】

トレーニングセットは、401でロードされる。トレーニングセットは、1又は複数のクラスを備える。各クラスは、同じクラスラベルを共有するモーション信号のサブ集合によって表される。各モーション信号は、エンドユーザーの経時的なモーションを表している。トレーニングセットは、その全体がエンドユーザーが行ったモーション例、又はエンドユーザーが含まれることを選択したでモーション例から作成可能である。実装にもよるが、モーション信号は、生の又は処理されたものである。プロセス400の説明の目的で、ここでは、モーション信号が処理されたものであると仮定する。

【 0 0 6 7 】

モーションレコグナイザー構築は、トレーニングセット中の全てのモーション信号の間の全ての対間距離を知ることが必要とする。全ての対間距離(pairwise distances)が必要であるが、計算が必要なのは一部であり、残りは推測が可能である。全ての対間距離を計算して記憶することは、典型的なホスト計算ユニット上の典型的なトレーニングセットについて、非現実的である。

【 0 0 6 8 】

402では、トレーニングセットを受け取ると、レコグナイザーメーカーは、トレーニングセット中の全てのモーション信号間の対間距離の可能な限り小さいサブ集合を計算するプロセスを開始する。トレーニングセットの同じクラスに属する実質的に異なる全てのモーション間の対間距離が計算される。一実施形態では、距離尺度(the distance measure)(又は「コスト関数」)は、2つのモーション信号中のフレーム毎の差異に対していくらかのコストを割り当てる、独自のタイムワープベースの尺度(a unique, a time-warp based measure)である。図1B中の例示的信号には、400フレームがあり、各フレームには、4つの浮動小数がある。このモーションは、第2モーション(例えば300フレームを有する。)と比較可能である。距離尺度にはいくつかのコスト成分(例：各ポイントでの一次、二次及び三次導関数の差異、経時的な弾性に対する感度の差異(different sensitivities to elasticity over time))がある。これらの全ては、重み付けがなされ、異なるように組み合わせられる。

【 0 0 6 9 】

403では、402で計算された対間距離のクラス毎の集合を受け取ると、トレーニングセット中の各クラスについて、その距離に基づいて、モーションのクラスターが計算される。各クラスターの幅は、同じクラスターの2つのメンバーの間の最大距離として計算される(The width of each cluster is computed as the maximal distance between two members of the same cluster.)。クラスター代表が一つ選ばれる。クラスターは、402で計算されなかったでモーション間の対間距離がクラスター代表間の距離にほぼ等しいと推測するために使用されるので、クラスター幅は、最小化される。

【0070】

404では、正確に推測できない、残りの対間距離が計算される。最初に、全てのクラス中の全てのクラスター代表間の対間距離が計算される。次に、後で詳細に説明するように、互いに混同しそうな2つのクラスターの全てのメンバーについて、対間距離が計算される。404の終了までに、全ての対間距離が計算又は推測され、プロセス400は、405に進み、プロトタイプ選択が開始する。

【0071】

初めて405に入ると、トレーニングセット中の全てのモーション信号が、(例えば、415、つまりレコグナイザーの確定(finalize recognizers)において)作成中のモーションレコグナイザー中のプロトタイプの候補とみなされる。トレーニングセット中の各クラスは、クラシファイアーのキャパシティがいくらかに設定されていても、レコグナイザー中に少なくとも一つのプロトタイプを有さなければならない。405に入る度にクラス毎の最良の候補プロトタイプ(the best candidate prototype per class)が再計算される。後で詳細に説明するように、最良の候補は、クラス中の他のメンバーの分離度(DOS)を最も減少させるものである。

【0072】

クラス毎の最良の候補プロトタイプが405で計算されると、406でテストが行われ、これが405でのプロトタイプ選択の最初のパスであるかどうかチェックされる。そうであれば、プロセス400は、407に進み、クラス毎の最良の候補が415(つまりレコグナイザーの確定)で生成されるモーションレコグナイザー中のプロトタイプとして追加される。そうでなければ、ただ一つのプロトタイプとして追加される。

【0073】

これがプロトタイプ選択の最初のパスでない場合(つまり406が不成立の場合)であって、411のテストにパスする場合(例：現在の、不完全モーションレコグナイザーのクラス毎の精度のバランスが取れていて、何れのクラスの認識能力も二番目に最悪なものよりもはるかに悪いということがない場合)、次に、412では、現在の、全体として最良の候補(the current best overall candidate)が415で生成される最終のレコグナイザーにプロトタイプとして追加される。そうでなければ、414では、現在の最も性能が悪いクラスに対する候補が選択されて、次のプロトタイプとして追加される。

【0074】

関数407、408及び412のそれぞれは、プロセス400が408に進む前に、プロトタイプとして選択される候補を確定する。例えば、所定の候補に対して405で計算された最良の分類距離(a best classification distance as computed in 405 for a given candidate)は、プロトタイプの一部として、設定及び記憶される。

【0075】

408では、キャパシティに到達すると、プロセス400は、415に進み、レコグナイザーが確定され、レコグナイザーメーカーは、416で終了する。キャパシティに到達していない場合、409で幅チェック(a width check)が行われる。415で生成されたモーションレコグナイザーは、どのムーブが利用可能であるか、及びエンドユーザーがそれをどのように実行すべきかについて制限なく、完全に、エンドユーザーによって生成されたものである点に注目すべきである。

【0076】

409では、幅チェックが行われる。残りの候補全てのDOSが0である場合、幅チェックは

10

20

30

40

50

成功である。この時点で、トレーニングセット401について、予測能力を高める候補がこれ以上追加できない。(At this point, no more candidates can be added that increase prediction performance on the training set 401)。推奨されるキャパシティは、この幅チェックを通過する最初の数回の関数として、所定のトレーニングセットに対して自動的に設定される(Recommended capacity is set automatically for any given training set as a function of the first few times this width check passes.)。

【 0 0 7 7 】

幅チェック409が成功すると、413では、全ての残りの候補について、既に選択されたプロトタイプの影響を無視して、DOSが再計算され、制御は、405に戻る。これは、モーションレコグナイザーを作成する際に、ユーザーが選択したキャパシティを完全に利用することを可能にする。候補プロトタイプの次の集合は、あたかもトレーニングセット401のサブサンプル(既に選択されたプロトタイプが存在しないもの)で動作するように、選択される。最初のパスの後に413を通過して追加されたプロトタイプは、確定後のレコグナイザーが実行されたときの認識能力を大きく向上させる傾向がある。

【 0 0 7 8 】

幅チェックが不成立である場合、全ての残りの候補に対するDOSが、407、412又は414からの最新のプロトタイプの追加を考慮して、アップデートされ、プロセス400は、405に進む(When the width check fails, the DOS for all remaining candidates is updated given the addition of the latest prototypes from 407, 412 or 414, and process 400 goes to 405.)。

【 0 0 7 9 】

レコグナイザーが415で確定された後、プロセス400は、問題検出及びフィードバック418に進む。一実施形態では、このプロセス400の実行中に生じる場合があるいくつかの問題が検出され、(実施形態100又は200のように)システムディスプレイを通じて、エンドユーザーに報告される。

【 0 0 8 0 】

本発明の一実施形態に関して、図5は、モーション認識ランタイムライブラリ(RTL)(例：図3中の認識RTL307)によって、モーション認識がどのように実行されるかについてのプロセス500のフローチャートを示す。この実施形態の特徴、恩恵及び利点の一つは、エンドユーザーが作成したパーソナライズ化アドホックモーションレコグナイザーを、エンドユーザーが、ロバストに、フレキシブルに、且つ効率的に使用する機能を提供することである。これにより、はるかに広い範囲のモーション検出アプリケーションが利用可能になる。

【 0 0 8 1 】

502では、RTLは、少なくとも一つのモーションレコグナイザーを選択及びロードすることによって初期化される。初期化の後、503では、レコグナイザー中の全てのプロトタイプに対する分類距離は、クラス毎の、おそらくモーション信号タイプ毎の関数(スラック)として、修正される。これによって、分類能力が、エンドユーザーによって制御可能になり、モーションレコグナイザーの構成を修正することなく分類能力を調節することができる。

【 0 0 8 2 】

505での分類開始の前に、新たな生のモーション信号504が認識RTLに加えられる。実際には、モーション信号504を生成する別個のモーションストリームが複数存在し、それと共に、並列に走る複数の別個の計算スレッド505-516が存在する場合がある。例えば、NintendoWiiは、全部で8つのモーション検出コントローラーを有しており、それぞれが、並列で認識可能なモーション信号504を生成している。

【 0 0 8 3 】

レコグナイザーが初期化され、新たな生のモーション信号が取り込まれると、505では、このプロセス500と情報をやりとりするアプリケーションは、モーションストリーム504について「分類開始」をコールすることができる。データバッファーは、アプリケーション

10

20

30

40

50

ンとRTLの間で共有され、これによって、505で開始した現在の認識スレッドは、モーション信号504が生成される度にフレーム毎に、モーション信号504にアクセスすることができる。このデータバッファに対する各アップデートは、生のモーション信号504データの0、1又は複数のフレームを含むことができる。

【0084】

分類が開始されると、506では、現在未処理のモーション信号データが1フレームずつ処理される。一実施形態では、この処理は、アダプティブフィルタリングによって行われ、生のデータの多くは、そのデータが507へ送られる前に、興味深いイベントを強調するように要約される。「興味深い」とは、動きの1又は複数の成分の全体の速度又は加速度がある閾値を超えて増加したか、又は最後に処理されたポイントが生成されてから十分な時間が経過した、フレームを要約することを意味することができる("Interesting" may mean summarizing frames where overall speed or acceleration of one or more components of the movement has increased over some threshold, or where a sufficient amount of time has gone by since the last processed point was generated.)。さらに、図1Bに関して、スタートサンプル114の前のサンプルとエンドサンプル116の後のサンプルが除去される。典型的に、慣性データに対しては、アダプティブフィルタリングによって、生の受信信号が50-90%圧縮される。例えば、図1b中の例示的モーションに対しては、生の入力の400フレームが変換されて、処理された入力の40ポイントになる。従って、507は、40回だけアクセスされる。

【0085】

507では、処理されたデータポイントが506から生成されると、レコグナイザー中の全ての残りのプロトタイプに対して走行距離(a running distance)がアップデートされる。使用される距離メトリック(the distance metric)は、図4の402と同じである。一実施形態では、反復ダイナミックプログラミング法を用いて、距離メトリックのインクリメンタルアップデートが行われる。

【0086】

508では、全ての残りのプロトタイプに対して、早期カット計算(an early cut computation)が行われる。この計算は、モーション信号504に対する現在の走行距離を鑑みて、予測される最良の最終距離がプロトタイプのスラック修飾された分類距離内であるかどうかをチェックする。その答えがノーであれば、そのプロトタイプは、アプリケーションからの新たな「分類開始」信号が新たな分類スレッド505-516を再開するまで、さらなる検討から除外される。

【0087】

509では、現在の最良予測(the current best prediction)がアップデートされる。予測の形態には多くの種類がある。一実施形態では、予測は、プロトタイプのランク付けされたリストであり、各プロトタイプを通じた現在の進捗、予測における現在の確かさ、モーション信号に対する現在の走行距離を備えている。これは、「いつでも」予測("anytime prediction)をリターン可能にするために本質的である。

【0088】

510では、残っているプロトタイプがなければ、515で、現在の最良予測がリターンされ、516で、スレッドが終了する。定義により、この予測は、「未決定」又は「未知の」である。

【0089】

残っているプロトタイプがあり、かつペンディングになっている「分類終了」コール511があれば、制御は、再び、515に戻る。この場合、515によってリターンされる予測は、現在の最良予測509の関数である。例えば、それは、残っているプロトタイプのうち現在最高位にランク付けされたクラス(the class of the currently highest-ranked remaining prototype)である。また、それは、現在の最良予測中の、全ての残っているプロトタイプの重み付けがなされた多数決(a weighted majority vote of all remaining prototypes in the current best prediction)であってもよい。

【 0 0 9 0 】

ペンディング中の分類終了コールがなければ、他のペンディング中のクエリー512のチェックが行われる。一実施形態では、クエリーは、「エンドユーザー認識スコア(ロックイン採点(lock-in scoring)を参照)が何であるか」、「所定のモーションを通じた現在の進捗(the current progress)が何であるか」、「現在の最良推測中の確かさが何であるか(what is the confidence in the current best guess)」及び「混同されたムーブの集合が何であるか」を含む。ペンディング中のクエリー512は、513において、509で計算された最良の現在の予測からなされた種々の計算で解決される。次に、両方の場合に、制御は、506に戻り、処理すべきモーション信号504データの次のビットを待っている間、計算が停止する。

10

【 0 0 9 1 】

図6は、レコグナイザーを使用するモーション検出アプリケーションと情報をやりとりすると同時にレコグナイザーを作成するプロセス600のフローチャートを示す。プロセス600は、ユーザーがアプリケーション(例：ビデオゲーム)を実行するときに開始可能である。プロセス600は、エンドユーザーが作成したアドホックレコグナイザーによってアプリケーションが制御されることを可能にする。このアドホックレコグナイザーは、個人的且つエンドユーザーに特有のものであり、アプリケーションが実行されているときに同時に作成又はアップデートされる。この機能の特徴、恩恵及び利点の一つは、アプリケーションがエンドユーザーに即座に適合し、知的な感覚を与えることができ、エンドユーザーが、アプリケーションに対して洗練されたパーソナライズ化モーションコントロールを有することができることである。

20

【 0 0 9 2 】

602では、プロセス600は、存在しているモーションレコグナイザーをロードすることによって開始する。このモーションレコグナイザーは、一実施形態では、図4のプロセス400に従って生成され、アプリケーションに予め定義されているか又は予めロードされているものである。

【 0 0 9 3 】

ユーザーは、603での要求に応答してコントローラーを動かす。これは、ビデオゲームに対するアクションを行うためであってもよく、又は単純にある位置から別の位置への動きであってもよい。コントローラーをあちこち移動させると、モーション信号が604で取り込まれ、ある手段によって少なくとも2つの別々のモジュール(レコグナイザーメーカー605と実行中のアプリケーション607)に並列で結合される。

30

【 0 0 9 4 】

605では、モーション信号は、好ましくは処理されたバージョンであり、新たなモーションレコグナイザーを作成するか又は既に生成されたモーションレコグナイザーをアップデートするために使用される。ユーザーによってなされた新たなタイプのモーションがあり、それに応答するモーションレコグナイザーがない場合、存在しているレコグナイザーが新たなタイプのモーションを認識するようにアップデートされるか、又は新たなモーションレコグナイザーが適宜作成される。エンドユーザーによってなされたモーションに応答するモーションレコグナイザーがある場合、そのモーションレコグナイザーは、そのモーションによりよく応答するようにアップデート又は強化される。

40

【 0 0 9 5 】

606では、アップデート及び新たに生成されたモーションレコグナイザーが記憶される。一実施形態によれば、新たに生成されたモーションレコグナイザーは、作成/アップデート605と並列で実行中のアプリケーション607に対して609でロードされ、元々ロードされていたモーションレコグナイザーと結合され、進行中のモーション認識プロセスが修正される。ユーザーは、ディスプレイを見たり、それ以外の方法でアプリケーションと情報をやりとりしながら、コントローラーの移動を続ける。プレーヤーのモーションは、そのときにロードされているモーションレコグナイザーを用いて、607で認識される。アプリケーション及びレコグナイザーメーカーの進展につれて、ディスプレイは、608でアップ

50

デートされる（詳細は、後述する。）。一実施形態では、適切なフィードバックがモーションレコグナイザーを構築する上で本質的である。

【0096】

一実施形態によれば、実行アプリケーション607は、認識RTL(例：図3の307)を組み込むか又はそれ以外の方法で認識RTLへのアクセスを有する。607での実行アプリケーションは、モーション応答アプリケーションとして動作する。つまり、このアプリケーションは、モーション信号を受け取り、組み込みモーション認識機能からモーション認識信号及び他の情報を受け取り、このような情報に応じて608でディスプレイをアップデートする。

3 エンドユーザー用アドホックパーソナライズ化レコグナイザーメーカー

10

【0097】

この発明の一実施形態によれば、一般人のメンバー、言い換えると、当業者では無い者がアドホックパーソナライズ化クロス-アプリケーションモーションレコグナイザーを作成することが可能になる。ダイナミックヒューマンモーションに対して、特定の方法で実行しなければならない予め定義されたモーションの集合を認識可能なロバストモーションレコグナイザーを構築することは、非常に困難なタスクであり、典型的に、かなりの背景知識及びかなりの時間及び労力を必要とする。予め定義されておらず且つ独自の及び予め定義されていない方法で実行されるモーションに対してロバストモーションレコグナイザーを構築することは、現在の技術水準を超えており、大部分の当業者は、その見通しによって圧倒されるし、一般人のメンバーについては言うまでもない。この発明の好ましい実施形態は、一般人のメンバーがこれを現在、正確に行うことを可能にする。

20

【0098】

アドホックパーソナライズ化レコグナイザーを作成する意欲と能力があるエンドユーザーに対して、レコグナイザーメーカーは、次の機能を有するように構成されている。(a) エンドユーザーモーションデザイン問題検出及びフィードバック、(b) ホスト計算ユニット上の高速近似クラシファイアー構築、及び(c) レコグナイザーパラメーターのデータ-主導型決定。これらの機能を提供することの詳細は、「アドホックモーションレコグナイザーを作成することに対するエンドユーザー制御」において後述する。

【0099】

次に、モーションレコグナイザーは、認識RTL(例：図3の307)と共に、次の機能を有するように構成されている。(a) いつでも最良推測モーション認識、(b) アプリケーション応答性に対するいつでも曖昧性除去ツリーフィードバック、及び(c) ロックインベースの採点。これらの機能の詳細は、「モーション検出アプリケーションに対して即座のフィードバックを提供すること」において後述する。

30

【0100】

次に、モーションレコグナイザーは、広い範囲の入力で、生成可能である。これには、(a) 3Dモーションデータから6D追跡パスに対するボタンプレスにまで及ぶ入力タイプ及びデバイス(input types and devices ranging from 3D motion data to button presses to 6D traced paths)、(b) デュアルモーション、ジョイントモーション及び他の認識様式を含む出力応答の対応する広さ、及び(c) 認識ランタイムライブラリがデバイスに依存しないように、モーション信号に対してデバイスに依存しない抽象化(a device-independent abstraction)を提供する仕様インターフェース。これらのの詳細は、「汎用認識」において後述する。

40

3.1 アドホックモーションレコグナイザーを作成することに対するエンドユーザー制御

【0101】

この発明の好ましい実施形態は、アドホックモーションレコグナイザーを生成する機能を、エンドユーザーに対して直接提供する。これは、開発時のレコグナイザーメーカーを、開発時のソフトウェアの全ての機能を有するランタイムエンドユーザーアプリケーション内に構成することによって行われる。レコグナイザーメーカーのユーザーがエンドユ

50

ーザーであり、プロのアプリケーション開発者でない場合には、かなりの差異が生じる。例えば、より少ない種類の人々から、より少ないトレーニングデータが得られること、エンドユーザーが許容する制御がかなり少ないこと、一般にホスト計算プラットフォームの能力が低いこと、及びエンドユーザーが存在している状態でレコグナイザーの作成を行う必要があること（オフライン「バッチ」スタイルトレーニングは、現実的な唯一の代用物とするには不利益が多すぎる。）。より多くの背景知識、スキル及び時間を有する開発者が以前は制御していたパラメーターは、今は、データから直接計算される。例えば、モーションレコグナイザーは、所定のモーションレコグナイザーにマッチしないモーションに対しては、「未知の」又は「未決定」をリターン可能でなければならず、アドホックモーションレコグナイザーについても、認められているモーションの予め定義された集合なしで、大部分のエンドユーザーが「正しいと感じる」方法でそのようにしなければならない。ホスト計算プラットフォーム上に存在しているアクティブレコグナイザーの即座又は一定の構築、又はチューニング-ベースの修復についての新たな方法の説明も行う。

10

3.1.1 エンドユーザームーブデザイン問題検出及びフィードバック

【0102】

スキルのあるモーションコントロール開発者は、多くの効果的なフィードバックと、柔軟に収集している多くのツールからの恩恵を得る傾向がある。このツールには、一連のデバッグ機能、モーションレコグナイザーバイアスの微調整のための制御パラメーター、及び種々のトレーニング及びテストセットを作成するためにモーション例のサブ集合の管理を助けるツールが含まれる。しかし、スキルのないエンドユーザーにとって、このデバッグ支援の多くのコレクションや制御ノブは、有害である。エンドユーザーにとっては、パーソナライズ化アドホックレコグナイザーの構築には、2つの形態のフィードバック（ムーブ混同検出及びフィードバック、及び思い出させるためのビジュアルトラッキング）が非常に望ましく、かつ十分である。

20

【0103】

ムーブ混同検出及びフィードバック。図3に関して、エンドユーザーがレコグナイザーメーカー305を用いてレコグナイザー306を構築する間に、自動的な方法によって取り扱うことができない唯一のタイプのエラーは、ムーブが下手にデザインされていて、変える必要があるときである。これは、エンドユーザーによるムーブの力が弱すぎてコントローラー302（つまり、その中にあるセンサー）が拾い上げることができなかつたり、ムーブが短すぎて検出可能なモーション信号304が生成されなかつたり、302中の内部センサーが「レイルする」、つまり振りきれられるほど激しいモーションである場合に起こる。このような場合には、検出とその後のフィードバックの両方が直接的である。エンドユーザーは、自身のムーブデザインを変えることによって問題を修復しなければならない。

30

【0104】

下手なムーブデザインと関連したより困難な問題は、2つのムーブ（例：2つのほぼ垂直な剣斬撃：斬撃180及び斬撃190）がモーション信号空間中で互いに近すぎて問題になる場合に生じる。このインパクトは、いくつかの方法がある中の一つで現れ得る。

【0105】

まず、斬撃180が斬撃190として誤分類されることが多いという点で、これらのムーブが互いに混同されている。両方のムーブが互いに混同されることが多い場合には、誤分類は、対称的である。斬撃180が斬撃190と混同されることが多いがその逆が成り立たない場合には、誤分類は、一方的になる。この場合、一実施形態によれば、トレーニングセットのサブ集合から混同マトリックスを構築することによって、レコグナイザー構築（プロセス400）中に検出が行われ、それを処理して、ホットスポットを探す。

40

【表 1】

モーション	allCircle	allCuckoo	allSquare	allHari	allJab
allCircle	660	5	13	0	42
allCuckoo	0	425	1	2	3
allSquare	2	10	520	4	9
allHari	4	6	0	385	25
allJab	1	10	4	26	299

【0106】

10

例示的混同マトリックスは、上記のものである。例えば、allJab行は、320回のテストジャブ(test jabs)のうち、299回が正しく認識され(allJab列)、1回が円(a circle)として、10回がカックウダンス(a cuckoo dance)として、などと誤って認識されたことを示す。一つのホットスポットは、allCircle行allJab列であり、これは、allJabが42回の円をジャブとして誤って(及び非対称的に)分類したことを示す。allJabについてのスラックを低減することは、これを解決する助けになるであろう。別のホットスポットは、allJab及びallHariのセルである。混同マトリックスエントリー(25及び26)は、これらのムーブが互いに混同されていることを示す。一実施形態では、エンドユーザーに対するフィードバックが、ムーブ(allJab及びallHari)は、高い信頼性で区別可能ではなく、このうちの一方を変更すべきであるという警告としてここで示される。

20

【0107】

第2に、さらに致命的なものであるが、ムーブが互いに混同されないかもしれないが、その代わりに、そのプロトタイプについての分類距離が小さくなりすぎて、そのムーブの実行を成功させることが非常に困難になる場合がある。この場合、検出は、レコグナイザーメーカープロセス400中に起こる。一実施形態では、距離尺度についての全体の予測される分類距離分布が402で計算され(a gross expected distribution of classification distances for the distance measure in 402 is computed)、トレーニングセット中の全ての対間距離に対する全体平均も計算される。最後に、クラス毎の分類距離の平均が計算され、全体の分布と全体平均の両方と比較される。1又は複数のエンドユーザームーブの平均プロトタイプ距離が予想外に小さい場合、警告が作成され、待ち行列に入れられ、ムーブデザインを変更する必要があることがエンドユーザーに提示される。

30

【0108】

思い出させるためのビジュアルトラッキング(Visual tracking for reminding)。典型的な使用のケースでは、エンドユーザーは、数セッションに渡って、レコグナイザーメーカー305、モーションレコグナイザー306及びモーション検出アプリケーション308と情報をやりとりするが、これらのセッションの間は、数時間、数日又は数週間離れている場合がある。残念なことに、多くの人々は、詳細なフィジカルモーションを記憶することが上手ではない。例えば、エンドユーザーは、ハンドセット上のブラウジングアプリケーション用のモーションレコグナイザーを月曜日に作成する。金曜日にそれを使用しに戻ってくると、彼らは、コントローラーを正確にどのように握っていたのか、又はインターネット

40

【0109】

一実施形態による、ここで説明するアプローチは、2つである。最初の方法は、基礎となるモーションレコグナイザーを連続的に修正することによって、ユーザーメモリー(又は、ユーザーメモリーがないこと)を無関係にすることである。適切な状況では、ユーザーがムーブの実行を試みて、連続して二度不成立になれば、最後のムーブをトレーニングセットへ追加して現在使用中のモーションレコグナイザーを再構成するオプションがプロンプトされる。このプロンプトには、最後のムーブについて、最も可能性が高いクラスで始まり最も可能性が低いクラスで終わるソートされたリストが含まれる。各クラスの可能

50

性は、プロセス500の509で計算された現在の最良予測と比較し、各クラスについてのスラックを大きくした場合に最良のフィットとなるクラスを選択することによって決定される。エンドユーザーは、このモーションをトレーニングセットに追加し、自身が引き出したラベルを単に選択することによって再構成することに同意する。

【0110】

第二の方法は、ユーザーの実際の追跡された経時的な動きのビジュアル表示でエンドユーザーに思い出させることである。このような表示を提供することは、追跡情報を提供するほどにモーション信号304が豊富であるシステムにおいて実現可能である。例えば、信号が映像を含むか、又は信号がコントローラ302の経時的な位置及び方向を追跡するのに十分な慣性信号を含むシステムである。このような場合には、エンドユーザーが307にある認識RTLにクエリーを行うとき、エンドユーザーの以前のモーションと、最も可能性が高いクラスのソートされたリスト中の最も近いプロトタイプ(the closest prototype in the sorted list of most likely classes)の両方が追跡された物体として再構築され、計算ユニット309のディスプレイ上に並べて表示される。2つのムーブ中の相違の各ポイントが強調され、以前のモーション例がどのようなものであったのかを思い出させるビジュアル手段がエンドユーザーに与えられる。再構築されたモーション追跡の形態そのものは、この発明に無関係であることは当業者にとって自明である。例えば、モーションコントローラを握っている手のように簡単なものであってもよく、矢を握っている、肉体的でない棒線画であってもよい。

10

20

3.1.2 高速近似クラシファイアー構築

【0111】

エンドユーザーアプリケーションがコマンドに応答することは、かなりの恩恵である。一実施形態では、ほぼ最適なレコグナイザーを作成し、最小限のCPU及びメモリーリソースを使用し、近似レコグナイザーをいつでもリターンすることができる(例：いつでも(計算中でさえ)妥当な結果をリターンすることができる)、次の3つの方法が使用される。

【0112】

いつでも応答できるオンライン構築(Online construction with any-time response)。レコグナイザーメーカー305は、例データの受信中にもオンラインで(非「バッチ」モードで)動作可能である。好ましい実施形態は、プロセス400のようなオンライン構築プロセスを使用し、新たなレコグナイザーが例単位で連続的に構築される。この粒度1のオンライン構築モード(granularity-one online construction mode)は、トレーニング用に得る可能性が高い全てのデータは、そのエンドユーザーとの一つのセッションで生じることが多いケースを当然取り扱う。このモードは、非常に望ましい。なぜなら、妥当なプラットフォーム上で、現在のデータを考慮した最高のレコグナイザーをエンドユーザーが要求し、いつでも受け取ることができるからである。

30

【0113】

アダプティブフィルタリング(Adaptive filtering)。システム300では、モーション信号304は、レコグナイザーメーカー305に結合される前に、アダプター310でアダプティブフィルタリングによって処理される。これによって、信号の興味深い部分のみが残される。慣性データに対する興味深い部分は、例えば、直線又は角加速度の相対的な大きさが、モーション中の隣接するサンプルから閾値を超えて変化するときや、1又は複数の軸の加速度の相対的な大きさが時間閾値を超えて変化したときや、又は最後処理されたサンプルが生成されてから比較的長い時間が経過したときを含む。興味深いもののコンセプトは、他のタイプのモーション信号についてもほぼ同じである。その利点は、(1)処理されたモーション信号は、典型的に、長さがはるかに短く(一実施形態では、5倍短くなる)、モーションレコグナイザーの作成及び使用の両方と関連した計算時間が低減されること、及び(2)信号の無関係な部分が分類前に除去されるので、分類精度が向上する、ことである。

40

50

【0114】

部分的距離計算。プロセス400で記載したように、最適なモーションレコグナイザー構築は、トレーニングセット中のモーション信号の間の全ての対間距離を必要とする。これを計算し、アクセスすることに、レコグナイザー構築の全メモリー及びCPU要求の99%が費やされる。好ましい実施形態では、分類精度に目立ったインパクトを与えずに、全ての可能性のある距離対(all possible pairs of distances)のほんの一部のみが計算される。大部分は、低コストで推測される。その結果生じる計算は、 $O(f(n/c)^2)$ である。ここで、 f は、アダプティブフィルタリングの後のモーション信号の平均長さであり、 c は、レコグナイザー中のクラス又はムーブの数である。典型的なケースでの利点は、エンドユーザー待ち時間(及びバッテリー消費)が、ケタ違いに低減されるからである。

10

【0115】

この方法(プロセス400の402-404で簡潔に説明した。)は、距離空間の次の共通の性質を利用する。つまり、AからBへの距離(つまり d_{AB})が小さく、 d_{AC} が大きい場合、 d_{BC} も大きい。レコグナイザーメーカー305がモーションA及びモーションBが同じクラス内にあり、モーションCが異なるクラスにあることを知っており、さらに d_{AB} が小さく、 d_{AC} が大きいことを知っている場合、レコグナイザーメーカーは、 d_{BC} を計算しないであろう。なぜなら、この計算は、A、Bのクラス又はCのクラスの何れに対しても、良好なプロトタイプ選択に無関係であると分かっているからである。

【0116】

この方法を使用する大きな障害は、時系列データに対してうまく働く大部分の距離関数は、距離空間の定義には十分に良好に振舞うとは言えない。従って、上記の性質に基づいてプロセス400の403及び404で使用される推測は、うまくいかない。具体的に、三角不等式($d_{AB}+d_{BC} \geq d_{AC}$)はうまくいかない場合がある。概念的に、その理由は、我々の距離計算のそれぞれは、実際は、高次空間(サンプル数×モーション軸数)で行っているものであり、それを単一の数に単純化しているものであるからである。

20

【0117】

一実施形態によれば、部分的距離計算のための方法では、三角不等式ではうまくいかない可能性が高い境界付近の対間距離の計算を十分に追加することによって、この欠陥を修復する。これによって、高い確実性で近似的な正しい結果が高い確率で得られる。

【0118】

その結果生じる方法は、次の通りである。(1)所定のクラス内の全ての対間距離の一部を計算する。(2)クラス毎にクラスターの小さい集合を撒き散らし、クラスター重心を選択し、その後のトレーニング例をクラス中の最も近いクラスターに割り当てるか、又はどれも十分に近くなければ新たなクラスターを作成する。これは、チェックされる各クラスターについて、クラスター重心と例の間の少なくとも一つの対間距離計算を必要とする。(3)全てのクラスについて、全てのクラスター重心間の全ての対間距離を計算する。(4)要求に応じて、それぞれのクラスター重心距離を使用することによって全ての他の対間距離を近似する。クラスター境界が交差するか又はほぼ交差する場合、これは、三角不等式ではうまくいかない可能性が高いことを示唆している。代表距離が十分に大きくなって三角不等式でうまくいかないことが多い場合、2つのそれぞれのクラスターのメンバーの間の距離の計算が追加される。この方法は、プロトタイプ選択が準最適になる場合があるという犠牲の下で、必要な対間タイムワープ距離計算の大部分を制御しながら除外ことに成功する。

30

40

3.1.3 レコグナイザーパラメーターのデータ-主導型決定

【0119】

一実施形態では、レコグナイザー構築には、3つのパラメーター(スラック、キャパシティ、及びスタート決定)が使用される。例えば、US特許出願#11/486,997での好ましい実施形態では、スラックとキャパシティの両方が開発者に利用可能なパラメーターであり、トレーニングセット用の全てのモーションは、ボタンプレスによって境界が定められる

50

。従って、閾値を設けてモーションスタートを検出する必要がない。レコグナイザー構築の技術的な詳細についてエンドユーザーとの不要な交流を除外するために、これらのパラメーター自動的に設定することには利点がある。

【 0 1 2 0 】

自動スラック設定。図5に関して、スラックは、各プロトタイプについて、分類許容値を制御するために、プロセス500の503で、分類距離に対する乗数として使用される。ユーザーデータ、アプリケーション及びムーブセットの各組み合わせにより、分類距離の最適な修正が様々なものになる。一実施形態では、クラス毎のスラックは、次のファクターによる最適化に基づいて、図4の411で、自動的に計算及び設定される。1)トレーニングセットの種々のサブ集合について、全体の分類率を最大化する。2)クラス毎の分類率の差異を最小化する。3)テストモーションデータの第2の関連のない集合に基づいて、許容可能な未決定率を維持する。4)クラスの間混同を均一にする(以下の「混同マトリックス」を参照)。ステップ1及び2は、プロセス400中で詳細に説明した通りである。

【 0 1 2 1 】

一実施形態では、ステップ3は、プロセス400でのレコグナイザー構築中に実行される。図4で説明したように、プロトタイプは、最も性能が悪いムーブ(the worst performing moves)に最初にフォーカスするように、不均一に、モーションレコグナイザーに追加される。このフェーズ中に、最初はタイムワープ距離関数から導き出されるバイアスに基づいて各プロトタイプの分類距離が確立され、この分類距離は、より多くのデータが処理されるにつれて分類能力によって上書きされる(During this phase, each prototype's classification distance is established based initially on a bias that is derived from the time warp distance function, and overridden by classification performance as more data is processed.)。未決定のテストセットについての新たなモーションレコグナイザーを用いた未決定分類率が許容可能なプリセット範囲外である場合、クラス毎のスラックは、全体の認識が許容範囲になるように増減して調節される。使用されるテストセットは、leave-one-out法で、トレーニングセットから直接構築可能である。例えば、新たなトレーニングセットは、一タイプのムーブに対応するデータの一つのサブ集合を除去することによって構築される。レコグナイザーが作成され、除去されたムーブがテストセットとして実行される。平均すると、そのムーブは、高い確率で未決定として分類されるはずである。

【 0 1 2 2 】

一実施形態では、ステップ4は、プロセス400の418での混同マトリックスの計算を含む。性能が悪いクラスの個々のクラス毎のスラックは、徐々に増加するように調節され、その後、テストされる。一方、性能が悪いクラスと頻繁に混同されるクラスのスラックは減少させる。クラス毎の差異が許容範囲になるか、又は全体の分類率が許容範囲外になるとこのフェーズは終了する。

【 0 1 2 3 】

要約された混同マトリックスの例は以下のものである。これは、「allGuidedDrop」の偽陽性率が高いことを示す。これは、これらのプロトタイプの分類距離が大きすぎるので、自動的にスラックを低いクラスに設定することによって補償すべきであることを示している。

【表 2】

モーション	例	偽陰性	偽陽性
allCircle	720	11.4%	5.3%
allCuckoo	430	10.2%	8.4%
allGuidedDrop	540	6.3%	35.9%
allHari	400	10.8%	8.8%
allJab	320	9.4%	6.2%

【 0 1 2 4 】

自動キャパシティ設定。キャパシティは、所定のモーションレコグナイザーに許されたプロトタイプの数に線形的に関係している。モーションレコグナイザー中のプロトタイプの数が多いほど、より多くのメモリー及びCPUが要求される。大ざっぱに言えば、キャパシティの数が0から無限大に増えるにつれて、所定のアプリケーションの分類率が素早く上昇し、横ばいになり、最終的に、落ち始める。なぜならレコグナイザーがトレーニングデータにオーバーフィットし始めるからである。キャパシティは、所定のモーションレコグナイザー達成可能な認識率(従って認識システム全体の成功又は失敗)を直接的に定めるので、キャパシティの制御が必要である。レコグナイザー構築に関する技術的な詳細についてエンドユーザーとの不要な交流を除外するのに有益であるので、キャパシティは、自動的に設定される。

10

【 0 1 2 5 】

好ましい実施形態では、プロセス400の405と409で説明したように、プロトタイプは、分離度(DOS)と呼ばれる独自の計算尺度(a unique computational measure called degree of separation)に基づいて選択される。所定の候補プロトタイプが同じクラスからの例に対して与えるDOSは、異なるクラスからの例でそのプロトタイプにより近いものがなければ0であり、異なるクラスからの例でそのプロトタイプにより近いものがN例あればNである。ある候補に対して、その候補のDOSは、所定のクラス中の全ての他の候補に対する合計によって与えられたDOSである(the imparted DOS summed over all other candidates in a given class)。これは、所定の分類距離を有する候補プロトタイプが行うであろう正確vs.不正確な分類の尺度を計算するのに、最適で、高速な方法である。一実施形態では、キャパシティは、プロセス400の409で計算された第1幅数と第2幅数の間の中間に自動的に設定される(In one embodiment, capacity is automatically set halfway between the first and second width number as computed at 409 of process 400.)。示唆されるように、プロトタイプは、キャパシティが正確にモーションレコグナイザーの状態を反映するように、確定作業中に除去される場合がある。

20

【 0 1 2 6 】

自動スタート決定(Automatic start determination)。スタート閾値は、許容値であり、この値を超える場合に、コントローラ302が動いていると判断される(例:図1bのポイント111及び114)。この時点で、分類対象のモーションが始まったと仮定される。スタート閾値は、モーションの開始を指示する外部信号(例:ボタンプレス、又はアプリケーション内の「Go!」信号(つまり「ボタンなしのデータ」))がない場合に、極めて重要である。このような場合には、モーションの探索をいつ開始して、いつ終了するのかを決定するために、受信モーション信号ストリームをセグメントに分割する必要がある。アプリケーション中で、モーションがいつ開始したのかを検出するためのスタートボタンイベントが要求されないことには利点がある。なぜなら多くのエンドユーザーは、それをややこしくて不自然であると感じるからである。

30

【 0 1 2 7 】

好ましい実施形態では、トレーニングセットからスタートクラシファイアーを構築することによって、スタート決定が計算される。トレーニング例の記憶されたモーションが2、3の追加のプロパティを有している。最初に、モーションの正式なスタート及びエンドの辺りのデータのエンベロップ(an envelope of data around the official start and end of the motion)が記録される(例:図1b中の114の前及び116の後のサンプル)。第2に、モーションの正式なスタートが、トレーニングのためのデータをcollectingしている間にのみ現れる追加のプロセス(例:ゲーム内の、「Go!」のようなトリガーイベント)によって設定される(the official start of the motion has been set by an additional process that shows up only while collecting data for training, such as an in-game triggering event like "Go!").多くのスタートクラシファイアーが可能であり、一例は、力閾値を検出するものであり、その閾値を超えた場合に、ムーブが正式に開始される。好ましい実施形態では、スタートクラシファイアーは、正式なモーションデータ(例:加速度マ

40

50

イナス重力の力)からエンベロープを区別するために使用される、エンベロープの特徴の辺りに作成される(the start classifier is built around features of the envelope that are used to differentiate envelope from official motion data, for example, force of acceleration minus gravity)。プロセス500のようなモーション認識の間において、このプロセスの重要な特徴は、「スタート」は、モーションが正式に開始された、まさに最初のフレームについて検出される必要がないことである。むしろ、データの周りのエンベロープ(envelopes around the data)が追跡されているので、特徴は、正式なスタートフレームの前後のいくつかのサンプルを追跡することができ、「スタート」が起こった数サンプル後に「スタート」の判定を行うことも許容される。一実施形態では、モーション信号ストリームを構成する、この「スタート」及び「エンド」(つまりセグメント化していること)は、ムーブのスタートのみを明示的にマークすることによって達成される。なぜなら、レコグナイザー自身が停止検出器を提供しているからである。

10

3.1.4 例

【0128】

この発明は、エンドユーザーの観点から多くの形態を取ることができる。

【0129】

例えば、モーション認識は、計算ユニット309上の全てのアプリケーションが利用するサービスとして提供してもよく、全てのモーション検出アプリケーションに別々に組み込んでもよい。

20

【0130】

例えば、レコグナイザーメーカーは、計算ユニット上の別個のアプリケーションとして作成してもよく、又はモーションコントロールサービスレイヤー205内に組み込んでもよい。

【0131】

例えば、レコグナイザーメーカーは、計算ユニット上で、常にバックグラウンドで動作するようにしてもよく、各セッションの後にディスプレイの制御を取得し、別のアプリケーションが終了したときに、関連するレコグナイザーについてのフィードバックをアップデートするようにしてもよい。

【0132】

この発明で使用可能な多くのモーション検出アプリケーションがある。

30

【0133】

例えば、スマートフォン上のアプリケーション選択は、計算ユニット208上のアプリケーション環境に組み込まれたモーション検出アプリケーションであってもよい。エンドユーザーは、電話内の各アプリケーションにアクセスするために、それぞれ異なるムーブの例を与えることができる。例えば、配偶者に電話するために空中にハートを描いたり、又はGoogleサーチをするために円を書いてブラウザを呼び出すなどといったことである。

【0134】

例えば、ユーザーがコントローラーを近くに引き寄せたり、顔から離したりすることを認識することによって、ズーム調整することができる。

40

【0135】

例えば、単純にアプリケーションと共に出荷された元来のレコグナイザーをエンドユーザーが作成したものに置き換えることによって、計算ユニット上のゲームに対して、新たなモーションコントロールを追加することができる。

【0136】

例えば、テレビジョン上でのブラウジングは、チャンネル76を選ぶ際に7と6を入力する代わりに、エンドユーザーがお気に入りのTVチャンネルに対してお気に入りのモーションを作成することによって行うことができる。

3.2 モーション検出アプリケーションに対して即座のフィードバックを提供すること

50

【 0 1 3 7 】

エンドユーザーは、アプリケーション（例：コンピュータービデオゲーム）がエンドユーザー入力（例：ボタンプレス、又はモーション）に応答して即座の及び連続的なフィードバックを与える能力を有することを期待する。モーション検出システムがこの要求を充足させようとする際の困難な点は、典型的なモーション（例：テニスでのクロスコートフォアハンド）が、完全に実行するには、数百フレームのデータを必要とする（例：図1bは400フレームを必要とする。）が、毎秒60フレームで動いているゲームは、5-8フレーム内にこのモーションのフィードバックの提供を開始する必要があることである。図2の認識RTL203がフィードバックに用いるアプリケーションに対して認識ラベルを提供する前に、全てのデータの入力を待つことは明らかに不十分である。多くの現存するモーション検出アプリケーションがこれを回避するためにすることは、アドホックモーションコントロールを扱うことを避けることである。例えば、ムーブ制御システムは、モーションが検出されるとすぐにトリガーすることができる。アドホックパーソナライズ化モーションレコグナイザーによって可能になったモーションコントロールを有するモーション検出アプリケーションを使用するエンドユーザーに対して即座のフィードバックを与えることができることには、明らかな恩恵及び利点がある。

10

3.2.1 いつでも最良推測モーション認識

【 0 1 3 8 】

「いつでも最良推測」("Anytime best guess")は、モーション検出アプリケーションが、モーション信号の一部又はプレフィックスが観察された後に（例：図1bでは、114と116の間）、プロセス500の509において、現在の最良推測予測を要求して、受け取ることを意味する。確かさ尺度は、現在の部分的モーションが現在のモーションレコグナイザー306の各クラスのメンバーである見込みを予測することによって、計算される。確かさ尺度は、ラベル及び確かさのランク付けされたリストを含む、現在の最良予測の重要部分である。確かさ尺度は、部分的流入モーションデータから、レコグナイザー中の各プロトタイプのうちベストフィットするものに対して現在のタイムワープ距離の関数として計算され、プロトタイプを通じた進展によって重み付けがなされる(The confidence measure is computed as a function of the current time warp distance from the partial incoming motion data to the best fit to each prototype in the recognizer, weighted by progress through that prototype.)。

20

30

【 0 1 3 9 】

これを達成する上での大きな障害は、プロトタイプリストが大きすぎて現在の最良予測を最新の状態に保つことが実現可能でない場合があることである。一実施形態では、これを克服する方法は、プロセス500の508で行われる早期カットに基づく。走行タイムワープ距離が非常に長くなり、次の分類に参加しそうにないプロトタイプは、現在の受信モーション信号の残りについては検討から外す。具体的に、蓄積コスト（例：時間及びリソース）は、プロトタイプ及び信号の長さに対して単調的に増大する。現在の蓄積コストがプロトタイプと信号の間の閾値を超え、プロトタイプの分類距離よりも大きくなると、たとえ、信号の残りの部分が完全にマッチした場合でも、そのプロトタイプがその信号を分類することはない。プロトタイプの残りについてのその後のコストが0である取り扱いことは、過度に慎重である。その代わりに、プロトタイプの残りのサイズに基づいて、ほぼ完全にマッチした場合のコストを追加し、蓄積コストとこの追加分の合計が分類距離内でない場合にはカットが行われる。つまり、蓄積コスト+残りのコスト>分類距離が成り立つ場合に、早期カットが行われ、プロトタイプが除去される。

40

【 0 1 4 0 】

早期カットの重要な恩恵及び特徴は、これが、はるかに多くのプレーヤーに対して、いつでも最良推測予測を可能にすることである。時間が経過するにつれて、レコグナイザー作成及び認識処理の速度が上がる。なぜなら、残っているプロトタイプが減少し続けるからである。例えば、200のアクティブプロトタイプでモーション認識を開始するレコグナ

50

イザーは、最後には、生き残っているプロトタイプが30のみになる。このことは、認識の最後には最初と比べて、認識に費やされるCPUリソースが1/7になることを意味する。1つのアクティブモーションデバイスが認識されるシステムにも役立つが、同時に複数のデバイス302を認識するときには極めて有益である。

【0141】

例えば、Nintendo Wiiは、同時にアクティブになる8つのモーションコントローラーを有している。大部分のケースでは、これらのコントローラーは、異なるステージで異なるモーションの実行に用いられる。認識ランタイムライブラリ307は、最初は、1つのモーションコントローラーを処理し、最後には、第2のコントローラーを処理し、中間には、残りの6つのコントローラーを処理する。早期カットを行うと、認識RTL307は、2つ又は3つのコントローラーをコンスタントに且つ高い信頼性で（高い測定可能な確率で）管理するリソースコストで、8つ全てのコントローラーの管理を行う。

10

3.2.2 曖昧性除去ツリー

【0142】

いつでも最良推測ラベルは、多くのモーション検出アプリケーションに対しては十分であり、使いやすい。しかし、ムーブが早期に混同された場合には、うまくいかない場合がある。エンドユーザが空中に「2」を描いたときと「3」を描いたときで慣性コントローラー302のモーション信号304が似ている場合を考える。この場合、モーションの最初の50-80%については、モーションが2と3のどちらであるかをデータから判断することはできない。このような場合、アプリケーションが「2」や「3」のアニメーションをタイムリーに開始することはできない。なぜなら、これらの区別ができないからである。

20

【0143】

しかし、このことは、モーション検出アプリケーションに対する有益なフィードバックがないということを意味しない。実際に、アプリケーションは、ジョイント「2-3」ムーブのアニメーションを即座に開始可能であり、そうすべきであり、データが揃ってから「2」又は「3」で終えて曖昧さをなくせばよい。以下の実施形態の重要な特徴は、このような「ジョイントムーブ」又は「混同セット」情報をアプリケーションに提供して、ユーザーに対してタイムリーに適切なフィードバックを提供することである。

【0144】

一実施形態では、曖昧性除去ツリー(a disambiguation tree)がプロセス400のフィードバック418の一部として作成され、プロセス500の512でのクエリーのために利用可能である。内部的に、区別可能なムーブを有するモーションレコグナイザーに対する曖昧性除去ツリーは、有向非巡回グラフ(a directed acyclic graph)である。スタートノード(the start node)(つまり、ルート(root))は、0%完了の状態であり、全てのムーブが混同されている。なぜならムーブがまだ始まっていないからである。葉ノードは、ムーブが確実に決定される完了パーセントでは、単一のムーブである。例えば、数値0-3は、0~8%完了の状態では、全てが混同されているが、8%において「1」が枝分かれする。「0」は、20%完了で「2、3」から分岐し、「2」と「3」は60%完了まで混同され続ける。確かさのレベルが異なる多くのムーブツリーが作成可能である。例えば、あるツリーでは、95%の確かさで、ムーブの曖昧性が除去され、別のものでは、80%の確かさで、ムーブの曖昧性が除去され、非葉ノードから分岐される。512でクエリーが行われたとき、その応答は、現在の状態(例:「2、3」ムーブ)を考慮した、最良推測「ジョイント」ムーブ(the best guess "joint" move)である。

30

40

【0145】

この情報には、いくつかの追加の利点がある。例えば、これは、モーション検出アプリケーションがフィードバックとしてエンドユーザーに対して提供し、ユーザーがムーブの集合をどのように修復すべきかについて十分に理解するのを助けるために、利用可能である。例えば、モーション入力に対して即座の応答を希望するエンドユーザーは、厳密にどのムーブのデザインをやり直す必要があるのかを知ることができる。なぜなら、曖昧性除

50

去ツリーは、厳密にどのムーブがどの位の長さ混同されているのかについての情報を提供するからである。

【 0 1 4 6 】

アプリケーションデザイナーは、ムーブが混同されたときにでも即座にアニメーションを開始し、ユーザーと連携してユーザーがアプリケーションの早期のアニメーション決定にフィットするアドホックモーションレコグナイザーを作成するのを助けるようにするのに、事前に作成したモーションレコグナイザーとアドホックレコグナイザーのどちらと共に曖昧性除去ツリーを使用してもよい。

3.2.3 ロックインベースの採点

10

【 0 1 4 7 】

エンドユーザーとモーション検出アプリケーションの両方にとって望ましいフィードバックの第3の形態は、現在のモーション信号がモーションレコグナイザー中のムーブにどの程度マッチしているのかについてのスコア又は尺度(a score or measure)である。この情報は、エンドユーザーの向上と記憶を助け、アプリケーションがユーザーのパフォーマンスのスコアを付けることを容易にする。単純な実装は、受信モーション信号を最良のプロトタイプにマッチさせ、そのプロトタイプの分類距離内で現在のモーションがどの程度離れているのかについてのパーセントをリターンすることである。この方法は、問題がある。なぜなら、エンドユーザーがムーブを行う度に異なるプロトタイプが採点の基礎になり、そのため、ユーザーが前回のプロトタイプに近いづいているかどうかとはほとんど関係なく、スコアは、前回の試みから増減するからである。このため、重要な情報を失っている。エンドユーザーに対してより安定した採点能力を提供することには利点がある。

20

【 0 1 4 8 】

一実施形態では、エンドユーザーに思い出させ、エンドユーザーを鍛えることを特に目指しており、アプリケーション206は、エンドユーザーがよりよく実行することを希望するムーブを選択するように、エンドユーザーに要求する。アプリケーションは、次に、このムーブについて2、3回の試みを行うよう、エンドユーザーに要求し、そして、これによって、これらの試みに対して最も近いプロトタイプを見つける。このプロトタイプを「ゴールデン」プロトタイプと称する。この時点から、アプリケーションは、ユーザーがムーブを実行するガイダンスセッションに入り、各実行の後、アプリケーションは、単一のゴールデンプロトタイプに基づいてモーションを採点する。

30

3.2.4 例

【 0 1 4 9 】

例えば、コンピュータービデオゲームアプリケーション又はモバイルゲームアプリケーション206は、いつでも最良推測分類を使用して、エンドユーザーモーションに応答して即座にアニメーションを開始することができる。

【 0 1 5 0 】

例えば、曖昧性除去ツリーは、特定のムーブの集合についてアニメーションを開始しても安全であり且つ単一のムーブにコミットしても安全である、できるだけ早い時点をアプリケーションに伝える(the disambiguation tree tells the application the earliest point in time when it is safe to begin animating for a specific set of moves, and when it is safe to commit to a single move.)。

40

【 0 1 5 1 】

例えば、最初に混同されたムーブは、同じスタートを共有するゲーム内アニメーションへ翻訳すべきである。アプリケーションは、エンドユーザー及び曖昧性除去ツリーの助けを借りて実行を可能である。

【 0 1 5 2 】

例えば、ロックイン採点は、コンピュータービデオゲームアプリケーション又はモバイルゲームアプリケーション206がエンドユーザーがどれくらい上手にムーブを行っている

50

のかを採点するために、使用可能である。最初にエンドユーザーに2、3の「予行演習」をさせて、ゴールデンプロトタイプを選ぶ。

【0153】

例えば、曖昧性除去ツリーは、コンピュータビデオゲーム又はモバイルゲームアプリケーション206が早期の「スタート」アニメーションをいつ再生して、混同されたムーブに対して中間アニメーション(intermediate animations)をいつ開始するのが有用であるのかを特定することができる。

3.3 汎用認識

【0154】

本発明は、エンドユーザー用のアドホックパーソナライズ化モーションレコグナイザーに関し、従って、意図によって又は実装によって、手持ち式コントローラー上の内蔵の慣性センサーからのモーション信号304に特に限定されない。モーションレコグナイザーは、広い範囲の入力に適用可能である。利用可能なモーション信号に追加の独立したデータストリームを追加することは、認識の有用性を高める。例えば、ヒューマンモーションコントロールの主要な要素をキャプチャーする完全なモーションコントロールシステムは、それぞれの手の中にあり且つ各コントローラーの位置及び方向を追跡する手持ち式コントローラーからの慣性情報(例：3dジャイロスコープ及び3D加速度計)、同じコントローラーからのLED及びボタン及びジョイスティック入力、及びプレーヤーの頭、肩及び肘の位置及び方向情報の十分な集合を含む。全部で、12の慣性ストリーム、12のビデオ関連ストリーム、及びボタン、LED及びジョイスティックに対するデータのいくつかのストリームがある。多くのモーション検出アプリケーションは、エンドユーザーとのより広い帯域の形態での通信にアクセスできることを望ましいと考えるであろう。

3.3.1 様々な入力タイプ及びデバイス

【0155】

一実施形態では、レコグナイザーメーカー用のモーション信号304に変換されるデータを提供するデバイス302は、タッチ検出スクリーン上に2D又は3D描画するためのスタイラス又はフィンガー、手持ち式コントローラー上のボタン、d-パッド、トリガー及びアナログスティック、手持ち式コントローラー、ビデオカメラ、スケール、マイクロフォン、及び他のデバイスに組み込まれ且つヒューマンモーションの種々の成分を追跡可能な内蔵の慣性センサーを含む。これを達成する大きな障害は、類似した認識「感覚」を達成するために、認識を行う種々のデータタイプをどのように処理すべきか、種々のストリームをどのように登録すべきかである。

【0156】

一実施形態では、図3の310では、全ての受信モーション信号が、早期処理フェーズ中に、擬似直線加速度、擬似角速度又は擬似ボタンプレスに変換される。例えば、直線加速度計の出力から擬似直線加速度へのマッピングは、1：1である。コントローラー上のアナログトリガーの出力から擬似直線加速度へのマッピングは、ほぼ1：1である。マイクロフォン出力から擬似角速度へのマッピングは、より複雑であり、周波数成分を分離することを伴う。マイクロフォンからのノイズ入力は、直線加速度又は角速度のコレクションとしておおざっぱに取り扱うことができる(周波数成分毎に1つ。この粗い近似は、多くのアプリケーション環境での多くの音及び喉の「ジェスチャー」の認識には十分である。)

【0157】

プロセス400中のレコグナイザーメーカー及びプロセス500中のランタイムRTLは、システム300で具現化されたように、両方が、モーション信号304を同じように用いる。各慣性関連、ビデオ関連及び位置関連ストリームがレコグナイザーメーカー又はランタイムRTLに送られる前に最初に速度又は加速度に変換される。一つの重要な恩恵は、位置データに基づいて認識を行うことを避けることである。位置データは、スタートポイントに対する

位置の変化として表しても、その変化が頻繁に起こりすぎるので、アダプティブフィルタリングが強調することができる興味深いポイントが埋もれてしまう。

【0158】

モーション信号304の上記の変換された慣性、ビデオ及び位置成分のいくつか又は全ては、次に、認識フレームワークを通過する。例えば、2つの手の中のコントローラーからの12の慣性信号は、トレーニングセットを構成する12成分モーションに構成される。プロトタイプは、プロセス400で説明したように、タイムワープ距離に基づいて選択され、モーションレコグナイザーの作成に使用される。次に、流入する新たな12成分モーション信号は、プロセス500で説明したように、そのプロトタイプに対するタイムワープ距離を計算することによってモーションレコグナイザーによって分類される

10

【0159】

残りの信号は、典型的に、ボタンプレス及びジョイスティックプッシュで構成される。ボタンプレス(アップ・アンド・ダウンパルス)はフィルターされることなく、及びその代わりにアダプティブフィルタリングのために「興味深い」時間ポイントをトリガーするために使用される。フィルタリングレベルでは、ジョイスティック入力、慣性入力とほぼ同じように扱われる。

【0160】

これらの信号は、モーションレコグナイザーの構築又は使用にタイムワープ距離計算(例:図4の402で説明したもの)が必要な時には、異なって取り扱われる。一実施形態では、ボタンパルスは、非常に二値的に扱われる。例えば、レコグナイザーで「a」キーが押された場合、受信ストリーム中で「a」が押されなければ、モーション信号の残りがよくマッチしていたとしても認識は失敗である。「a」の代わりに「b」を押した場合には、部分点は与えられない。

20

【0161】

さらに、距離メトリックが入力信号中のタイムシフトを見過ごす能力(従って、その名前がタイムワープである。)は、低減又は修正され、これらの信号が、同じ認識率を達成するために、実際のフィジカルモーションよりも注意深くマッチする必要があるようにする(the ability for the distance metric to overlook time shifts in the input signal (hence the name time warp) is tuned down and modified so that these signals need to match more carefully than the actual physical motions in order to achieve the same recognition rates.)。

30

【0162】

具体的に、一実施形態では、特定のタイプのモーション信号についてのタイムワープのインパクトを変化させるために、スラックに類似した概念が使用される。スラックは、クラスに特有の、分類距離の修飾子であり、モーションとプロトタイプを比較する際にモーションの認識を容易にしたり難しくしたりする。同様に、「弾性」("elasticity")は、モーション信号の一部の修飾子であり、モーションとプロトタイプを比較する際に信号を時間的に前又は後ろにシフトさせる相対コストを制御する。典型的に、慣性信号に対する弾性は、比較的高く、このことは、例えば、x加速度中のスパイク(a spike in x acceleration)は、タイムワープ距離スコアに大きな影響を与える前に、プロトタイプと流入モーションの間でかなりシフトされる。ボタンプレスに対する弾性は、典型的には、非常に低い。従って、このような混合モーション信号の場合には、タイムワープ距離関数は、1又は複数の成分で構成され、それぞれは、信号を時間的にシフトさせることに対する感度がおそらく異なっている。

40

3.3.2 認識出力様式

【0163】

特に入力が変化に富んでいる場合には、モーション検出アプリケーションにとって望ましいいくつかの認識出力様式(recognition output modalities)がある。そのベースラインは、モーションレコグナイザー306がユーザーの手持ち式慣性検出コントローラー302の

50

ダイナミックモーションを認識することである。一実施形態では、認識RTL307は、同時の独立モーション(「パラレルモーション」)、同時の従属モーション(「ジョイントモーション」)、及び静止ポーズを認識可能である。これらの全ては、モーション検出アプリケーションを扱うエンドユーザーにとって、望ましい機能である。

【0164】

パラレルモーションは、モーション信号304が2つ以上の別々のソース302からのものである場合である。例えば、一つのソースがエンドユーザーの左手の中の慣性検出コントローラーであり、一つが右手の中のコントローラーであり、及び第3がエンドユーザーの顔の位置及び方向である場合である。便利な認識様式(recognition modality)は、両手がい

10

くらかのモーションを行い、それと同時に頭が何か他のことをしている場合に認識を行うことである。例えば、エンドユーザーが頭を縦に振り、左手で四角、右手で円を描くモーションを行っている場合に認識を行うことである。モーションが同時に行われ、各モーションが基準に達している場合、認識RTLは、パラレルモーションを認識すべきである。一実施形態では、これは、3つの別々のモーションレコグナイザーを作成し、これらを同時に走らせることによって実行される(1つは左手用、1つは右手用、1つは頭用)。別の実施形態では、パラレルモーション認識は、同時のモーションに対して1つのレコグナイザーを有することによって実行される。同時のモーションは、パラレルモーションの一部と意図され、アプリケーションが組み合わせさせた結果を提供することを可能にする。

【0165】

ジョイントモーションは、2つ以上の別々のモーションソース302を含む。ジョイントモーション認識は、そのモーションが独立しては達成されないという点でパラレルモーション認識と異なる。針の目に糸を通すことを想像する。針を保持して、その目に糸を通すことに成功するには、両手を一緒に働かせなければならない。一方が針を持ち上げては落とし、他方が糸を通そうとしてもうまくいかないのは明らかである。例えば、ゲームアプリケーションでは、エンドユーザーは、スペシャルアタックを実行するために、一方の手でシールドを突き出し、他方の手で水平に斬りつけを行うことが要求される。タイミングが正しくなければ失敗する。一実施形態では、ジョイントモーション認識は、別々のソース302を組み合わせさせて一つのジョイントモーション信号にし、その組み合わせさせたストリームに対する一つのモーションレコグナイザーを作成することによって、達成される。従って、例えば、3D加速度計及び3Dジャイロスコープを有する2つのコントローラーは、認識システムの観点からは、実際には、一つの12Dコントローラーになる。

20

30

【0166】

静止ポーズが、第4の認識様式である。モーションのダイナミックパスには興味がない。その代わりに、エンドユーザーの静止位置がその焦点である。この機能を提供することは、率直であり、単純に、モーション信号304から形成された時系列データをカットし、ポーズの前後の2、3フレームにし、既に説明したように認識システムを動作させることを含む。

3.3.3 デバイスに依存しない認識(Device-independent recognition)

【0167】

好ましい実施形態は、モーション信号304を提供するデバイス302の詳細を抽象化するアプリケーションに対して、固定のアプリケーションプログラミングインターフェース(API)(標準デバイスに依存しないモーションデータAPI)を確立し、新たなデバイスの製造業者又は販売業者又はユーザーがデバイスの十分な統計をシステムに知らせるための登録インターフェースを提供する。これは、アプリケーション開発者にとっては非常に重要な要素である。デバイスフラグメンテーションが少ないほど、所定のモーション検出アプリケーション用の抽象プラットフォームが広がる。エンドユーザーは、モーション検出アプリケーションと情報をやりとりするときに、より広い範囲の入力デバイスを使用できるという点で、間接的にのみAPIの恩恵を受ける。しかしながら、より多くのプラットフォーム上でより多くのモーション検出アプリケーションが利用可能になることの重要な恩恵及び

40

50

利点は明らかである。

【 0 1 6 8 】

様々な仕様、エラー特性及び機能を有する多くの異なる慣性検出デバイス及びビデオキャプチャデバイスがある。位置0にある慣性センサーを有するデバイスを、異なる相対位置1にあるセンサーを有する異なるデバイス用の数学及びコードに基づいて動作させることは、深刻な障害を引き起す場合がある。

【 0 1 6 9 】

認識のために、一実施形態では、モーション信号304は、デバイスに特有の特性の多くを除去するように処理されている。これによって、合理的な限界の範囲内で、一つのタイプのデバイスがモーションレコグナイザーを生成するために用いられ、第2のタイプのデバイスがプレイ中に使用される。例えば、様々な加速度計について、最大の感度及び範囲が既知であり、剛性コントローラーボディ内のセンサーのそれぞれの位置が既知であれば、2つの異なるデバイスの出力は、認識に影響を与えるほどの情報を失わずに互いにマップすることができる。

【 0 1 7 0 】

デバイス独立性は、汎用のモーションコントロール環境での追跡にも適用されなければならない。一つの例タスクは、デバイスの可視部の位置及び方向を追跡することである。部分的には、追跡結果を認識のための入力として使用するためである。既知のセンサー位置を有する既知のデバイス上の既知の位置を追跡するとき、標準アプローチは、センサーの経時的な位置を追跡し、最後には、その結果をユーザーに報告するときに、実際のセンサー位置を報告する代わりに、コントローラーの剛性ボディ上の既知の可視ポイントを報告することである。例えば、センサーがコントローラーの質量中心にある場合、最初に、質量中心の位置及び方向を追跡し、次に、 $Pos - orientation * vecAcc$ として、可視ポイントの位置を計算する。ここで、Posは、ワールドフレーム(world frame)中の慣性センサーの追跡位置であり、orientationは、コントローラーの方向であり、vecAccは、我々が位置を特定しようとする可視ポイントに対する慣性センサーの位置である。

【 0 1 7 1 】

より多くの恩恵があるが困難な問題は、レコグナイザーを生成しているデバイスの特性が認識されるデバイスと異なる場合に、モーションレコグナイザーを変化させずに使用することである。(言い換えると、位置1にある慣性センサーからのデータが位置2に置かれたそのデバイスから生成されたようにそのデータを変換することである。)データ変化を行う単純なアプローチは、實際上、うまくいかない。なぜなら慣性センサーノイズが強すぎるからである。センサーノイズを考慮した次の方法は、標準モーションデータAPIを用いてデバイスに依存しない認識を行うことと可能にする。次の擬似-コードカットアウトは、質量中心に配置されていないセンサーからの慣性測定値を補正する際に含まれるステップを示す。質量中心に対しては、物体が剛体であれば角速度データについての補正が必要でなく、角速度データは、次のように、質量中心で測定された測定値を見積もるために使用される。

```
LX = accX;
LY = accY;
LZ = accZ;
```

//質量中心を中心とした加速度計の回転の接戦効果を引く

```
LZ -= aaX*yOffset;
LY += aaX*zOffset;
LX -= aaY*zOffset;
LZ += aaY*xOffset;
LY -= aaZ*xOffset;
LX += aaZ*yOffset;
```

10

20

30

40

50

```
// 求心加速度、質量中心での加速度に戻る
LX += xOffset*(          avY*avY + avZ*avZ);
LY += yOffset*(avX*avX +          avZ*avZ);
LZ += zOffset*(avX*avX + avY*avY          );
```

```
// ジャイロ作用を補償する
LX -= avX*(          avY*yOffset + avZ*zOffset);
LY -= avY*(avX*xOffset +          avZ*zOffset);
LZ -= avZ*(avX*xOffset + avY*yOffset          );
```

キー : accX, accY, accZ - センサー位置でそれぞれの軸に沿って測定された直線加速度 10

avX, avY, avZ - それぞれの軸の周りで測定された角速度
 aaX, aaY, aaZ - それぞれの軸の周りで計算された角加速度
 xOffset, yOffset, zOffset - 加速度計と質量中心の間の物理的分離
 LX, LY, LZ - 質量中心についての計算された直線加速度

センサーノイズを考慮した改良

1) 実際には、センサーデータの複数の期間に渡って角加速度を測定することにより、計算された直線加速度についてノイズの影響が減少したスムーズな推定値が得られる。使用した測定値の数は、特定のジャイロスコープのサンプリングレート及びノイズ特性に従って変化した。 20

```
dt = history[endIndex].time ? history[startIndex].time;
aaX = (history[endIndex].avX ? history[startIndex].avX)/dt;
aaY = (history[endIndex].avY ? history[startIndex].avY)/dt;
aaZ = (history[endIndex].avZ ? history[startIndex].avZ)/dt;
```

2) 対応する角速度が小さいとき、角加速度を減少させた（大部分の加速度は、この場合、ノイズの結果であると分かる。）

// 角速度が小さい場合、角加速度は、主に、複数の値の間をジャンプするジャイロの測定に依っており、約5 rad/sec²までのジャンプが生じる

```
if ( reduceAA )
{
```

real const aaReduction = 5.0f; // ゼロ角速度(rad/sec/sec)では、aaをこれくらい減少させる。

real const smallAngularVelocity = 0.5f; // 角速度がこの値(rad/sec)よりも大きければ加速度を調節しない。

```
moveTowardsZero( aaX, aaReduction*(smallAngularVelocity - fabsf(
avX ))/smallAngularVelocity );
```

```
moveTowardsZero( aaY, aaReduction*(smallAngularVelocity - fabsf(
avY ))/smallAngularVelocity );
```

```
moveTowardsZero( aaZ, aaReduction*(smallAngularVelocity - fabsf(
avZ ))/smallAngularVelocity );
```

```
}
```

【 0 1 7 2 】

例えば、加速度計が強い力を表現できず、モーションが強い力を要求する場合、マッピングは失敗する。また、測定するデータが本質的に非常に異なるデバイス同士の間でもマッピングは失敗する。

【 0 1 7 3 】

例えば、ジョイスティックプッシュを加速度計にマッピングしようとしても無駄である。しかし、合理的な制限の範囲内で、ある成分を別のものに単純にマッピングすることは、ハードウェア詳細を抽象化して、多くの場合は、クロス-デバイス認識サービスを可能にする。全てのモーション信号は、それを生成したモーションデバイスにタグ付けされて 50

いる。このことは、分類されるモーション信号304を現在生成しているモーションデバイスに、所定のモーションレコグナイザー306を認識RTLがマッピングすることを可能にする（このようなマッピングが有用である場合はいつでも。）。

3.3.4 例

【0174】

例えば、入力は、タブレット又はタッチ検出スクリーン上の2Dトレースから生成されるモーション信号304を含み、任意的にボタンプレスと組み合わせられる。

【0175】

例えば、上記の広い範囲の入力及び出力は、ユーザーが上半身を使用して、コンピュータービデオゲーム内の対応するアバターを操縦したり（スロープを下るボブスレーを考えなさい。）、ドッジさせたり、ダックさせたり、ブロックさせたり、ジャンプさせたり、引いたり、押したりさせることを可能にする。例えば、モーション認識は、ヒューマンプレイヤーからゲーム内のアバター（例えばゴリラ、蟻、蜂、などの形態。）にターゲットを変えることができる。主な障害は、もはや制御テクノロジーではなく、創造性の限界である。

10

【0176】

例えば、入力は、二人以上から受け取って関連させ、近いタイミングで相補的に対となるモーション（例えばダンス）を実行してもよい。

【0177】

例えば、出力様式は、エンドユーザーモーションについて明示的に予測を行うために、モーションレコグナイザーを使用することを含む。明らかに早期の最良推測及び早期のアニメーションフィードバックは、ユーザーモーションを予測することの一つの非常に具体的な用途である。この機能は、実際には、多くの効果（例：ゲーム内でユーザーの心を読む振りをすること。）に対して使用可能な汎用のモーション予測機能である。

20

【0178】

本発明は、ある程度の具体性で十分に詳細に説明してきた。種々の実施形態の現在の開示は、例示に過ぎず、部材の構成や組み合わせにおいて数多くの変形が本発明の精神と範囲を外れることなく可能であることは当業者に自明である。ここで議論した実施形態は、形式や構成の点で情報ユニットの表示に関していくらかの限定を含んでいるように見えるが、本発明の適用範囲はこのような実施形態をはるかに超えており、そのことは当業者は理解可能である。従って、本発明の範囲は、上記の実施形態よりもむしろ添付のクレームによって定義される。

30

【図 1 A】

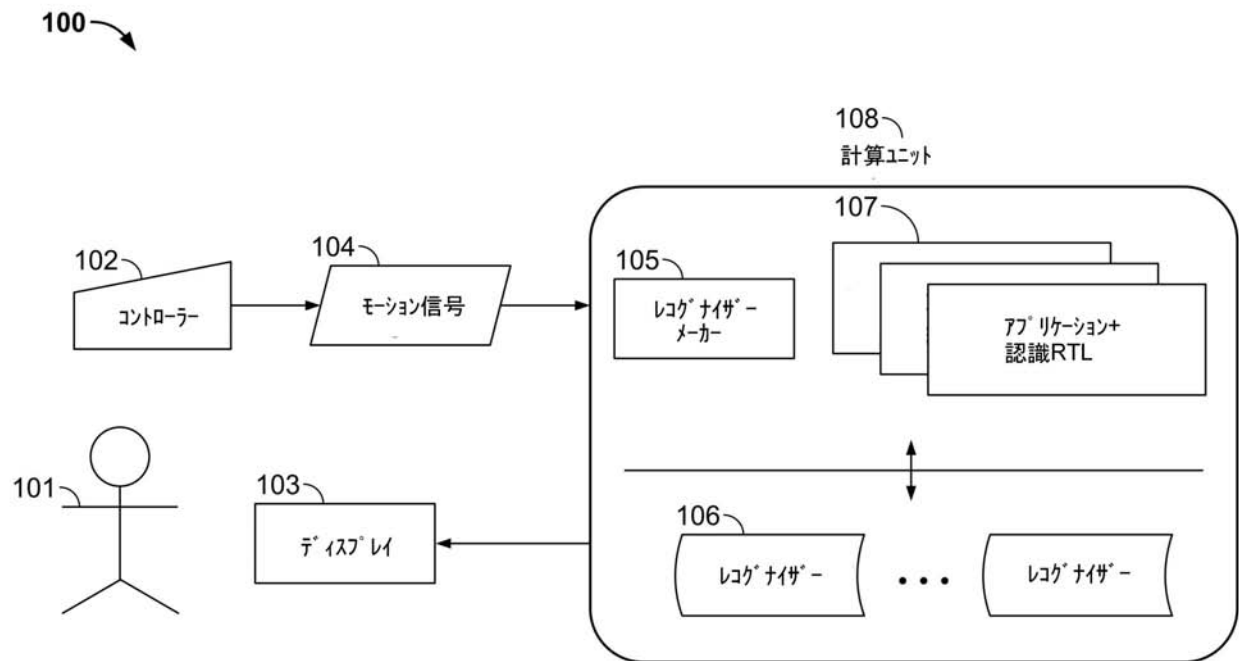


FIG. 1A

【図 1 B】

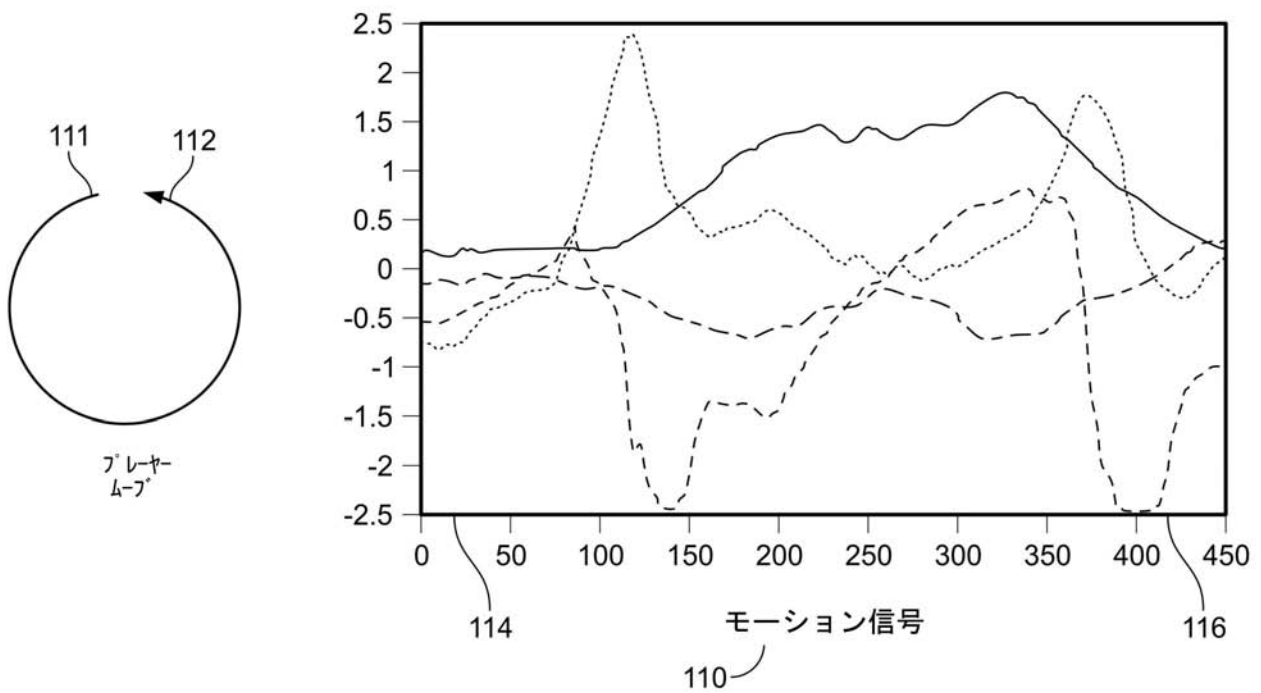


FIG. 1B

【図 2】

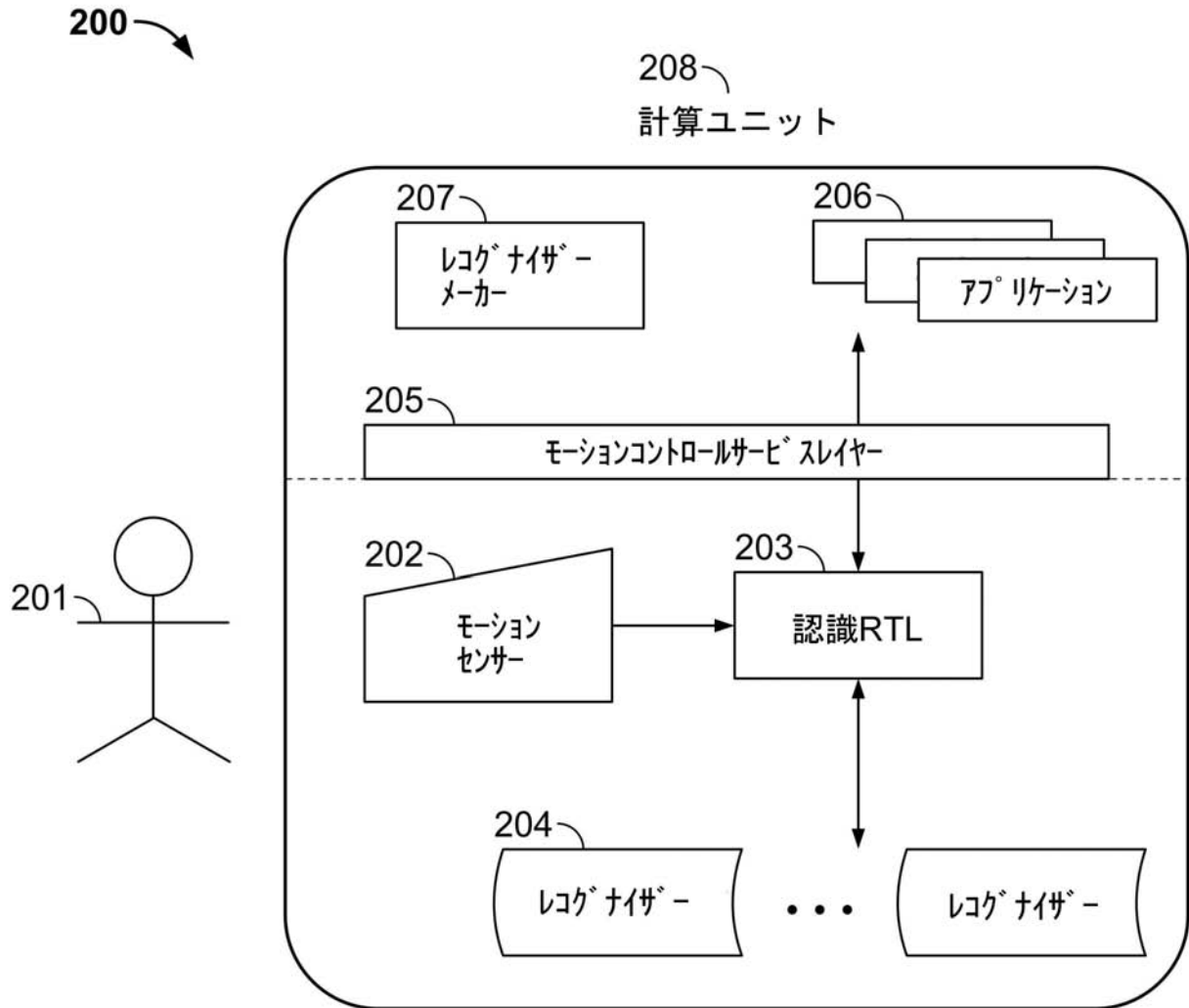


FIG. 2

【図 3】

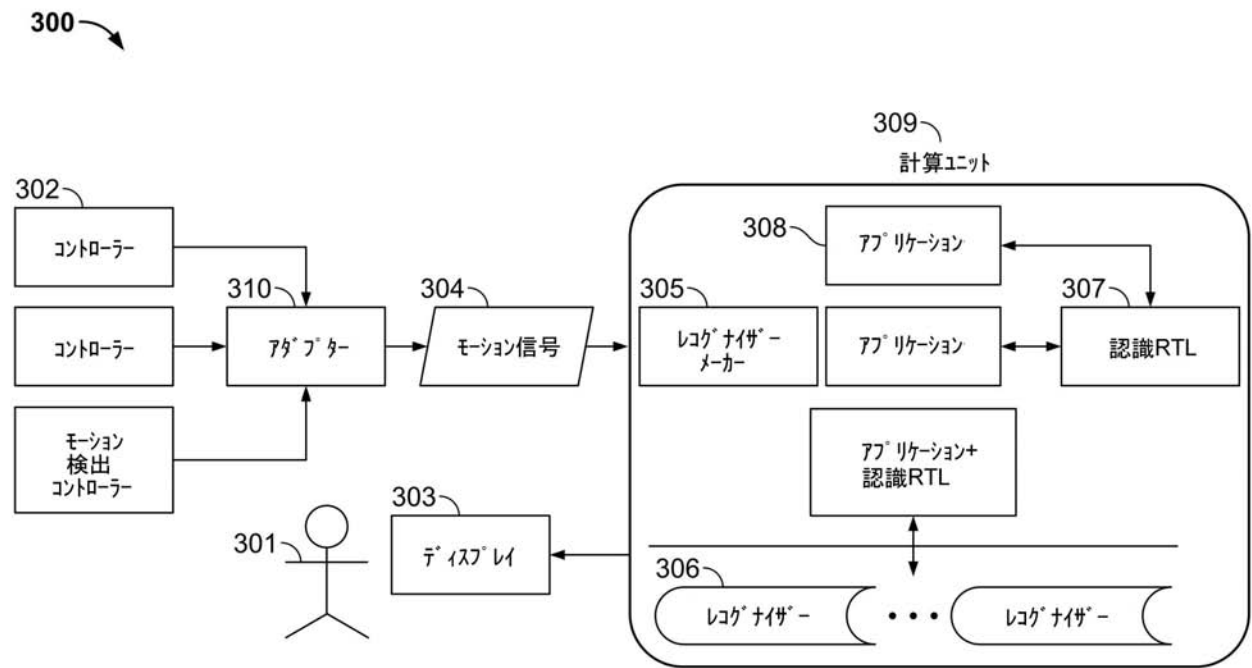
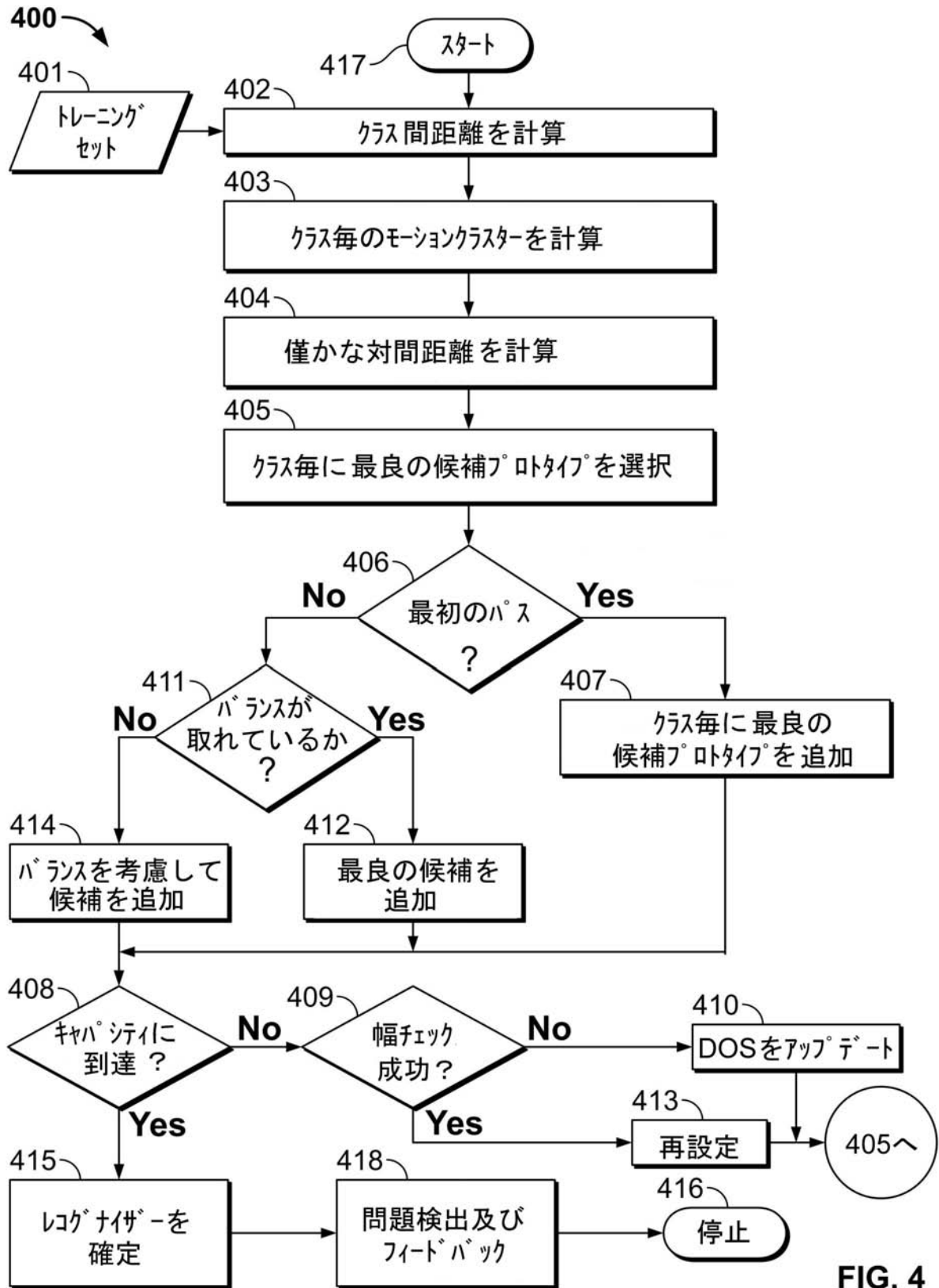
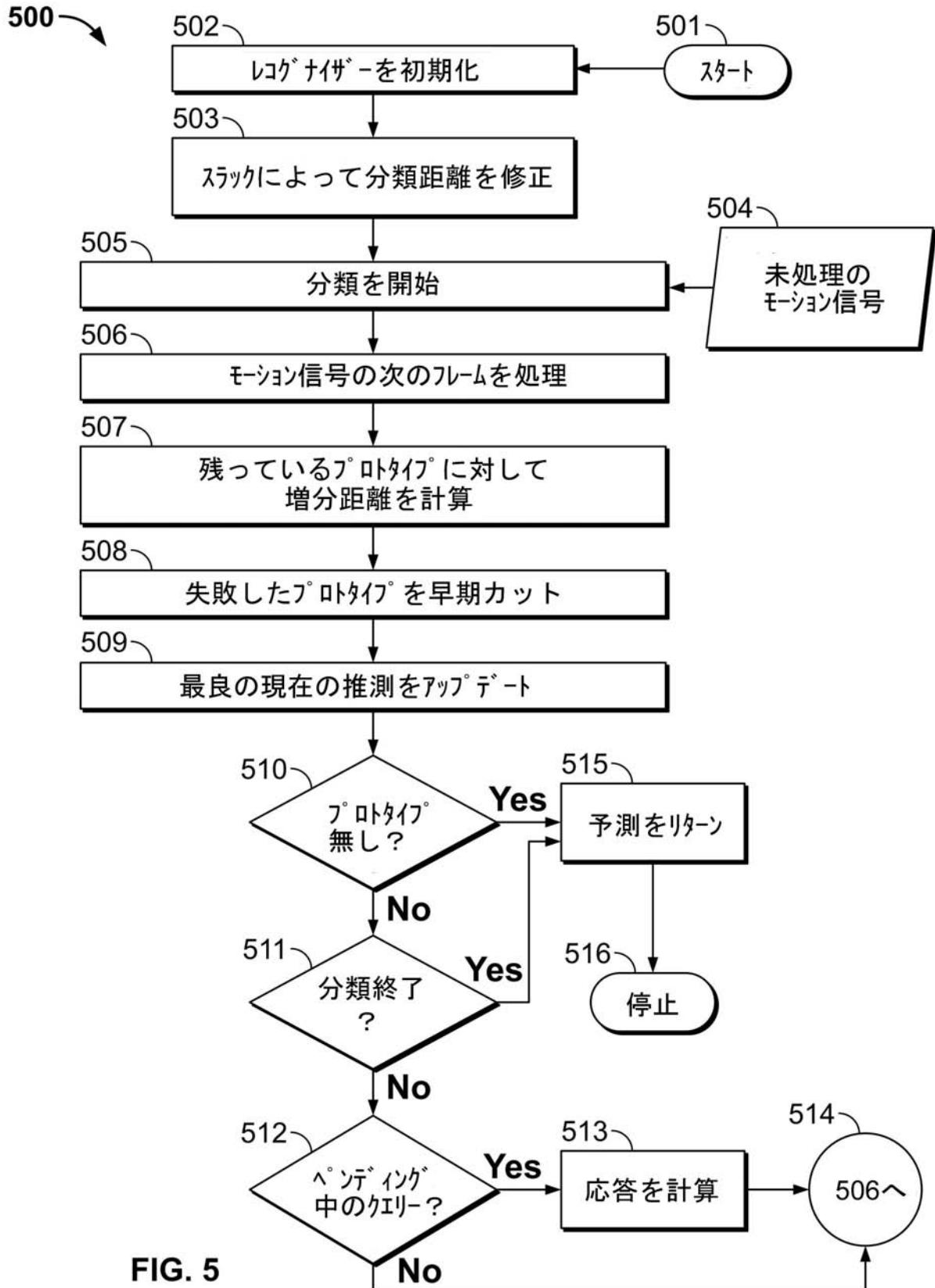


FIG. 3

【図4】



【図5】



【図 6】

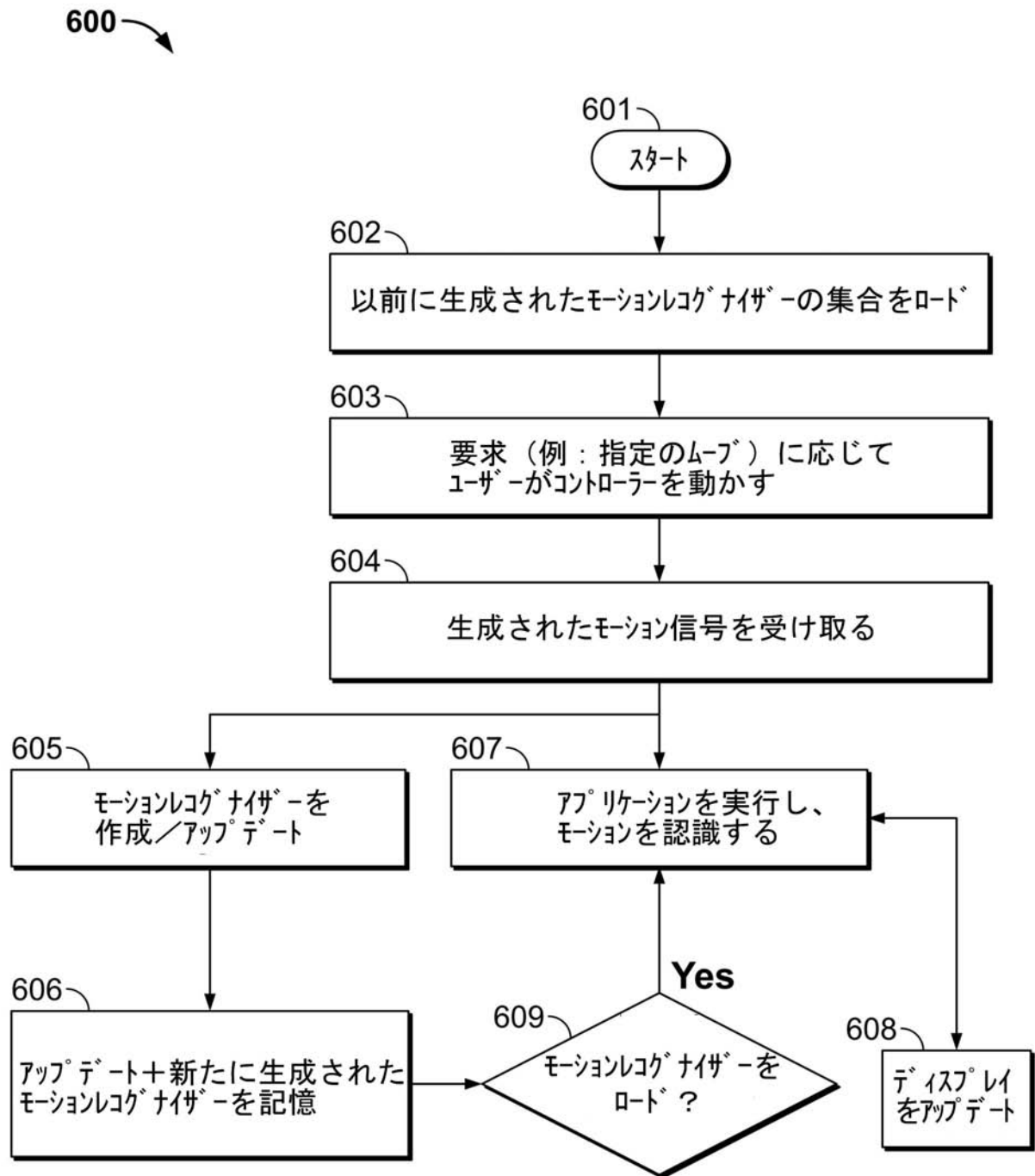


FIG. 6

【手続補正書】

【提出日】平成23年3月1日(2011.3.1)

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】0021

【補正方法】変更

【補正の内容】

【0021】

【図 1 A】図1Aは、本発明の一実施形態による構成を示し、この構成では、ディスプレイ、コントローラー、及び計算ユニットが3つの別々のデバイスである。

【手続補正 2】

【補正対象書類名】明細書

【補正対象項目名】0 0 2 2

【補正方法】変更

【補正の内容】

【0 0 2 2】

【図 1 B】図1Bは、モーションを行っているプレーヤーから生じる例示的モーション信号を示す。

【手続補正 3】

【補正対象書類名】明細書

【補正対象項目名】0 0 2 3

【補正方法】変更

【補正の内容】

【0 0 2 3】

【図 2】図2は、本発明の別の実施形態による、コントローラーとして働く単一のデバイスを示す。このデバイスは、ディスプレイ及び計算ユニットを含む。

【手続補正 4】

【補正対象書類名】明細書

【補正対象項目名】0 0 2 4

【補正方法】変更

【補正の内容】

【0 0 2 4】

【図 3】図3は、例示的構成を示す。この構成では、モーション信号は、エンドユーザーの種々の動き及びアクションを測定する多くの異なる入力で構成され、そのデータに応答するモーションレコグナイザーを構築可能なレコグナイザーメーカーに与えられる。

【手続補正 5】

【補正対象書類名】明細書

【補正対象項目名】0 0 2 5

【補正方法】変更

【補正の内容】

【0 0 2 5】

【図 4】図4は、本発明の一実施形態による、パーソナライズ化アドホックモーションレコグナイザーを作成するプロセスを示す。

【手続補正 6】

【補正対象書類名】明細書

【補正対象項目名】0 0 2 6

【補正方法】変更

【補正の内容】

【0 0 2 6】

【図 5】図5は、本発明の一実施形態による、モーション認識を行うプロセスを示す。

【手続補正 7】

【補正対象書類名】明細書

【補正対象項目名】0 0 2 7

【補正方法】変更

【補正の内容】

【0 0 2 7】

【図 6】図6は、アドホックパーソナライズ化モーションレコグナイザーを作成するとともに、その同じレコグナイザーを使用するモーション検出アプリケーションと情報をやり

とりして、モーションコントロールを提供するプロセスを示す。 - 1 -

フロントページの続き

- (72)発明者 ヨーイチロウ カワノ
アメリカ合衆国 カリフォルニア州、サンノゼ市 サラトガ アベニュー 4 5 6 アパートメント
1 1 1
- (72)発明者 チャールズ ムジック, ジュニア
アメリカ合衆国 カリフォルニア州 9 4 0 0 2、ベルモント市 セメリア アベニュー 2 2 2
5
- (72)発明者 ウィリアム ロバート パワーズ, I I I
アメリカ合衆国 カリフォルニア州 9 4 1 1 0、サンフランシスコ市 2 4 番ストリート 3 6 8 7
- (72)発明者 スチュアート レイノルズ
アメリカ合衆国 カリフォルニア州 9 4 0 4 1、マウンテンビュー市 フェアモントアベニュー 6
1 9
- (72)発明者 ダナ ウィルキンソン
アメリカ合衆国 カリフォルニア州、マウンテンビュー市 ラッサム ストリート 2 2 5 0 ユ
ニット 6 8
- (72)発明者 イアン ライト
アメリカ合衆国 カリフォルニア州 9 4 0 8 7 サニーベール市 ブルックライン ドライブ
8 5 5 アpartment H
- (72)発明者 ウェイ イェン
アメリカ合衆国 ワシントン州 9 8 1 0 5、シアトル市 エヌイー ベルヴォアール プレイス
3 9 3 0

F ターム(参考) 5B050 BA09 CA06 EA19 EA24 EA26 FA02 FA05 FA08 FA09 FA10