

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号  
特許第7589712号  
(P7589712)

(45)発行日 令和6年11月26日(2024.11.26)

(24)登録日 令和6年11月18日(2024.11.18)

(51)国際特許分類

F I

G 0 6 F 9/48 (2006.01)

G 0 6 F 9/48 3 0 0 Z

請求項の数 5 (全17頁)

(21)出願番号	特願2022-81059(P2022-81059)	(73)特許権者	000003207
(22)出願日	令和4年5月17日(2022.5.17)		トヨタ自動車株式会社
(65)公開番号	特開2023-169751(P2023-169751 A)		愛知県豊田市トヨタ町1番地
(43)公開日	令和5年11月30日(2023.11.30)	(74)代理人	100099759
審査請求日	令和6年1月18日(2024.1.18)		弁理士 青木 篤
		(74)代理人	100123582
			弁理士 三橋 真二
		(74)代理人	100092624
			弁理士 鶴田 準一
		(74)代理人	100147555
			弁理士 伊藤 公一
		(74)代理人	100123593
			弁理士 関根 宣夫
		(74)代理人	100133835
			弁理士 河野 努

最終頁に続く

(54)【発明の名称】 リソース管理装置、リソース管理方法及びリソース管理用コンピュータプログラム

(57)【特許請求の範囲】

【請求項1】

複数の処理のうちの共有リソースを利用して実行中の処理の優先度及び当該実行中の処理が利用する、前記共有リソースの範囲を表す処理登録情報を記憶する記憶部と、

前記複数の処理のうちの何れかの処理の実行が開始される度に、当該処理についての前記優先度及び前記範囲を前記処理登録情報に追加し、一方、前記複数の処理のうちの何れかの処理について前記共有リソースの利用が終了する度に、当該処理についての前記優先度及び前記範囲を前記処理登録情報から削除するよう、前記処理登録情報を更新する更新部と、

前記複数の処理のうちの前記共有リソースを利用して実行中の第1の処理が前記共有リソースに記憶されているデータを書き換える場合、前記処理登録情報を参照して、前記複数の処理のうち、当該第1の処理が利用する前記共有リソースの範囲を利用して実行中の第2の処理が存在するか否かを判定する判定部と、

前記第1の処理の優先度よりも高い優先度を有する前記第2の処理が存在する場合、前記第2の処理が終了するまで前記第1の処理を中断させ、一方、前記第1の処理の優先度よりも低い優先度を有する前記第2の処理が存在する場合、前記第1の処理が終了するまで前記第2の処理を中断させる処理制御部と、  
を有するリソース管理装置。

【請求項2】

前記処理制御部は、前記第1の処理を中断させた場合、前記第1の処理の優先度よりも

10

20

高い優先度の前記第 2 の処理が終了すると、前記第 1 の処理を最初から再実行させる、請求項 1 に記載のリソース管理装置。

【請求項 3】

前記複数の処理の一つはセンサにより生成されたセンサ信号から、当該センサ信号に表された所定の物体を検出する物体検出処理であり、前記複数の処理の他の一つは前記センサに生じた異常または不調を検出する異常検出処理であり、前記異常検出処理の優先度は前記物体検出処理の優先度よりも高く設定される、請求項 1 または 2 に記載のリソース管理装置。

【請求項 4】

複数の処理のうちの何れかの処理の実行が開始される度に、当該処理についての優先度及び当該処理が利用する共有リソースの範囲を、記憶部に記憶された、前記複数の処理のうちの前記共有リソースを利用して実行中の処理の優先度及び当該実行中の処理が利用する、前記共有リソースの範囲を表す処理登録情報に追加し、一方、前記複数の処理のうちの何れかの処理について前記共有リソースの利用が終了する度に、当該処理についての前記優先度及び前記範囲を前記処理登録情報から削除するよう、前記処理登録情報を更新し、

10

前記複数の処理のうちの前記共有リソースを利用して実行中の第 1 の処理が前記共有リソースに記憶されているデータを書き換える場合、前記処理登録情報を参照して、前記複数の処理のうち、当該第 1 の処理が利用する前記共有リソースの範囲を利用して実行中の第 2 の処理が存在するか否かを判定し、

前記第 1 の処理の優先度よりも高い優先度を有する前記第 2 の処理が存在する場合、前記第 2 の処理が終了するまで前記第 1 の処理を中断させ、

20

前記第 1 の処理の優先度よりも低い優先度を有する前記第 2 の処理が存在する場合、前記第 1 の処理が終了するまで前記第 2 の処理を中断させる、  
ことを含むリソース管理方法。

【請求項 5】

複数の処理のうちの何れかの処理の実行が開始される度に、当該処理についての優先度及び当該処理が利用する共有リソースの範囲を、記憶部に記憶された、前記複数の処理のうちの前記共有リソースを利用して実行中の処理の優先度及び当該実行中の処理が利用する、前記共有リソースの範囲を表す処理登録情報に追加し、一方、前記複数の処理のうちの何れかの処理について前記共有リソースの利用が終了する度に、当該処理についての前記優先度及び前記範囲を前記処理登録情報から削除するよう、前記処理登録情報を更新し、

30

前記複数の処理のうちの前記共有リソースを利用して実行中の第 1 の処理が前記共有リソースに記憶されているデータを書き換える場合、前記処理登録情報を参照して、前記複数の処理のうち、当該第 1 の処理が利用する前記共有リソースの範囲を利用して実行中の第 2 の処理が存在するか否かを判定し、

前記第 1 の処理の優先度よりも高い優先度を有する前記第 2 の処理が存在する場合、前記第 2 の処理が終了するまで前記第 1 の処理を中断させ、

前記第 1 の処理の優先度よりも低い優先度を有する前記第 2 の処理が存在する場合、前記第 1 の処理が終了するまで前記第 2 の処理を中断させる、

ことをコンピュータに実行させるためのリソース管理用コンピュータプログラム。

40

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、複数の処理が共通に利用可能な共有リソースを管理するリソース管理方法及びリソース管理用コンピュータプログラムに関する。

【背景技術】

【0002】

複数の処理を並列に実行することが可能なシステムにおいて、各処理の実行順序を動的に決定する技術が提案されている（特許文献 1 を参照）。

【0003】

50

特許文献 1 に開示された方法は、個々のクロックサイクルで複数の命令を発行できるマルチスレッドプロセッサにおいて、複数の利用可能な命令から個々のクロックサイクルでどの命令を発行すべきかを動的に決定する。この方法は、複数の利用可能な命令から最も優先度が高い命令を決定し、最も優先度が高い命令と残りの利用可能な命令のそれぞれとの互換性を判断する。その際、この方法は、個々の命令が同じリソースを必要としない場合に互いに互換性があると判断する。そしてこの方法は、最も優先度が高い命令を、その命令と互換性があるとともに互いに互換性がある他の命令とともに同じクロックサイクルで発行する。そして最も優先度が高い命令は投機的命令とされない。

【先行技術文献】

【特許文献】

【 0 0 0 4 】

【文献】特表 2 0 1 2 - 5 2 5 6 2 0 号公報

【発明の概要】

【発明が解決しようとする課題】

【 0 0 0 5 】

同時に実行される複数の処理のうち、優先度が相対的に低い処理が共有リソースを先に占有してしまうことがある。このような場合、同時に実行される複数の処理のうち、優先度が相対的に高い処理の実行を待機する必要がある。

【 0 0 0 6 】

そこで、本発明は、共有リソースを使用する複数の処理のうちの優先度の高い方の処理が優先的に実行されることを保障できるリソース管理装置を提供することを目的とする。

【課題を解決するための手段】

【 0 0 0 7 】

一つの実施形態によれば、リソース管理装置が提供される。このリソース管理装置は、複数の処理のうちの共有リソースを利用して実行中の処理の優先度及び実行中の処理が利用する、共有リソースの範囲を表す処理登録情報を記憶する記憶部と、複数の処理のうちの何れかの処理の実行が開始される度に、その処理についての優先度及び共有リソースの範囲を処理登録情報に追加し、一方、複数の処理のうちの何れかの処理について共有リソースの利用が終了する度に、その処理についての優先度及び共有リソースの範囲を処理登録情報から削除するよう、処理登録情報を更新する更新部と、複数の処理のうちの共有リソースを利用して実行中の第 1 の処理が共有リソースに記憶されているデータを書き換える場合、処理登録情報を参照して、複数の処理のうちの、第 1 の処理が利用する共有リソースの範囲を利用して実行中の第 2 の処理が存在するか否かを判定する判定部と、第 1 の処理の優先度よりも高い優先度を有する第 2 の処理が存在する場合、第 2 の処理が終了するまで第 1 の処理を中断させ、一方、第 1 の処理の優先度よりも低い優先度を有する第 2 の処理が存在する場合、第 1 の処理が終了するまで第 2 の処理を中断させる処理制御部と、を有する。

【 0 0 0 8 】

このリソース管理装置において、処理制御部は、第 1 の処理を中断させた場合、第 1 の処理の優先度よりも高い優先度の第 2 の処理が終了すると、第 1 の処理を最初から再実行させることが好ましい。

【 0 0 0 9 】

また、このリソース管理装置において、複数の処理の一つはセンサにより生成されたセンサ信号から、そのセンサ信号に表された所定の物体を検出する物体検出処理であり、複数の処理の他の一つはセンサに生じた異常または不調を検出する異常検出処理であり、異常検出処理の優先度は物体検出処理の優先度よりも高く設定されることが好ましい。

【 0 0 1 0 】

本発明の他の形態によれば、リソース管理方法が提供される。このリソース管理方法は、複数の処理のうちの何れかの処理の実行が開始される度に、その処理についての優先度及び共有リソースの範囲を、記憶部に記憶された、共有リソースを利用して実行中の処理

10

20

30

40

50

の優先度及び実行中の処理が利用する、共有リソースの範囲を表す処理登録情報に追加し、一方、複数の処理のうちの何れかの処理について共有リソースの利用が終了する度に、その処理についての優先度及び共有リソースの範囲を処理登録情報から削除するよう、処理登録情報を更新し、複数の処理のうちの共有リソースを利用して実行中の第１の処理が共有リソースに記憶されているデータを書き換える場合、処理登録情報を参照して、複数の処理のうち、第１の処理が利用する共有リソースの範囲を利用して実行中の第２の処理が存在するか否かを判定し、第１の処理の優先度よりも高い優先度を有する第２の処理が存在する場合、第２の処理が終了するまで第１の処理を中断させ、一方、第１の処理の優先度よりも低い優先度を有する第２の処理が存在する場合、第１の処理が終了するまで第２の処理を中断させる、ことを含む。

10

**【 0 0 1 1 】**

本発明のさらに他の形態によれば、リソース管理用コンピュータプログラムが提供される。このリソース管理用コンピュータプログラムは、複数の処理のうちの何れかの処理の実行が開始される度に、その処理についての優先度及び共有リソースの範囲を、記憶部に記憶された、共有リソースを利用して実行中の処理の優先度及び実行中の処理が利用する、共有リソースの範囲を表す処理登録情報に追加し、一方、複数の処理のうちの何れかの処理について共有リソースの利用が終了する度に、その処理についての優先度及び共有リソースの範囲を処理登録情報から削除するよう、処理登録情報を更新し、複数の処理のうちの共有リソースを利用して実行中の第１の処理が共有リソースに記憶されているデータを書き換える場合、処理登録情報を参照して、複数の処理のうち、第１の処理が利用する共有リソースの範囲を利用して実行中の第２の処理が存在するか否かを判定し、第１の処理の優先度よりも高い優先度を有する第２の処理が存在する場合、第２の処理が終了するまで第１の処理を中断させ、一方、第１の処理の優先度よりも低い優先度を有する第２の処理が存在する場合、第１の処理が終了するまで第２の処理を中断させる、ことをコンピュータに実行させるための命令を含む。

20

**【発明の効果】****【 0 0 1 2 】**

本発明に係るリソース管理装置は、共有リソースを使用する複数の処理のうちの優先度の高い方の処理が優先的に実行されることを保障できるという効果を奏する。

**【図面の簡単な説明】**

30

**【 0 0 1 3 】**

【図１】リソース管理装置が実装される車両制御システムの概略構成図である。

【図２】リソース管理装置の一実施形態である電子制御装置のハードウェア構成図である。

【図３】リソース管理処理を含む車両制御処理に関する、電子制御装置のプロセッサの機能ブロック図である。

【図４】リソース管理処理の概要を説明する図である。

【図５】リソース管理処理の動作フローチャートである。

**【発明を実施するための形態】****【 0 0 1 4 】**

以下、図を参照しつつ、リソース管理装置、及び、リソース管理装置において実行されるリソース管理方法及びリソース管理用コンピュータプログラムについて説明する。このリソース管理装置は、複数の処理のうち、共有リソースを用いて実行中の処理の優先度及びその処理が利用する共有リソースの範囲を、処理登録情報を用いて管理する。すなわち、このリソース管理装置は、何れかの処理の実行が開始される度に、その処理についての優先度及びその処理が利用する共有リソースの範囲を処理登録情報に追加する。また、このリソース管理装置は、何れかの処理について共有リソースの利用が終了する度に、その処理についての優先度及び共有リソースの範囲を処理登録情報から削除する。さらに、このリソース管理装置は、共有リソースを利用して実行中の第１の処理が共有リソースに記憶されているデータを書き換える場合、処理登録情報を参照する。そしてこのリソース管理装置は、第１の処理が利用する共有リソースの範囲を利用して実行中の第２の処理が存

40

50

在するか否かを判定する。このリソース管理装置は、第２の処理が存在し、かつ、第２の処理の優先度が第１の処理の優先度よりも高い場合、第２の処理が終了するまで第１の処理を中断させる。一方、このリソース管理装置は、第２の処理が存在し、かつ、第２の処理の優先度が第１の処理の優先度よりも低い場合、第１の処理が終了するまで第２の処理を中断させる。

#### 【００１５】

以下では、リソース管理装置を、車両制御システムに適用した例について説明する。この例では、リソース管理装置は、車両に搭載されたカメラにより得られた画像から、車両の周囲に存在する物体を検出し、検出した物体と車両とが衝突しないように、車両を自動運転制御する。その際、このリソース管理装置は、カメラから受信した画像を共有リソースに一時的に記憶する。そしてこのリソース管理装置は、共有リソースに記憶された画像に基づいて物体を検出する処理（以下、物体検出処理と呼ぶ）及びカメラの故障または不調を検出する処理（以下、異常検出処理と呼ぶ）を実行する。

#### 【００１６】

図１は、リソース管理装置が実装される車両制御システムの概略構成図である。また図２は、リソース管理装置の一つの実施形態である電子制御装置のハードウェア構成図である。本実施形態では、車両１０に搭載され、かつ、車両１０を制御する車両制御システム１は、車両１０の周囲を撮影するためのカメラ２と、リソース管理装置の一例である電子制御装置（ＥＣＵ）３とを有する。カメラ２とＥＣＵ３とは、コントローラエリアネットワークといった規格に準拠した車内ネットワークを介して通信可能に接続される。なお、車両制御システム１は、車両１０の自動運転制御に用いられる地図を記憶するストレージ装置（図示せず）をさらに有していてもよい。さらに、車両制御システム１は、LiDARあるいはレーダといった、車両１０の周囲の物体までの距離を測定するための距離センサ（図示せず）を有していてもよい。さらにまた、車両制御システム１は、GPS受信機といった、衛星測位システムに準拠して車両１０の自己位置を測位するための受信機（図示せず）を有していてもよい。さらにまた、車両制御システム１は、他の機器と無線通信するための無線端末（図示せず）、及び、車両１０の走行予定ルートを探索するためのナビゲーション装置（図示せず）などを有していてもよい。

#### 【００１７】

カメラ２は、センサの一例であり、CCDあるいはC-MOSなど、可視光に感度を有する光電変換素子のアレイで構成された２次元検出器と、その２次元検出器上に撮影対象となる領域の像を結像する結像光学系を有する。そしてカメラ２は、車両１０の前方の所定の領域を撮影するように、例えば、車両１０の車室内において車両１０の前方へ向けて取り付けられる。なお、車両１０には、撮影方向または焦点距離が互いに異なる複数のカメラが設けられてもよい。カメラ２は、所定の撮影周期（例えば１/30秒～１/10秒）ごとに車両１０の周囲の所定の領域を撮影し、その所定の領域が写った画像を生成する。カメラ２により得られた画像は、カラー画像であってもよく、あるいは、グレイ画像であってもよい。カメラ２により得られた画像は、優先度が指定される複数の処理のそれぞれの対象となるデータの一部であり、かつ、センサ信号の一部である。

#### 【００１８】

カメラ２は、画像を生成する度に、その生成した画像を、車内ネットワークを介してＥＣＵ３へ出力する。

#### 【００１９】

ＥＣＵ３は、車両１０を自動運転制御する。本実施形態では、ＥＣＵ３は、カメラ２により得られた時系列の一連の画像から検出された物体に基づいて車両１０を自動運転するよう、車両１０を制御する。そのために、ＥＣＵ３は、通信インターフェース２１と、メモリ２２と、バッファメモリ２３と、プロセッサ２４とを有する。なお、通信インターフェース２１、メモリ２２、バッファメモリ２３及びプロセッサ２４は、互いに異なる回路として構成されてもよく、あるいは、一つの集積回路として一体的に構成されてもよい。

#### 【００２０】

通信インターフェース 21 は、ECU3 を車内ネットワークに接続するためのインターフェース回路を有する。すなわち、通信インターフェース 21 は、車内ネットワークを介して、カメラ 2 と接続される。そして通信インターフェース 21 は、カメラ 2 から画像を受信する度に、受信した画像をプロセッサ 24 へわたす。

【0021】

メモリ 22 は、記憶部の一例であり、例えば、揮発性の半導体メモリ及び不揮発性の半導体メモリを有する。そしてメモリ 22 は、ECU3 のプロセッサ 24 により実行される車両制御処理のアルゴリズム、車両制御処理において使用される各種のデータ及びパラメータを記憶する。例えば、メモリ 22 は、地図情報、車両制御処理で利用される識別器を特定するためのパラメータセット、画像に対して実行される複数の処理のそれぞれの優先度などを記憶する。さらに、メモリ 22 は、処理登録情報、及び、検出された物体に関する情報といった、車両制御処理の途中で生成される各種のデータを一定期間記憶する。

10

【0022】

バッファメモリ 23 は、記憶部の他の一例であり、例えば、揮発性の半導体メモリを有する。そしてバッファメモリ 23 は、通信インターフェース 21 を介して受信した、カメラからの画像を一時的に記憶する。

【0023】

プロセッサ 24 は、制御部の一例である。本実施形態では、プロセッサ 24 は、例えば、1 個または複数の CPU (Central Processing Unit) 及びその周辺回路を有する。プロセッサ 24 は、数値演算回路またはグラフィック処理回路 (Graphics Processing Unit, GPU) をさらに有していてもよい。さらに、プロセッサ 24 は、共有メモリ 241 を有する。共有メモリ 241 は、共有リソースの一例であり、例えば、プロセッサ 24 が有する CPU 及び GPU といった各演算回路または各処理回路からアクセス可能なメモリ回路として構成される。

20

【0024】

プロセッサ 24 は、車両 10 が走行している間、カメラ 2 から画像を受信する度に、受信した画像をバッファメモリ 23 に書き込む。さらに、プロセッサ 24 は、バッファメモリ 23 に保存されている画像に基づいて車両制御処理を実行する。

【0025】

図 3 は、リソース管理処理を含む車両制御処理に関する、ECU3 のプロセッサ 24 の機能ブロック図である。プロセッサ 24 は、物体検出部 31 と、異常検出部 32 と、車両制御部 33 と、更新部 34 と、判定部 35 と、処理制御部 36 とを有する。プロセッサ 24 が有するこれらの各部は、例えば、プロセッサ 24 上で動作するコンピュータプログラムにより実現される機能モジュールである。また、プロセッサ 24 が有するこれらの各部のうち、更新部 34、判定部 35 及び処理制御部 36 により実行される処理が、リソース管理処理に相当し、例えば、プロセッサ 24 が有する何れかの CPU に実行される。また、物体検出部 31 により実行される物体検出処理及び異常検出部 32 により実行される異常検出処理は、それぞれ、共有リソースの一例である共有メモリ 241 を用いて実行される複数の処理の一例である。さらに、本実施形態では、物体検出処理と異常検出処理とは互いに対して非同期に実行され、かつ、物体検出部 31 の処理の優先度よりも異常検出部 32 の処理の優先度の方が高いものとする。そのため、異常検出処理の実行が物体検出処理の実行により待機させられることが防止される。その結果として、プロセッサ 24 は、カメラ 2 に異常または不調が生じたときに、遅滞なく、ドライバに車両 10 の制御主体を移管することができる。なお、共有メモリ 241 を用いて実行される複数の処理には、物体検出処理及び異常検出処理以外の処理が含まれてもよい。例えば、車両制御部 33 による処理も、共有メモリ 241 を用いて実行されてもよい。さらに、各処理がデータを書き込み、あるいは書き込まれたデータを参照する共有メモリ 241 のメモリ領域は、各処理が利用する共有リソースの範囲の一例である。

30

40

【0026】

物体検出部 31 は、第 1 の周期ごとに、物体検出処理の実行許可について処理制御部 3

50

6に問い合わせる。処理制御部36により物体検出処理の実行の許可が得られると、バッファメモリ23に保存されている最新の画像を共有メモリ241に書き込む。そして物体検出部31は、物体検出処理の識別情報及び優先度と共有メモリ241において物体検出部31が使用するメモリ領域を表す情報とを含む処理開始情報を更新部34へ通知する。さらに、物体検出部31は、共有メモリ241に書き込まれた画像に対する物体検出処理を実行する。そしてその物体検出処理により、物体検出部31は、画像に表された検出対象となる、車両10の周囲の物体を検出する。

#### 【0027】

本実施形態では、物体検出部31は、識別器に画像を入力することで、画像に表された検出対象となる物体を含む物体領域を検出し、かつ、検出対象となる物体の種類を識別する。検出対象となる物体には、例えば、他の車両、歩行者、車線区画線といった道路標識、各種の道路標識あるいは信号機が含まれる。物体検出部31は、そのような識別器として、いわゆるディープニューラルネットワーク（以下、単にDNNと呼ぶ）を使用することができる。そして識別器として利用されるDNNは、Single Shot MultiBox Detector (SSD)、または、Faster R-CNNといった、コンボリユーションアルニューラルネットワーク型のアーキテクチャを持つものとして行うことができる。あるいは、識別器として利用されるDNNは、Vision Transformerといった、Self attention network型のアーキテクチャを有するものであってもよい。このような識別器は、検出対象となる物体が表された画像（教師画像）を多数用いて、誤差逆伝搬法といった所定の学習手法に従って、画像から検出対象となる物体を検出するように予め学習される。

#### 【0028】

物体検出部31の処理は、プロセッサ24が有するGPUといった演算回路により実行される。物体検出部31（実質的には、物体検出部31の処理を実行する演算回路。以下同様）は、共有メモリ241から画像を読み込み、読み込んだ画像を識別器に入力することで、検出対象となる物体を検出する。その際、物体検出部31は、識別器による演算の途中で生成される中間データを共有メモリ241へ書き込む。ただし、物体検出部31は、判定部35及び処理制御部36に対して、中間データの書き込みを行う処理の識別情報及び中間データを書き込む領域を表す情報を含む書換情報を通知する。そして物体検出部31は、処理制御部36から、中間データの書き込みに対する許可が得られると、中間データを共有メモリ241へ書き込む。

#### 【0029】

一方、中間データの書き込みに対する許可が得られない場合、物体検出部31は、処理制御部36からリトライ要求が通知されるまで待機する。そしてリトライ要求が通知されると、物体検出部31は、物体検出処理を最初からやり直す。その際、物体検出部31は、バッファメモリ23から最新の画像を再度読み出して共有メモリ241にその最新の画像を書き込む。そして物体検出部31は、共有メモリ241に書き込んだ最新の画像に対して物体検出処理を再実行する。

#### 【0030】

さらに、物体検出処理の実行中において、中間データの書き込みの有無にかかわらず、処理制御部36からリトライ要求が通知されたときも、物体検出部31は、上記と同様に、物体検出処理を最初からやり直す。

#### 【0031】

物体検出部31は、画像から検出対象となる物体を検出すると、画像上でのその検出対象となる物体が表された物体領域の位置及び範囲と、その物体領域に含まれる物体の種類とを表す検出情報を車両制御部33へ通知する。また、物体検出部31は、物体検出処理が終了したことを表す処理終了通知を更新部34へ出力する。

#### 【0032】

異常検出部32は、第1の周期と異なる第2の周期ごとに、異常検出処理の実行許可について処理制御部36に問い合わせる。処理制御部36により異常検出処理の実行の許可が得られると、バッファメモリ23に保存されている直近の一定期間に得られた一連の画

10

20

30

40

50

像を共有メモリ241に書き込む。そして異常検出部32は、異常検出処理の識別情報及び優先度と共有メモリ241における異常検出部32が使用するメモリ領域を表す情報を含む処理開始情報を更新部34へ通知する。さらに、異常検出部32は、共有メモリ241に書き込まれた一連の画像に対する異常検出処理を実行する。そしてその異常検出処理により、異常検出部32は、カメラ2に生じた異常または不調を検出する。

#### 【0033】

本実施形態では、異常検出部32は、一連の画像のそれぞれについて、画素値の平均値または中央値を、異常指標値として算出する。そして異常検出部32は、各画像について算出した異常指標値が異常判定条件を満たすか否か判定する。例えば、異常検出部32は、各画像について算出された異常指標値が、画像全体が黒つぶれになっていることを表す黒つぶれ閾値よりも小さい場合、異常判定条件が満たされると判定する。また、異常検出部32は、各画像について算出された異常指標値が、画像全体が白飛びになっていることを表す白飛び閾値よりも大きい場合、異常判定条件が満たされると判定する。

10

#### 【0034】

異常検出部32の処理は、プロセッサ24が有するCPUといった演算回路により実行される。異常検出部32（実質的には、異常検出部32の処理を実行する演算回路。以下同様）は、共有メモリ241から一連の画像を読み込み、読み込んだ一連の画像に対して上記の異常検出処理を実行することで、異常判定条件が満たされるか否か判定する。その際、異常検出部32は、画像ごとの異常指標値など、異常検出処理の途中で生成される中間データを共有メモリ241へ書き込む。ただし、異常検出部32は、判定部35及び処理制御部36に対して、中間データの書き込みを行う処理の識別情報及び中間データを書き込む領域を表す情報を含む書換情報を通知する。そして異常検出部32は、処理制御部36から、中間データの書き込みに対する許可が得られると、中間データを共有メモリ241へ書き込む。

20

#### 【0035】

一方、中間データの書き込みに対する許可が得られない場合、異常検出部32は、処理制御部36からリトライ要求が通知されるまで待機する。そしてリトライ要求が通知されると、異常検出部32は、異常検出処理を最初からやり直す。その際、異常検出部32は、バッファメモリ23から直近の期間における一連の画像を再度読み出して共有メモリ241にその最新の画像を書き込む。そして異常検出部32は、共有メモリ241に書き込んだ一連の画像に対して異常検出処理を再実行する。

30

#### 【0036】

さらに、異常検出処理の実行中において、中間データの書き込みの有無にかかわらず、処理制御部36からリトライ要求が通知されたときも、異常検出部32は、上記と同様に、異常検出処理を最初からやり直す。

#### 【0037】

異常検出部32は、異常判定条件が満たされる場合、カメラ2に異常または不調が生じたことを車両制御部33へ通知する。また、異常検出部32は、異常検出処理が終了したことを表す処理終了通知を更新部34へ出力する。

#### 【0038】

車両制御部33は、目的地までの走行予定ルートに沿って車両10が進むよう、直近の所定の区間（例えば、500m～1km）における車両10の走行予定経路（トラジェクトリ）を1以上生成する。走行予定経路は、例えば、所定の区間を車両10が走行する際の各時刻における、車両10の目標位置の集合として表される。そして車両制御部33は、その走行予定経路に沿って車両10が走行するように車両10の各部を制御する。車両制御部33の処理は、プロセッサ24が有するCPUといった演算回路により実行される。

40

#### 【0039】

車両制御部33は、カメラ2により得られた時系列の一連の画像から検出された、車両10の周囲に存在する物体（例えば、他の車両）と車両10とが衝突しないように走行予定経路を生成する。そのために、車両制御部33は、物体検出部31から通知された、一

50

連の画像のそれぞれについての検出情報を参照することにより、検出された物体を追跡する。

【 0 0 4 0 】

例えば、車両制御部 33 は、Lucas-Kanade法といった、オプティカルフローに基づく追跡処理を、カメラ 2 により得られた最新の画像における、着目する物体領域及び過去の画像における物体領域に対して適用することで、その物体領域に表された物体を追跡する。そのため、車両制御部 33 は、例えば、着目する物体領域に対してSIFTあるいはHarrisオペレータといった特徴点抽出用のフィルタを適用することで、その物体領域から複数の特徴点を抽出する。そして車両制御部 33 は、複数の特徴点のそれぞれについて、過去の画像における物体領域における対応する点を、適用される追跡手法に従って特定すること

10

【 0 0 4 1 】

車両制御部 33 は、追跡中の各物体について、カメラ 2 についての車両 10 への取り付け位置などの情報を用いて視点変換処理を実行することで、その物体の画像内座標を鳥瞰画像上の座標（鳥瞰座標）に変換する。その際、車両制御部 33 は、各画像の取得時における、車両 10 の位置及び姿勢と、検出された物体までの推定距離と、車両 10 からその物体へ向かう方向とにより、各画像の取得時における、検出された物体の位置を推定できる。なお、各画像取得時における車両 10 の位置及び姿勢は、カメラ 2 により得られた画像と高精度地図とを照合することで推定されればよい。そして車両制御部 33 は、一連の鳥瞰座標に対してKalman FilterまたはParticle filterなどを用いた予測処理を実行することで、その物体の所定時間先までの予測軌跡を推定することができる。

20

【 0 0 4 2 】

車両制御部 33 は、追跡中の各物体の予測軌跡に基づいて、何れの物体についても所定時間先までの追跡中の物体のそれぞれと車両 10 間の距離の予測値が所定距離以上となるように、車両 10 の走行予定経路を生成する。さらに、車両制御部 33 は、GPS受信機（図示せず）により測位された車両 10 の現在位置及び地図情報を参照して、車両 10 が目的地へ向かうために車線変更する必要があるか否か判定する。そして車線変更の必要がない場合、車両制御部 33 は、検出された車線区画線を参照して、車両 10 が走行中の自車線からはみ出さないように走行予定経路を生成する。また、車線変更の必要がある場合、車両制御部 33 は、車両 10 が目的地へ向かう車線へ移動するように走行予定経路を生成する。なお、車両制御部 33 は、複数の走行予定経路を生成してもよい。この場合、車両制御部 33 は、複数の走行予定経路のうち、車両 10 の加速度の絶対値の総和が最小となる経路を選択してもよい。

30

【 0 0 4 3 】

車両制御部 33 は、走行予定経路を設定すると、車両 10 がその走行予定経路に沿って走行するように車両 10 の各部を制御する。例えば、車両制御部 33 は、走行予定経路、及び、車速センサ（図示せず）により測定された車両 10 の現在の車速に従って、車両 10 の目標加速度を求め、その目標加速度となるようにアクセル開度またはブレーキ量を設定する。そして車両制御部 33 は、設定されたアクセル開度に従って燃料噴射量を求め、その燃料噴射量に応じた制御信号を車両 10 のエンジンの燃料噴射装置へ出力する。あるいは、車両制御部 33 は、設定されたアクセル開度に従って車両 10 を駆動するモータへ供給する電力量を求め、その電力量がモータへ供給されるように、モータの駆動回路を制御する。また、車両制御部 33 は、設定されたブレーキ量に応じた制御信号を車両 10 のブレーキへ出力する。

40

【 0 0 4 4 】

さらに、車両制御部 33 は、車両 10 が走行予定経路に沿って走行するために車両 10 の進路を変更する場合には、その走行予定経路に従って車両 10 の操舵角を求める。そし

50

て車両制御部 33 は、その操舵角に応じた制御信号を、車両 10 のステアリングを制御するアクチュエータ（図示せず）へ出力する。

【0045】

また、車両制御部 33 は、異常検出部 32 から、カメラ 2 に異常または不調が生じたことが通知されると、自動運転制御を終了して、車両 10 の運転制御の主体を車両 10 のドライバへ移管する。そのために、車両制御部 33 は、車両 10 の車室内に設けられたユーザインターフェース（図示せず）を介して、運転制御の主体をドライバへ移管することを通知する。ユーザインターフェースは、例えば、インストルメントパネルに設けられる 1 以上の光源あるいは表示装置、スピーカ、ステアリングあるいはドライバシートに設けられる振動子のうちの少なくとも一つを有する。そして車両制御部 33 は、1 以上の光源を所定のパターンで点灯あるいは点滅させることで運転制御の主体をドライバへ移管することを通知する。あるいは、車両制御部 33 は、表示装置に所定のメッセージあるいはアイコンを表示させることで運転制御の主体をドライバへ移管することを通知する。あるいはまた、車両制御部 33 は、スピーカに所定の音声を出力させ、または、振動子を所定のパターンで振動させることで、運転制御の主体をドライバへ移管することを通知する。車両制御部 33 は、上記の通知方法の二つ以上を用いて、運転制御の主体をドライバへ移管することを通知してもよい。

10

【0046】

車両制御部 33 は、ドライバへの運転制御主体の移管の通知を開始してから所定時間を経過すると、車両 10 の自動運転制御を終了する。自動運転制御の終了後、車両 10 は、ドライバの運転操作に従って制御される。

20

【0047】

更新部 34 は、共有メモリ 241 を利用する複数の処理のうちの何れかの処理の実行が開始される度、あるいは、その実行が終了する度に、処理登録情報を更新する。処理登録情報には、共有メモリ 241 を利用して実行中の処理ごとに、その処理の優先度及び共有メモリ 241 においてその処理が利用するメモリ領域を表す情報が登録される。すなわち、更新部 34 は、物体検出部 31 または異常検出部 32 から処理開始情報を受け取ると、その処理についての優先度及びその処理が利用する共有メモリ 241 のメモリ領域を表す情報を処理登録情報に追加する。一方、物体検出部 31 または異常検出部 32 から処理終了通知を受け取る度に、その処理終了通知で示される、実行を終了した処理についての優先度及びその処理が利用するメモリ領域を表す情報を処理登録情報から削除する。

30

【0048】

更新部 34 は、処理登録情報を更新する度に、更新した処理登録情報をメモリ 22 に保存する。さらに、更新部 34 は、処理登録情報から終了した処理に関する優先度などを削除すると、その終了した処理について処理制御部 36 へ通知する。

【0049】

判定部 35 は、共有メモリ 241 を利用して実行中あるいは実行開始しようとする第 1 の処理が共有メモリ 241 に記憶されているデータを書き換える場合、処理登録情報を参照する。そして判定部 35 は、複数の処理のうち、第 1 の処理が利用する共有メモリ 241 のメモリ領域を利用して実行中の第 2 の処理が存在するか否かを判定する。本実施形態では、物体検出部 31 から書換情報が通知されると、判定部 35 は、処理登録情報を参照して、通知された書換情報に示されるメモリ領域を利用して実行中の異常検出処理が存在するか否かを判定する。この場合、物体検出処理が第 1 の処理の一例となる。処理登録情報に、実行中の異常検出処理が利用するメモリ領域が登録されている場合、判定部 35 は、登録されているメモリ領域と、物体検出部 31 から通知された書換情報に示されるメモリ領域とが重なるか否かを判定する。そして判定部 35 は、実行中の異常検出処理が利用するメモリ領域と、書換情報に示されるメモリ領域の少なくとも一部が重なる場合、物体検出処理が利用する共有メモリ 241 のメモリ領域を利用して実行中の異常検出処理が存在すると判定する。この場合、実行中の異常検出処理は、第 2 の処理の一例となる。

40

【0050】

50

なお、処理登録情報に、共有メモリ 241 を利用して実行中の異常検出処理が登録されていない場合、判定部 35 は、物体検出処理が利用する共有メモリ 241 のメモリ領域を利用して実行中の異常検出処理は存在しないと判定する。また、実行中の異常検出処理が利用するメモリ領域と物体検出部 31 から通知された書換情報に示されるメモリ領域とが重ならない場合も、判定部 35 は、物体検出処理が利用するメモリ領域を利用して実行中の異常検出処理は存在しないと判定する。

【0051】

同様に、異常検出部 32 から書換情報が通知されると、判定部 35 は、処理登録情報を参照して、通知された書換情報に示されるメモリ領域を利用して実行中の物体検出処理が存在するか否かを判定する。この場合、異常検出処理が第 1 の処理の一例となる。処理登録情報に、実行中の物体検出処理が利用するメモリ領域が登録されている場合、判定部 35 は、登録されているメモリ領域と、異常検出部 32 から通知された書換情報に示されるメモリ領域とが重なるか否かを判定する。そして判定部 35 は、実行中の物体検出処理が利用するメモリ領域と、書換情報に示されるメモリ領域の少なくとも一部が重なる場合、異常検出処理が利用する共有メモリ 241 のメモリ領域を利用して実行中の物体検出処理が存在すると判定する。この場合、実行中の物体検出処理は、第 2 の処理の一例となる。

10

【0052】

なお、処理登録情報に、共有メモリ 241 を利用して実行中の物体検出処理が登録されていない場合、判定部 35 は、異常検出処理が利用する共有メモリ 241 のメモリ領域を利用して実行中の物体検出処理は存在しないと判定する。また、実行中の物体検出処理が利用するメモリ領域と異常検出部 32 から通知された書換情報に示されるメモリ領域とが重ならない場合も、判定部 35 は、異常検出処理が利用するメモリ領域を利用して実行中の物体検出処理は存在しないと判定する。

20

【0053】

また、共有メモリ 241 を利用する他の処理が存在する場合、判定部 35 は、上記と同様の処理を行って、他の処理が利用する共有メモリ 241 のメモリ領域を利用して実行中の物体検出処理などが存在するか否かを判定すればよい。同様に、判定部 35 は、上記と同様の処理を行って、物体検出処理または異常検出処理が利用する共有メモリ 241 のメモリ領域を利用して実行中の他の処理が存在するか否かを判定すればよい。

【0054】

30

判定部 35 は、第 1 の処理が利用する共有メモリ 241 のメモリ領域を利用して実行中の第 2 の処理が存在するか否かの判定結果、及び、第 1 の処理及び第 2 の処理の優先度を処理制御部 36 へ通知する。

【0055】

処理制御部 36 は、判定部 35 から通知された判定結果を参照する。そして、第 1 の処理の優先度よりも高い優先度を有する第 2 の処理が存在する場合、処理制御部 36 は、第 2 の処理が終了するまで第 1 の処理を中断させる。そして処理制御部 36 は、優先度が相対的に高い第 2 の処理の終了後に、第 1 の処理に対してリトライ要求を通知して、第 1 の処理を最初からやり直させる。

【0056】

40

一方、第 1 の処理の優先度よりも低い優先度を有する第 2 の処理が存在する場合、処理制御部 36 は、第 1 の処理が終了するまで第 2 の処理を中断させる。そして処理制御部 36 は、第 1 の処理の終了後に、第 2 の処理に対してリトライ要求を通知して、第 2 の処理を最初からやり直させる。このように、処理制御部 36 は、実行が中断された相対的に優先度が低い処理についても、相対的に優先度が高い処理の終了後に再実行させるので、相対的に優先度が低い処理が実行されなくなることを抑制できる。

【0057】

本実施形態では、上記のように、異常検出処理の優先度は物体検出処理の優先度よりも高い。したがって、物体検出部 31 が共有メモリ 241 のメモリ領域に画像または中間データを書き込もうとしたときに、そのメモリ領域を参照して実行中の異常検出処理が存在

50

する場合、物体検出処理の実行は中断される。すなわち、処理制御部 36 は、物体検出部 31 に対して、物体検出処理の実行を中断させるとともに、物体検出処理が待機中であることを表すフラグをメモリ 22 に記憶させる。そして処理制御部 36 は、更新部 34 から異常検出処理が終了したことが通知されると、メモリ 22 を参照して、物体検出処理が待機中であることを表すフラグの有無を判定する。そのフラグがメモリ 22 に記憶されている場合、処理制御部 36 は、物体検出部 31 に対してリトライ要求を通知して、物体検出処理を最初からやり直させる。

【0058】

逆に、異常検出部 32 が共有メモリ 241 のメモリ領域に一連の画像または中間データを書き込もうとしたときに、そのメモリ領域を参照して実行中の物体検出処理が存在する場合も、物体検出処理の実行は中断される。すなわち、処理制御部 36 は、物体検出部 31 に対して、物体検出処理の実行を中断させるとともに、物体検出処理が待機中であることを表すフラグをメモリ 22 に記憶させる。そして処理制御部 36 は、更新部 34 から異常検出処理が終了したことが通知されると、メモリ 22 を参照して、物体検出処理が待機中であることを表すフラグの有無を判定する。そのフラグがメモリ 22 に記憶されている場合、処理制御部 36 は、物体検出部 31 に対してリトライ要求を通知して、物体検出処理を最初からやり直させる。このように、優先度が相対的に低い物体検出処理が実行中であっても、優先度が相対的に高い異常検出処理は、物体検出処理の終了を待つことなく実行される。

【0059】

物体検出処理及び異常検出処理以外に共有メモリ 241 を利用する処理が存在する場合も、処理制御部 36 は、上記と同様の処理を行って、優先度が相対的に高い方の処理を優先的に実行させればよい。

【0060】

図 4 は、リソース管理処理の概要を説明する図である。この例では、共有メモリ 241 を利用する 4 種類の処理が存在する。図 4 では、これら 4 種類の処理は、それぞれ、スレッド 1 ~ スレッド 4 として表される。そして各処理の優先度は、スレッド 2 > スレッド 1 > スレッド 3 > スレッド 4 の順に低くなる。すなわち、スレッド 2 の優先度が最も高く、スレッド 4 の優先度が最も低い。また、図 4 において、縦軸は経過時間を表す。そしてスレッドごとに示されるブロック 401 ~ 404 は、そのスレッドの演算が実行されている期間を表す。この例では、スレッド 1 ~ スレッド 4 は、同じ時刻  $t_0$  にて並列に実行が開始される。そしてスレッド 1 が共有メモリ 241 のアドレス  $0x10$  から始まるメモリ領域に記憶されているデータを最初に参照する。次に、スレッド 2 が共有メモリ 241 のアドレス  $0x10$  から始まるメモリ領域を参照する。そしてスレッド 2 が、時刻  $t_1$  にてアドレス  $0x10$  から始まるメモリ領域のデータを書き換える。そのため、スレッド 2 によりデータが書き換えられたメモリ領域を参照し、かつ、スレッド 2 の優先度よりも低い優先度を持つスレッド 1 の実行は中断される。そしてスレッド 2 の実行が終了した時刻  $t_3$  において、スレッド 1 は最初から再度実行される。

【0061】

また、スレッド 2 がアドレス  $0x20$  から始まるメモリ領域を参照した後、時刻  $t_2$  にて、スレッド 3 がそのアドレス  $0x20$  から始まるメモリ領域のデータの書き換えを試みる。しかし、スレッド 3 の優先度はスレッド 2 の優先度よりも低いため、アドレス  $0x20$  から始まるメモリ領域のデータの書き換えは行われず、スレッド 2 の実行が終了した時刻  $t_3$  において、スレッド 3 は最初から再度実行される。

【0062】

さらに、スレッド 4 は、その実行中において共有メモリ 241 のアドレス  $0x30$  から始まるメモリ領域にアクセスするものの、そのメモリ領域は、他の処理が参照するメモリ領域と重なっていない。そのため、スレッド 4 の優先度は他の処理の優先度よりも低いものの、中断されることなく最後まで実行される。

【0063】

10

20

30

40

50

図5は、プロセッサ24により実行される、リソース管理処理の動作フローチャートである。プロセッサ24は、所定の周期ごとに、以下の動作フローチャートに従ってリソース管理処理を実行すればよい。

【0064】

プロセッサ24の更新部34は、物体検出部31または異常検出部32から処理開始情報を受け取ったか否か判定する(ステップS101)。更新部34は、処理開始情報を受け取った場合(ステップS101 - Yes)、メモリ22から処理登録情報を読み出す。そして更新部34は、処理開始情報に示される、実行が開始される処理の優先度及び共有メモリ241の利用するメモリ領域を処理登録情報に追加する(ステップS102)。

【0065】

ステップS102の後、あるいは、更新部34が処理開始情報を受け取っていない場合(ステップS101 - No)、更新部34は、物体検出部31または異常検出部32から処理終了通知を受け取ったか否か判定する(ステップS103)。更新部34は、処理終了通知を受け取った場合(ステップS103 - Yes)、メモリ22から処理登録情報を読み出す。そして更新部34は、その処理終了通知において終了したことが通知された処理(この例では、物体検出処理または異常検出処理)についての優先度及び利用するメモリ領域を表す情報を処理登録情報から削除する(ステップS104)。さらに、プロセッサ24の処理制御部36は、待機中の処理が存在するか否か判定する(ステップS105)。そして待機中の処理が存在する場合(ステップS105 - Yes)、処理制御部36は、待機中の処理を実行するよう、リトライ要求を通知する(ステップS106)。

【0066】

ステップS106の後、プロセッサ24の判定部35は、物体検出部31または異常検出部32から書換情報を受け取ったか否か判定する(ステップS105)。また、更新部34が処理終了通知を受け取っていない場合(ステップS103 - No)、あるいは、待機中の処理が存在しない場合(ステップS105 - No)も、判定部35は、書換情報を受け取ったか否か判定する(ステップS107)。

【0067】

判定部35は、書換情報を受け取った場合(ステップS107 - Yes)、処理登録情報を参照する。そして判定部35は、書換情報に含まれる識別情報で示される処理(第1の処理)が書き換えようとするメモリ領域を利用して実行中の他の処理(第2の処理)が存在するか否か判定する(ステップS108)。

【0068】

第2の処理が存在する場合(ステップS108 - Yes)、処理制御部36は、処理登録情報を参照して、第1の処理の優先度が第2の処理の優先度よりも低いかなどを判定する(ステップS109)。第1の処理の優先度が第2の処理の優先度よりも低い場合(ステップS109 - Yes)、処理制御部36は、第1の処理の中断を通知する(ステップS110)。例えば、第1の処理が物体検出処理であり、第2の処理が異常検出処理である場合、処理制御部36は、物体検出部31に対して物体検出処理の中断を通知する。

【0069】

一方、第1の処理の優先度が第2の処理の優先度よりも高い場合(ステップS109 - No)、処理制御部36は、実行中の第2の処理の中断を通知する(ステップS111)。例えば、第1の処理が異常検出処理であり、第2の処理が物体検出処理である場合、処理制御部36は、物体検出部31に対して物体検出処理の中断を通知する。そして処理制御部36は、第1の処理によるメモリ領域の書き換えを許可することを通知する(ステップS112)。

【0070】

また、ステップS108において、第2の処理が存在しない場合(ステップS108 - No)も、処理制御部36は、第1の処理によるメモリ領域の書き換えを許可することを通知する(ステップS112)。

【0071】

10

20

30

40

50

ステップ S 1 1 0 または S 1 1 2 の後、プロセッサ 2 4 は、リソース管理処理を終了する。

【 0 0 7 2 】

以上に説明してきたように、このリソース管理装置は、複数の処理のうち、共有リソースを用いて実行中の処理の優先度及びその処理が利用する共有リソースの範囲を、処理登録情報を用いて管理する。すなわち、このリソース管理装置は、共有リソースを利用して実行中の第 1 の処理が共有リソースに記憶されているデータを書き換える場合、処理登録情報を参照する。そしてこのリソース管理装置は、第 1 の処理が利用する共有リソースの範囲を利用して実行中の第 2 の処理が存在するか否かを判定する。このリソース管理装置は、第 1 の処理の優先度よりも高い優先度を有する第 2 の処理が存在する場合、第 2 の処理が終了するまで第 1 の処理を中断させる。一方、このリソース管理装置は、第 1 の処理の優先度よりも低い優先度を有する第 2 の処理が存在する場合、第 1 の処理が終了するまで第 2 の処理を中断させる。したがって、このリソース管理装置は、共有リソースを使用する複数の処理がその共有リソースの使用において競合した場合、複数の処理のうちの優先度の高い方の処理が優先的に実行されることを保障できる。

10

【 0 0 7 3 】

このリソース管理装置は、車両制御システム以外に適用されてもよい。このリソース管理装置は、互いに非同期で、優先度が指定される複数の処理に対して共有リソースを割り当てることでその複数の処理が実行される様々な装置またはシステムに適用することが可能である。

20

【 0 0 7 4 】

以上のように、当業者は、本発明の範囲内で、実施される形態に合わせて様々な変更を行うことができる。

【符号の説明】

【 0 0 7 5 】

- 1 車両制御システム
- 2 カメラ
- 3 電子制御装置（リソース管理装置）
  - 2 1 通信インターフェース
  - 2 2 メモリ
  - 2 3 バッファメモリ
  - 2 4 プロセッサ
  - 3 1 物体検出部
  - 3 2 異常検出部
  - 3 3 車両制御部
  - 3 4 更新部
  - 3 5 判定部
  - 3 6 処理制御部

30

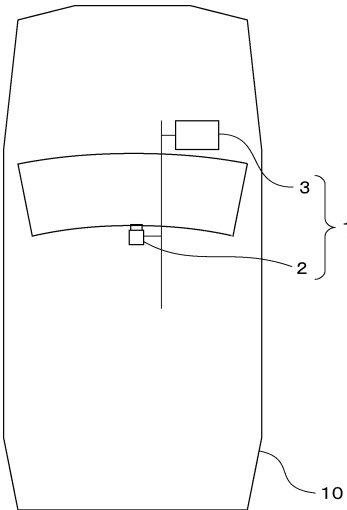
40

50

【図面】

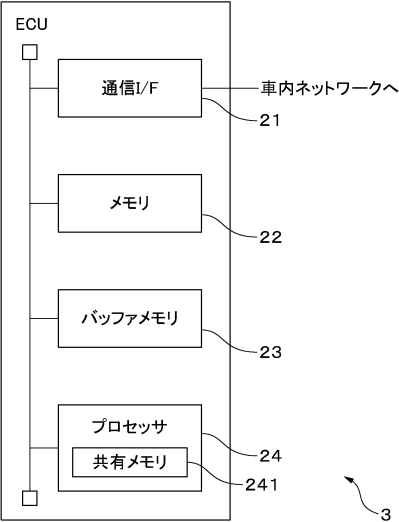
【図 1】

図1



【図 2】

図2



10

20

【図 3】

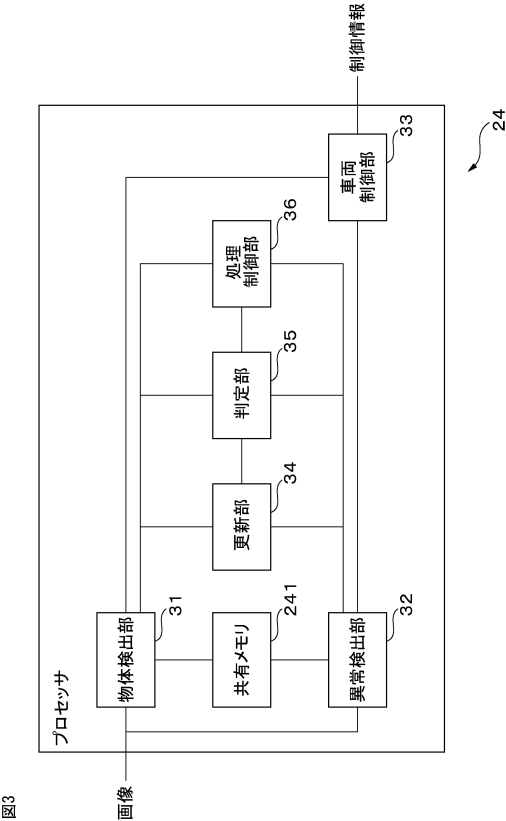


図3

【図 4】

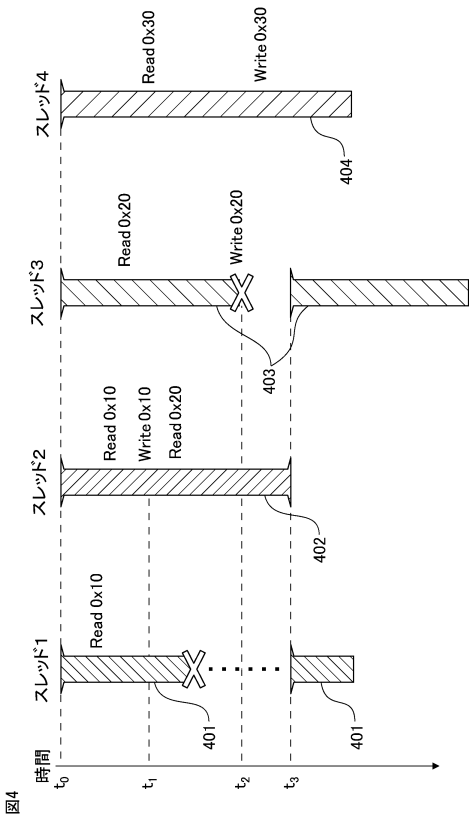


図4

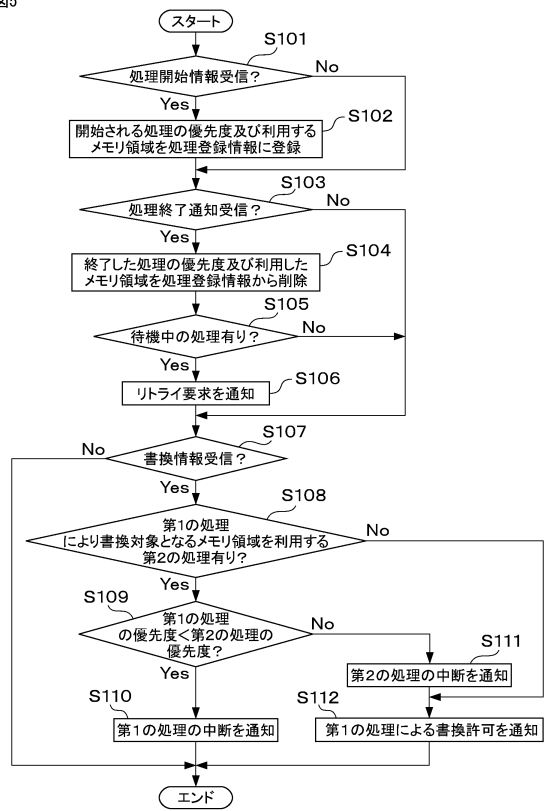
30

40

50

【 図 5 】

図5



10

20

30

40

50

---

フロントページの続き

- (72)発明者 橋本 高志良  
東京都中央区日本橋室町三丁目 2 番 1 号 ウーブン・コア株式会社内
- 審査官 三坂 敏夫
- (56)参考文献 特開昭 6 0 - 1 0 1 6 6 6 ( J P , A )  
特開平 1 1 - 1 9 4 9 5 1 ( J P , A )  
国際公開第 2 0 0 9 / 1 5 7 1 3 6 ( W O , A 1 )
- (58)調査した分野 (Int.Cl. , D B 名)  
G 0 6 F 9 / 4 5 5 - 9 / 5 4