

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5249969号
(P5249969)

(45) 発行日 平成25年7月31日 (2013. 7. 31)

(24) 登録日 平成25年4月19日 (2013. 4. 19)

(51) Int. Cl.

F I

G 0 6 N 5/04 (2006. 01)

G 0 6 N 5/04 5 6 0 M

G 0 6 T 7/00 (2006. 01)

G 0 6 T 7/00 3 5 0 Z

請求項の数 24 外国語出願 (全 33 頁)

(21) 出願番号 特願2010-30845 (P2010-30845)
 (22) 出願日 平成22年2月16日 (2010. 2. 16)
 (65) 公開番号 特開2010-262625 (P2010-262625A)
 (43) 公開日 平成22年11月18日 (2010. 11. 18)
 審査請求日 平成22年4月20日 (2010. 4. 20)
 (31) 優先権主張番号 09153896.7
 (32) 優先日 平成21年2月27日 (2009. 2. 27)
 (33) 優先権主張国 欧州特許庁 (EP)
 (31) 優先権主張番号 10151739.9
 (32) 優先日 平成22年1月27日 (2010. 1. 27)
 (33) 優先権主張国 欧州特許庁 (EP)

(73) 特許権者 503113186
 ホンダ リサーチ インスティテュート
 ヨーロッパ ゲーエムベーハー
 Honda Research Inst
 itute Europe GmbH
 ドイツ連邦共和国 デー 6 3 0 7 3 オ
 ッフェンバッハアムメイン カール・レギ
 エン・シュトラッセ 30
 (74) 代理人 110000246
 特許業務法人 OFH 特許事務所
 (72) 発明者 ユリアン・エガート
 ドイツ国 6 3 1 7 9 オーベルシャウゼン
 、パーター・デルプ・シュトラッセ 33

最終頁に続く

(54) 【発明の名称】 知識ベースの選択的な視覚分析のための人工視覚システムおよび方法

(57) 【特許請求の範囲】

【請求項 1】

視覚センサが検出した世界、物体及び当該物体の特性、当該世界における感覚事象、現在の視覚タスク、感覚コンテキスト、および知覚装置についての蓄積された知識を使用して、コンピュータビジョンシステムの情報処理を適応化および最適化する方法であって、

前記コンピュータビジョンシステムは関係記憶を有し、当該関係記憶は、ノードと当該ノード間の関係を示すリンクとを含むグラフにより表わされた情報を含み、前記ノードは、物体の一般的特性である物体特性、及び当該物体特性の情報取得に用いる視覚サブタスクの情報を表わし、前記リンクは前記ノード間の従属関係の情報を含み、前記物体特性を表わすノードには当該ノードの処理である当該物体特性の情報取得の方法を表わすルーチンが関連付けられ、前記視覚サブタスクを表わすノードには当該ノードの処理である当該視覚サブタスクの実行方法を表わすルーチンが関連付けられており、

前記コンピュータビジョンシステムは、さらに、各ノードについて、当該ノードの処理の結果を表わすノード情報と、当該ノード情報の有効性すなわち更新の必要有無を表わすノード状態と、を記憶しており、

前記方法は、

a) 対象物の物体特性を表すノードであって、所与の視覚タスクを実行するため当該物体特性の情報の取得を必要とするノードを特定するステップと、

b) 前記関係記憶に基づいて、前記特定された各ノードについて、当該特定されたノードと当該ノードの処理に必要な他のノードとを含むグラフを生成するステップであって、

10

20

当該グラフは前記特定されたノードを初期ノードとして含み、前記初期ノードと前記他のノードとが、ノード間の従属性を表わすリンクにより接続されている、ステップと、

c) ステップbにおいて生成された前記グラフが表す情報フローに従い、物体特性の情報を得るためのルーチンを構成するステップと、

d) ステップcにおいて構成されたルーチンを実行し、前記特定されたノードが表す物体特性の情報を取得して、前記所与の視覚タスクを実行するステップと、

を有し、

前記ステップbは、

b 1) 前記特定されたノードを初期ノードとし、当該初期ノードに対し、当該初期ノードの処理に直接的な影響を与える他のノードを、前記関係記憶に記憶されている当該初期ノードに対する従属関係を表わすリンクを用いて付加して、前記グラフを生成するステップと、

b 2) 前記付加したノードから、処理に用いるべき前記ノード情報を与えるノードを選択するステップと、

b 3) 前記選択したノードから、ノード情報の更新が必要なノードを識別するステップと、

b 4) 前記識別された更新が必要な各ノードに対し、当該ノードの処理に直接的な影響を与える他のノードを、前記関係記憶に記憶されているノード間の従属関係を表わすリンクを用いて付加するステップと、

b 5) ステップb 2において付加すべきノードがなくなるか、又はステップb 3においてノード情報の更新が必要なノードがなくなるまで、ステップb 2～b 4を繰り返すステップと、

を有し、

前記グラフが表す前記情報フローは、前記グラフ内の前記リンクが表す従属関係に従って前記初期ノードに向かう、当該グラフ内の前記選択された各ノードからのノード情報の流れであって、

前記特定されたノードが表す物体特性の情報を得るための前記ルーチンは、前記ノード情報の更新を要しないノードの当該ノード情報へのアクセス、及び又は前記ノード情報の更新を要するノードに関連付けられたルーチンの実行を、前記情報フローに従う順序で行うように構成される、

方法

【請求項 2】

前記世界は、音響センサ、触覚センサ、および/またはソフトウェアないしハードウェアインターフェースを介して受信された情報により検知される、請求項 1 に記載の方法。

【請求項 3】

前記方法は、さらに、記憶パラメータ、システムパラメータ、事象パラメータ、グラフパラメータ、および/またはセンサパラメータを適応させる、請求項 1 または 2 に記載の方法。

【請求項 4】

前記感覚事象の記憶が表現される、請求項 1 乃至 3 のいずれか 1 項に記載の方法。

【請求項 5】

現在の視覚的な感覚コンテキストの記憶は、前記感覚事象の接続グラフで表現される、請求項 1 乃至 4 のいずれか 1 項に記載の方法。

【請求項 6】

前記コンピュータビジョンシステムは、前記ステップb 2において、リソース競合が回避されるように前記ノードを選択することにより、当該コンピュータビジョンシステムが備える処理リソースを柔軟に適応させる、請求項 1 乃至 5 のいずれか 1 項に記載の方法。

【請求項 7】

前記コンピュータビジョンシステムは、前記ステップbにより生成された前記グラフが示すノード間の従属関係に基づいて前記関係記憶を改善する、請求項 1 乃至 6 のいずれか

10

20

30

40

50

1 項に記載の方法。

【請求項 8】

前記コンピュータビジョンシステムは、前記各視覚サブタスクについての、必要な視覚パラメータにアクセスする方法及び当該パラメータを計測する方法についての情報を含む視覚サブタスク情報を記憶し、

前記最適化は、前記ステップ b 2 において、前記視覚サブタスク情報に加えて所与のタスク要件にも基づいて、前記ノードを選択することにより行われる、請求項 1 乃至 7 のいずれか 1 項に記載の方法。

【請求項 9】

前記最適化は、前記ステップ b 2 において、タスク固有のゲインおよび / またはコスト要件に基づいて、前記ノードを選択することにより行われる、請求項 1 乃至 8 のいずれか 1 項に記載の方法。

【請求項 10】

前記最適化は、前記ステップ b 2 において、期待される情報フローおよび現在の情報フローに関連する不確実性情報に基づいて、および / または、確率的手法を用いて、前記ノードを選択することにより行われる、請求項 1 乃至 9 のいずれか 1 項に記載の方法。

【請求項 11】

前記視覚サブタスク情報は、前記視覚サブタスクの実行中に収集された実行時の統計に基づいて漸進的に調節され、前記実行時の統計は、経過時間及び又は処理結果の精度を含むものである、請求項 1 乃至 10 のいずれか 1 項に記載の方法。

【請求項 12】

前記リンクは、複数の従属関係タイプのうちの一の従属関係タイプを示すものとしてことができ、複数の前記従属関係タイプが用いられることにより、前記情報フロー、及び、前記特定されたノードが表す物体特性についての情報取得のための前記ルーチンにおける前記アクセス及び又は前記実行の順序を、様々に変化させることを可能とし、

前記各ノードは、当該ノードの処理の結果の有効性を監視して、当該ノードについて保存された前記ノード状態を更新する、

請求項 1 乃至 11 のいずれか 1 項に記載の方法。

【請求項 13】

前記リンクは、有向リンクであり、当該有向リンクの始点に接続された前記ノードの処理結果が、当該有向リンクの終点に接続された前記ノードの処理結果により、直接的に影響されることを示すものである、請求項 1 ないし 12 のいずれか一項に記載の方法。

【請求項 14】

前記リンクが表す前記従属関係タイプの一つは、接続された 2 つのノードの処理に厳密な順序を課すものであって、一のノードの処理に他のノードの有効な結果を必要とすることを表わしている、請求項 1 乃至 13 のいずれか 1 項に記載の方法。

【請求項 15】

前記リンクが表す前記従属関係タイプの一つは、接続された一のノードの処理が他のノードの処理結果を改善し得るものの、当該接続された一のノードの処理の実行は必ずしも必要となれないという、“選択的”な従属関係を示すものである、

請求項 1 乃至 14 のいずれか 1 項に記載の方法。

【請求項 16】

前記リンクが表す前記従属関係タイプの一つは、一の共通するノードが複数の他のノードと接続されており、当該他のノードの少なくとも一つが有効な処理結果を持ったときに前記共通するノードの処理の実行が可能となることを表わすものである、

請求項 1 乃至 15 のいずれか 1 項に記載の方法。

【請求項 17】

前記ノード状態により表わされる前記ノード情報の有効性は、前記ノードの処理の、計測の信頼性、取得されたデータの品質、または最後に計測してから経過時間、によって決定される、請求項 1 乃至 16 のいずれか 1 項に記載の方法。

【請求項 18】

前記コンピュータビジョンシステムは、前記各ノードについて、マークされたか否かの情報を記憶し、

前記ステップ b 2 は、さらに、

b 2 1) 前記選択されたノードの中に、既にマークが付された第 2 のノードが含まれていれば、

前記リンクが表す従属関係に従い、当該マークが付されたノードに代えて、当該ノードと同時にグラフに追加された他のノードの中から、当該マークが付されたノードの代替となるノードを特定し、当該代替となるノードを選択するか、又は、

当該マークが付されたノードから前記初期ノードの方向へ前記リンクに沿って上流にあるノードの中から、同時に追加された代替となるノードを持つものを特定し、前記マークが付されたノードに代えて当該特定したノードの代替のノードを選択して、当該特定したノードと当該特定したノードより下流にある全てのノードとを削除する、ステップと、

b 2 2) 前記選択された各ノードにマークを付すステップと、
を有する、

請求項 1 ないし 17 のいずれか一項に記載の方法。

【請求項 19】

前記ノード状態が表すノード情報の有効性は、さらに、前記リンクに沿って下流にあるノードのノード情報の有効性により定まり、前記ノード情報は、前記下流にある少なくとも一つのノードのノード情報が無効のとき、無効と決定される、請求項 18 に記載の方法。

【請求項 20】

前記ステップ b 5 の処理と、前記選択されたノードのノード情報へのアクセスとが、非同期に又は並行して実行される、請求項 1 乃至 19 のいずれか 1 項に記載の方法。

【請求項 21】

前記ステップ b 5 の処理と、前記選択されたノードに関連付けられたルーチンの実行とが、並行して実施される、請求項 1 乃至 20 のいずれか 1 項に記載の方法。

【請求項 22】

請求項 1 乃至 21 のいずれか 1 項に記載の方法を、コンピューティング・ユニットに実行させる、コンピュータプログラム。

【請求項 23】

請求項 1 乃至 21 のいずれか 1 項に記載の方法を用いるコンピュータビジョンシステムであって、ロボット又は車両を含む、可動のプラットフォームに接続されており、データが与えられたことに応じて実行される所与の視覚計算と、リソースに関し最適化された視覚演算とを用いるものであって、所定のタスクに関連する視覚的側面に選択的に集中して処理を行う、コンピュータビジョンシステム。

【請求項 24】

請求項 23 に記載のコンピュータビジョンシステムを備える、車両またはロボット。

【発明の詳細な説明】**【技術分野】****【0001】**

本発明の背景は、概して、人工視覚システム、すなわち、視覚検知手段（たとえば、ビデオカメラ）およびコンピューティング・ユニットを用いて実現される後続の処理ステージを備えるシステムの分野である。該処理ステージは、視覚的に分析されたシーン(scene)の表現を出力し、この出力は、その後、車両（自動車、飛行機等）のパーツ、またはロボット（好ましくは、たとえば人間型ロボット（一例は、ホンダのアシモ（ASIMO）ロボット）といった自律型ロボット）のような、様々な実体(actor)を制御するのに供給されることができる。

【背景技術】

【 0 0 0 2 】

この進歩的な視覚システム（ビジョンシステム）は、人間のオペレータを補助するように設計されたシステムを含め、地上、水上、および（または）大気中を行く移動体（車両）のためのシステムのような、様々な他の領域にも適用されることができる。ここで開示される方法および（または）システムを、技術的（たとえば、電子的）システムが自律的に学習し、該システムに提示される対象物の特徴および（または）特性（たとえば、サイズ、距離、絶対的位置、他の対象物に対する相対的位置、空間的なアラインメント、相対的な移動、速度、および（または）方向、他の関連する対象物の特徴、または特徴のパターン等）を取得することを必要とするどのような場合にも、一般的に使用することができる。

10

【 0 0 0 3 】

このような、コンピュータに基づく人工視覚システムは通常、連結したいくつかの処理ステップに依拠しており、これは、1つ以上のカメラ・デバイスからの入力から始まり、これがその後、順番に、フィルタリングおよび選択アルゴリズムによって処理される。該連結した処理の最後において、結果が、コンピュータ・ビジョン(computer vision)のシステムを使用するアプリケーション（適用業務）に関心のある側面について、低次元の表現となって現れる。

【 0 0 0 4 】

既知の現在のコンピュータ・ビジョン・システムに特有なのは、それらの、フィードフォワード特性である。この基本的な事象は、新しい画像の到着によって起動されて処理され、そして結果が、次の処理ステージに渡される。典型的な順次処理のチェーン（連なり）は、画像取得、事前処理、特徴抽出、検出／セグメント化、および高レベル処理からなる。主要な情報フローがフィードフォワードであるとしても、それらのステップのそれぞれは、いくつかの繰り返しを含むかもしれない、また、再帰の情報フローを必要とするかもしれない。再帰の情報フローは、異なるステージ間でも出現するかもしれないが、そのようなシステムは制御するのがより困難となるために、これは滅多にないケースである。

20

【 0 0 0 5 】

既知のコンピュータ・ビジョン・システムの他の特性は、これらが、貧弱な内部表現で動作すること、すなわち、これらが、非常にコンパクトな形態（たとえば、対象物のアイデンティティのID、または他の小規模なデータ）で表現されることのできる、視覚的入力の特定の側面に専念しており、付加的な、よりグローバルな知識（たとえば、視覚的なシーンのコンテキスト全体、または対象物の感覚的な外観）をほとんど使用しないということである。

30

【 0 0 0 6 】

これにより、特定の領域に非常に特化されたシステムとなってしまう、他に対する柔軟性がない。それに対し、生物学的な視覚システム（視覚系）は、汎用的であり、様々な視覚タスクに適応している。さらに、人間によるシーンの解釈では、そのシーンおよび対象物についての多くの事前知識が、それらの視覚的および物理的な特性と共に、視覚処理のための制約ないし変調（調整）パラメータとして取り入れられる。さらに、多くの最新技術のシステムに比べ、人間における情報取得プロセスは能動的であると考えられる。すなわち、人間は、現在の視覚タスクのために、関係のある対象物について視覚シーンを能動的にスキャンしている。

40

【 0 0 0 7 】

動作中に異なる視覚サブタスクに十分適応可能な一般的な視覚システムについて、シーンについての既存の（偏見のない）知識から、視覚処理のリソース（資源）を調整して編成することは、中心的な要素となる。なぜならば、該システムの同じ要素が、様々な目的のために用いられなければならないからである。これは、制限されたリソースの効率的な使用のために、視覚システムにおける情報フローの特別な表現および特別な制御を必要とする。

【 0 0 0 8 】

50

視覚的入力(visual input)により支配される静的な情報フローの手法に依拠する既知のコンピュータ・ビジョン・システムは、視覚サブタスクの数が増えるにつれ、その拡張性が低下する。それに対し、現在のコンテキストおよび視覚タスクに従って、その処理フローおよびリソースを柔軟に適應させるシステムの場合には、全体的な計算上の要件だけでなく、必要なアーキテクチャの複雑性の観点から、有利な点を提示する(多くのモジュールを効率的に再使用することができるので)。ここで、シーンの能動的な(アクティブな)検査が、主要な役割を果たす。

【 0 0 0 9 】

この発明はまた、コンパイラによる最適化(compiler optimization)およびグラフ構造に基づく計算で広い分野に知られているように、スケジューリング・メカニズムに関連する(より詳細には、当該明細書を参照のこと)。

10

【 0 0 1 0 】

さらに、この作業は、ロボット工学およびコンピューショナル・インテリジェンス(計算知能)の分野において、たとえばP O M D P (Partially Observable Markov Decision Processes (部分観測マルコフ決定過程))により与えられるような、感覚アクション(sensory action)のための最適プランニング(optimal planning)および逐次的意思決定(sequential decision making)にアプローチする確率的手法に基づいたアルゴリズムに関連する。具体的には、後述する参考文献(14, 15)には、能動的かつ動的な手法で視覚ワークフローを適應させることが提案されている。該文献(14, 15)では、非常にシンプルなシーンの構成において、現在のシーンで所定の特徴を発見する確率によって、選択プロセスが排他的に導かれる。多彩なコンテキストについての知識は、対象物の構成や感覚プロセスについての長期知識(long term knowledge)または幅広い感覚コンテキストのいずれの観点からも、使用されない。

20

【 0 0 1 1 】

コンピュータ・ビジョンのための方法は、その情報処理フロー、その処理リソース、およびそのパラメータを、知覚可能な世界、その対象物とその特性、現在の視覚タスク、感覚コンテキスト、および知覚装置(perception apparatus)自体に関する明示的知識(explicit knowledge)を用いて適應し、よって、制約されたリソースでリアルタイムで動作しなければならない、かつ様々な経験を考慮する(これは、経験から明示的に取り入れられる、または学習される)柔軟な視覚システムについて、非常に多くの利点を提供する。

30

【 0 0 1 2 】

典型的には、入力信号または入力パターンはセンサから受け取られ、これが、ハードウェアユニットおよびソフトウェア構成要素によって処理される。出力信号または出力パターンが得られ、これは、さらなる処理のため(たとえば、視覚化(visualization)のため)他のシステムへの入力となることができる。入力信号は、1または複数のセンサ(たとえば、視覚的または音響的な検知であり、ソフトウェアインターフェースによるものだけでなく、ハードウェアインターフェースによるものも含む)により供給されることができる。出力パターンは、ソフトウェアおよび(または)ハードウェアインターフェースを介して出力されることができ、または他の処理ユニットないし実体に転送されることができ、これは、ロボットや車両のアクションないし挙動に影響を与えるよう使用されることができる。

40

【 0 0 1 3 】

本発明によって必要とされる計算および変換(transformation)を、1または複数のプロセッサ(C P U s)、信号処理ユニット、または他の計算、処理、コンピュータのハードウェアおよび(または)ソフトウェアのような処理手段によって実行されることができる。これらの処理手段はまた、並列処理用に適應されることができる。処理および計算を、標準的な市販の(O T S)ハードウェアまたは特別に設計されたハードウェア構成要素上で実行することができる。プロセッサのC P Uは、該計算を実行することができ、また、メインメモリ(R A M, R O M)、制御ユニットおよび論理演算装置(A L U)を含むことができる。また、C P Uは、特化されたグラフィックプロセッサをアドレスすること

50

もでき、該グラフィックプロセッサは、必要な計算を扱うのに専用のメモリおよび処理機能を提供する。

【 0 0 1 4 】

本発明は、データ記憶手段を使用する関係意味記憶（リレーショナル・セマンティック・メモリ (relational semantic memory)）を利用する。該データ記憶手段は、処理、結果、およびデータ構造のために、取得され必要とされる情報および（または）データを記憶するのに用いられる。該記憶手段はまた、事象およびそこから得られた知識に関連する、観察による記録を格納ないし記憶することを可能にし、これにより、将来の事象についてのアクションおよびリアクションに影響することができる。

【 0 0 1 5 】

該記憶装置は、ハードディスク（SSD、HDD）、RAMおよび（または）ROMのようなデバイスにより提供されることができ、これらは、フロッピーディスク、CD-ROM、テープ、USBドライブ、スマートカード、ペン・ドライブ等のような他の（携帯可能な）記憶媒体によって提供されることができ、よって、本発明に従う方法をコード化したプログラムと、該進歩的なシステムのアプリケーションにおいて、または該アプリケーションのために、または該進歩的な方法において、または該方法のために、取得され、処理され、学習され、必要とされるデータとを、それぞれ記憶媒体に格納することができる。

【 0 0 1 6 】

特に、本発明により記述される方法を、該方法を実行するシステムないしデバイスに命令するため、物理的な（たとえば、携帯可能な）記憶媒体上に、ソフトウェアプログラム製品として提供することができ、該記憶媒体を、処理システムまたはコンピューティング・デバイスに該プログラム製品を転送するのに用いることができる。さらに、該方法を、コンピューティング・デバイス上で直接的に実現してもよいし、または、該コンピューティング・デバイスと組み合わせて提供してもよい。

【 従来技術 】

【 0 0 1 7 】

既知の技術的なコンピュータ・ビジョン・システムは、対象物認識、識別および検出、対象物追跡、画像復元、動き推定、およびシーンの再構成のような多数のタスクを解決するように設計されている。

【 0 0 1 8 】

コンピュータ・ビジョンの良好に定義された様々なタスクを解決するための既知の方法がいくつか存在し、ここで、該方法の多くは、非常にタスク指向的 (task specific) なものであり、広い範囲のアプリケーションについて一般化できることはほとんど無い。該方法およびアプリケーションの多くは、基礎研究の段階では静的なものであるが、より多くの方法が商業製品に進出してきており、そこでは、複雑なタスク（たとえば、医療画像の領域、または産業量プロセスの品質制御および計測）を解決できるより大きいシステムの一部を構成する。多くの実用的なコンピュータ・ビジョンのアプリケーションでは、コンピュータは、特定のタスクを解決するよう予めプログラムされているが、学習に基づく方法は、現在、より一般的なものとなってきた。

【 0 0 1 9 】

コンピュータ・ビジョンの従来の良好に定義されたタスクは、通常、感覚周辺 (sensory periphery) の近くに位置付けられており、すなわち、それらは、低レベルのビジョンを扱うにすぎない。いわゆる“高レベル”ビジョンは、データがアプリケーション固有の前提を満たすことを検証するため、またはアプリケーション固有のパラメータ（たとえば、対象物の姿勢または大きさ）を推定するため、様々な情報を組み合わせることに対処する。

【 0 0 2 0 】

それにもかかわらず、現在のコンピュータ・ビジョン・システムは、固定された情報フロー方式に基づいている。以下では、我々は、（たとえば、追跡、セグメント化、対象物

10

20

30

40

50

分類等に必要な部分的な処理ステップによって与えられるように)いくつかの基本的な視覚サブタスク、または視覚サブルーチンが利用可能であると前提とする。このタイプのサブタスクを扱う非常に多くの作業(ワーク)が以前からある。他方で、この作業は、コンパイラによる最適化およびグラフ構造に基づく計算といった広い分野で既知のように、スケジューリング・メカニズムに関連している。

【0021】

他方、下記の非特許文献1, 2(これは、後述する参考文献の(12, 13)に相当する)のような、所与の視覚タスクについて視覚処理プログラムをコンパイルするAIAプローチが存在し、これは、問題の定式化(problem formalization)から開始する。これらのシステムは、フィードフォワードの手法で動作し、視覚アルゴリズムについてのそれらの知識を使用して、プログラム構築の処理を導く。それにもかかわらず、AIAアプローチの目標は、固定された視覚アプリケーションを構成することであり、すなわち、視覚処理についての知識は取り入れられるものの、最終的な結果は、再び、固定した情報フローを有するシステムであり、この場合、上記の貧弱な拡張性に関する考察がすべてあてはまる。

10

【先行技術文献】

【非特許文献】

【0022】

【非特許文献1】Clouard R., Elmoataz A., Porquet C. & Revenu M.: Borg: A Knowledge-Based System for Automatic Generation of Image Processing Programs. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1999. Vol. 21 (2).

20

【非特許文献2】Shekhar C., Moisan S., Vincent R., Burlina P. & Chellappa R.: Knowledge-based control of vision systems. Image and Vision Computing. 1999. Vol. 17.

【発明の概要】

【発明が解決しようとする課題】

【0023】

本発明の目的は、独立請求項に従う方法およびシステムによって達成される。利点を備える実施形態は、従属請求項において規定される。

【課題を解決するための手段】

30

【0024】

本発明は、その情報処理フロー、その計算リソース、およびその計算上のパラメータを、視覚センサによって検知された知覚可能な世界、その対象物とその特性、現在の視覚タスク、感覚コンテキスト、および知覚装置自体に関する明示的知識を用いて適応させる、コンピュータ・ビジョンのための方法を提供する。該方法は、視覚的に知覚可能な対象物のような感覚事象を、機能的な関係を表す直接リンクによって他のノードに接続される、計測可能な感覚パラメータを備えたノードエンティティとして表現するステップと、現在の視覚的な感覚コンテキストを、感覚事象の接続されたグラフで表現するステップと、空間サブグラフおよび該サブグラフのノードとリンクに結びつけられたローカルプロセスにおいて、上記ステップの感覚パラメータにアクセスして計測する方法についての視覚サブタスク情報を表現するステップと、上記ステップからの、いわゆる“関係記憶(リレーショナル・メモリ)”からの情報を用いることによって、実行時に計算リソースを動的に割り振って情報フローを編成するステップと、該リソース割り振りおよび情報フローを、上記ステップからのグラフ構造に従って最適化するステップと、グラフの対応するノードおよびリンクのエンティティにおいて、感覚計測から取得した推定値を記憶し、これにより、上記ステップからの情報を更新するステップと、たとえばさらなる視覚対象物についての情報、および一般的には、上記ステップからの一般的な感覚コンテキストにおける追加および修正についての情報を表す新しいグラフ構造を漸進的に適応して作成するステップと、前のステップに戻り、知覚された世界およびその感覚プロセスについての知識の表現を改良するために、その処理リソースを柔軟に適応すると共に、その関係記憶を改善(リ

40

50

ファイン)するシステムで、前のステップを繰り返すステップと、を含む。

【0025】

上記の知覚可能な世界は、音響センサ、触覚センサ、および(または)ソフトウェアないしハードウェアのインターフェースを介して受け取った情報によって検知されることができる。

【0026】

該方法は、また、記憶パラメータ、システムパラメータ、事象パラメータ、グラフパラメータ、および(または)センサパラメータを適応させることができる。

【0027】

感覚事象の記憶(メモリ)を表現することができる。

10

【0028】

現在の視覚的な感覚コンテキストの記憶を、感覚事象の接続されたグラフによって表現することができる。

【0029】

計算リソースは、並列処理に適応されることのできる処理手段であることができる。

【0030】

システムは、検知された世界の表現を改良するために、上記ステップから、その処理リソースを柔軟に適応し、および(または)その関係記憶を改善(リファイン)する。

【0031】

最適化は、上記視覚サブタスク情報に加えて、タスク要件を考慮することで行われることができる。

20

【0032】

タスク要件は、時間対精度の調停、リソースの衝突、同じ感覚デバイスを要する異なる計測、順番および機能的な従属性、どの視覚サブタスクが、他より先行しなければならないか、どれが他からの情報を用いるか、および、以前の動作中における経験から編集されたサブタスク情報、のうちの少なくとも1つであることができる。

【0033】

以前の動作からの経験は、期待される精度、情報のゲイン、結果を得るのに消費される処理コスト/時間のうちの少なくとも1つであることができる。

【0034】

最適化は、タスク固有のゲインおよび(または)コスト要件を用いることによって行われることができる。

30

【0035】

最適化は、期待される情報フローおよび現在の情報フローに関連する不確実性情報を用いることによって、および(または)確率的手法を用いることによって、行われることができる。

【0036】

用いられる視覚サブタスク情報は、サブタスク動作からの実行時の統計を収集することによって、漸進的に(incrementally)調節されることができる。

【0037】

実行時の統計は、経過時間および(または)結果の精度であってよい。

40

【0038】

実行時の統計を、最適化ステップで使用するすることができる。

【0039】

視覚事象間の従属性は、次のような方法で、サブグラフに格納されることができる。

- 従属性情報は、感覚事象を表現するノード間のリンクとして明示的に格納される、
- 従属性情報は、既に編集された感覚情報の現在の状態と共に用いられる、
- 従属性リンクは、異なるタイプであることができ、これにより、処理順序および情報フローの異なる評価を可能にする、
- 感覚事象を表現する各ノードは、その結果の有効性を追跡し、この有効性を判断する

50

ことができる。

【 0 0 4 0 】

ノード間のリンクは、第 1 のノードの結果が、第 2 のノードの結果に従属することを表現することができ、これにより、第 2 のノードは、第 1 のノードより前に処理されなければならない。

【 0 0 4 1 】

従属性情報は、視覚サブタスクの処理順序を決定するのに寄与することができる。

【 0 0 4 2 】

従属性リンクのタイプは、強制的 (“ mandatory ”)、選択的 (“ optional ”)、および (または) 多数のうちの 1 つ (“ one-of-many ”)、であることができる。

10

【 0 0 4 3 】

従属性リンクの第 1 のタイプについては、厳密な順序が、2 つの接続ノード間に課され、これにより、1 つのノードの処理は、従属ノードの有効な結果を必要とする。

【 0 0 4 4 】

従属性リンクの第 2 のタイプについては、処理ノードは、従属ノードの結果を、それを処理することなく使用することができる。

【 0 0 4 5 】

従属性リンクの第 3 のタイプについては、いくつかの従属ノードのうちの 1 つが有効な結果を持ち次第、従属性を解決することができる。

【 0 0 4 6 】

結果の有効性を、計測の信頼性、取得したデータの品質、最後に計測してからの経過時間、によって判断することができる。

20

【 0 0 4 7 】

視覚サブタスクの処理順序を、動的に決定することができる。

【 0 0 4 8 】

また、視覚サブタスクの処理順序を、視覚分析中に決定することができる。

【 0 0 4 9 】

従属性リンクの第 2 のタイプにおいて、処理ノードは、従属ノードが有効な情報を使用できるならば、該従属ノードの結果を使用することができ、そうでなければ、それを無視することができる。

30

【 0 0 5 0 】

視覚サブタスクの従属性は自動的に解決される。これは、次のステップを含む。

- 先行する視覚事象の結果を表現するノードの従属性リンクを得ること。
- 従属性リンクが接続する各先行ノードについて、
 - 先行ノードのデータが有効かどうかをチェックすること。これが有効なら、次の従属性に続けること。
 - その後続ノードの場合と同じやり方で先行ノードの従属性を解消すること。
 - 先行ノードのデータを取得すること。
 - 先行ノードからのデータを用いて、ノード自体の感覚プロセスを、実行すること。
 - 実行の結果の有効性を評価すること。
 - 解決された従属性にマークを付け、後続ノードに戻る。
 - ノードの結果が無効ならば、視覚サブタスクを再び呼び出す時を、高レベル上で決定すること。

40

【 0 0 5 1 】

ノードの無効な結果は、その先行ノードが既に無効なデータを含んでいるか、もしくは品質 / データの信頼性が所定の基準を下回っている、ことを示すことがある。

【 0 0 5 2 】

従属性の解決は、非同期になされることができる。

【 0 0 5 3 】

ローカルプロセスの呼び出し、視覚サブタスクの呼び出し、および (または) 従属性の

50

解決は、並列になされることができる。

【 0 0 5 4 】

他の側面では、本発明は、コンピューティング・ユニット上で実行されるときに上記の方法を実行する、コンピュータプログラム製品を提供する。

【 0 0 5 5 】

さらに他の側面では、本発明は、データ駆動型でリソースが最適化される視覚動作について、要求に応じて提示される視覚計算を用い、これにより、所定のタスクに関連する視覚の側面に選択的に専念する、ロボットや車両のような可動のプラットフォームに接続される、コンピュータ・ビジョン・システムを提供する。

【 0 0 5 6 】

添付の図面とともに好ましい実施形態の以下の記載を参照すれば、当業者にとって、さらなる利点、目的、および特徴は明らかである。

【図面の簡単な説明】

【 0 0 5 7 】

【図 1】使用される関係記憶を示し、これは、任意の数のリンクパターンを表現することができ、いくつかの例が該図に示されている。

【図 2】ノード A の計測が、ノード B の動作 o p に従属することを示し、該パターンは、図 1 に示されるような記憶に格納される。

【図 3】従属性リンクパターンについての異なるケースが、修正子を用いてどのようにカバーされるかを示し、ここで、a) 従属性が選択的または強制的であること、b) 異なる動作（送受信）が目標ノードについて要求されること、および、c) すべての従属性について満たされる必要がある場合と、多くの従属性のうちの 1 つのみについて満たされる必要がある場合とが区別可能であること、が示されている。

【図 4】システムで使用される対象物構造（上部）を示し、視覚ルーチン（四角形）に対する結合（グレーのライン）が示され、これは、フィードフォワードであり、またフィードバックでもある。図の下部には、従属性構造が示され、四角形は、視覚ルーチンの変数に対する結合を示す（グレーのライン）。

【図 5】対象物の世界位置特性について解決された従属性ツリー（簡略化のため、全てを示していない）を示し、経路が示されている。

【図 6】網膜像位置ノードにおける循環型の従属性の検出を示し、これは、経路のトレースバック（後戻り）となる。循環型の従属性を、網膜像位置ノードおよび空間変調マップノードの間の枝（ブランチ）を切断することにより解決することができる。

【図 7】システムによって既に取得された知識が、どのようにして、効果的な従属性グラフの構造を変化させ、そのサイズを縮小させるかを示す図。

【図 8】対象物のマスクを示し、これは、アルゴリズムによって計算されることができる。

【図 9】左側には、“hasProperty”および“dependsOn”接続の両方が示され、右側には、純粋な構造的定義（上部）および長期記憶の内容（下部）が示される。

【図 10】現在従属性を解決しているノード、データの受信を待っているノード、データの送信を待っているノード、およびデータ送信を終えたノードで、短期記憶内の対象物 1 の色特性についての更新プロセスを示す。

【図 11】どのようにして、到着する情報（網膜像の大きさおよび領域拡張マスク）が従属ツリーに沿って進行し、受信を終えたノードの親ノードでの計算を起動するかを示す図。

【図 12】どのようにして、システムが対象物の距離を要求するかを示し、このプロセス中、空間変調マップについての既存の情報が再使用され、これにより、計算負荷を動的に低減する。

【図 13】主要な構成要素を備える進歩的なシステムを示す図。

【図 14】概念記憶および視覚プロセスの記憶を示す図。

10

20

30

40

50

【図 1 5】概念記憶および視覚プロセスの記憶のサブネットに格納される、内部の視覚記憶に沿った視覚的入力により詳細な概要を示す図、。

【図 1 6】プログラムのフロー図。

【発明を実施するための形態】

【 0 0 5 8 】

認知的視覚(cognitive vision)システムは、技術的においても、また生物学的においても、少なくとも普遍性を意図して、該システムの環境から自身が取得した情報を、注意深く選択しなければならない。

【 0 0 5 9 】

これは、コンピュータおよびメモリのリソースに関する制約を満たすのに必要なことである。したがって、これらのシステムは、その必要性、タスク、および蓄積してきた世界に関する知識に従って、周囲のシーン(scene)の所定の側面に焦点を当てるアルゴリズムを実現する。後述する参考文献(1)で提案されているような、この柔軟性ある制御のアーキテクチャは、システムの処理経路を動的に再配置し、既に取得している知識を使用し、該システムのアクションのコストおよび利点を見積もることができなければならない。これを達成するため、或る合理的な手法では、該システムは、対象物間の関係についての知識のみならず、その周辺についての情報を取得するのに使用することのできる内部ルーチン間の関係についての知識をも必要とする。この知識は、その後、対象物の所定の特性を計測するのに該システムがどのアクションを実行しなければならないかを決定するのに使用されることができる。

【 0 0 6 0 】

もし、たとえば該システムが、対象物は何色であるかを計測するのを欲したとき、該システムは最初に、該対象物がどこに存在し、該対象物の網膜像の大きさ(retinal size)がほぼどのくらいであるかを知る必要がある。

【 0 0 6 1 】

対象物の位置を判定することは、さらなる処理を必要とするかもしれない。該さらなる処理は、局所化モジュール(ローカルモジュール)の従属等であるかもしれない。我々が選択した構成は、この従属性を、該システムが有する世界知識(world knowledge)に従って、関係記憶(リレーショナル・メモリ(relational memory))にモデリングすることを可能にする。この明細書では、異なるルーチン間の従属(依存)性についての知識を効率的に表現する方法、および、システム・コンテキストでこれを使用する方法、について集中して述べる。

【 0 0 6 2 】

コンピュータサイエンスにおいて、従属性の表現に類似の問題が存在する。コンピュータプログラムのデータフローを表現することに関するこれらの問題は、後述する参考文献 Dennis の(2)、(3)の作業にまで遡る。この作業において、またその後の作業において、グラフ構造が、コンピュータプログラムのデータおよび制御フローを分析するのに用いられ、コンパイラによって該プログラムを並列化および最適化する(参考文献(4)、(5))。ここで、該プログラムの従属性のグラフは、オリジナルのプログラムのセマンティックス(意味)を保つのに従わなければならないプログラム中のステートメントおよび述語に対して半順序(partial ordering)を取り入れる(第 3 2 2 頁の 4)。コンピュータ・ビジョンの領域では、データフローのグラフもまた、視覚システムの設計を容易にしてそれらの複雑性を管理しやすくするのに使用される(参考文献(6))。

【 0 0 6 3 】

しかしながら、すべての上記の方法は、固定し予め定義されたアルゴリズムを、グラフ構造にマッピングするものである。この構造は、その固定したアルゴリズムをその後に並列化および最適化するために使用される。

【 0 0 6 4 】

これに対し、我々は、オンデマンド型の視覚システムを実現するための方法を提案し、これは、従属性の内部表現を解析し、対象物の要求される特性を取得するためのプログラ

10

20

30

40

50

ムを動的に作成する。

【 0 0 6 5 】

コンピュータサイエンスの分野における非常に多数の文献が示すように、グラフ構造は、その目的に良く適している。この明細書では、我々は、以下を示す：

- 対象物の特性間の機能的な従属性を、対象物の特性構造および世界知識と共に（これは、短期および長期の両方である）、整合性良くモデリングすることができるグラフ構造を使用する。

- グラフ構造を使用して、システムには、所定の計測のコストを見積もるための手段が備えられる。或る特性を計測するためのグラフのサイズ（大きさ）を、コスト関数として使用することができる。

- ここに提案する解析アルゴリズムを使用して、該システムによって既に取得されている知識を、簡単に効率的なやり方で再使用することができる。これは、計算時間を低減し、システムの動作を速めることにつながる。

- ここに提案する解析アルゴリズムを使用して、視覚システムを設計する複雑性は、直接的な従属性のみをモデリングすることにより、顕著に低減される。

【 0 0 6 6 】

以下では、機能的従属性がモデリングされるやり方と共に、システムのメモリ構造が提示される。また、従属性構造における視覚システムの全体の機能性をカバーするのに必要な修正子(modifier)が、明らかにされる。

【 0 0 6 7 】

さらに、以下では、前述したグラフ構造を採用した解析アルゴリズムを提案する。該構造で動作する際に遭遇しうる特定の状況が、説明される。提示される該解析アルゴリズムを用いて、参考文献(1)で提案されているアーキテクチャに基づく概念証明型のシステム(proof-of-concept system)におけるいくつかの実験が示され、その結果が説明される。

【 0 0 6 8 】

関係記憶 (Relational Memory: リレーショナル・メモリ)
記憶構造 (Memory Structure)

この進歩的なシステムでは、参考文献(7)で提案されている、関係意味記憶(リレーショナル・セマンティック・メモリ(relational semantic memory))を、短期記憶(short-term memory)および長期記憶(long-term memory)において情報を表現するのに用いる。

【 0 0 6 9 】

この関係記憶は、多くの他の意味記憶(セマンティック・メモリ)とは異なり、任意の数のリンクパターンを表現することができる。こうして、図1に示すように、“hasProperty”、“isPartOf”、または“isContainedIn”のような古典的なリンクタイプを定義することができる。

【 0 0 7 0 】

さらに、感覚の表現(sensory representation)が、該システムにその情報をその後フィードバックすることができるように、特性(property)ノードに格納される。感覚表現と共に、或る特性を取得するのに使用される視覚(ビジュアル)ルーチンへの直接リンクが、該特性ノードに格納される。

【 0 0 7 1 】

こうして、結び付けられた視覚ルーチンに対し、情報を伝えるよう指示する(要求する)ことができる。該記憶における対象物(object)は、いくつかの視覚特性から構成される。古典的なリンクパターンの他に、従属性パターンを構築することができる。図2に示される従属性パターンは、“Aの計測が、Bの動作opに従属する”ということを示す。このリンクが与えられると、我々は、要求駆動(demand-driven)のやり方で、Aを計測することができる。すなわち、システムが、ノードAを計測する必要があるならば、該システムは、Aを処理することのできる前に、ノードBのopを実行しなければならないことを知る。

10

20

30

40

50

【 0 0 7 2 】

Bの動作opは、さらなる従属性を持たず、よって直接的に実行されることができる。その後、Aを計測することができる。この構成がより複雑になると、グラフは、より深くなり、より複雑なアルゴリズムが、該グラフを解析するのに必要とされる。この点についての詳細は、以下に述べられる。

【 0 0 7 3 】

リンクの修正子 (Link Modifier)

Bal lanceらが、彼らの論文で、スイッチも制御従属性も、要求により駆動される解釈には必要とされないと述べているけれども(第261頁の8)、我々は、視覚システムの関心のあるケースをカバーするため、従属性リンクのパターンについていくつかの修正子を必要とする。これらの関心のあるケースは、以下の通りである。

【 0 0 7 4 】

- ノードBの動作が、選択的であり、ノードAを計測するのに絶対的に必要とされるものではないが、たとえば計測の結果を向上させるものである場合。たとえば、空間変調マップは、対象物の探索空間を制約するものとなるが、空間全体が該対象物について探索されるべきである場合には、必ずしも必要とはされない(図3a参照)。

【 0 0 7 5 】

- システムが、現在のノードを処理することができる前に、実行されるべき目標ノードについて異なる動作を要求するとき(図3b参照)。

【 0 0 7 6 】

- 或る特性を計測するのに代替の手法が存在することがあり、システムは、いくつかの従属性のうちの1つのみを満たしさえすればよい。対象物の形状を推定するための異なるセグメント化アルゴリズムを考えてみると、これらのアルゴリズムのうちの1つのみが、形状を取得するのに必要とされる(図3c参照)。

【 0 0 7 7 】

我々が実現する一般的なパターンは、“Aは、Bの(operation:動作)に従属し(dependency type:従属性タイプ)、(logical mode:論理モード)、Cの(operation:動作)に従属し(dependency type:従属性タイプ)...”と表現される。この一般的パターンの修正子は、以下のようになる。

【 0 0 7 8 】

- 従属性タイプ(dependency type): ノード間のリンクは、図3aに示すように、強制的(mandatory)または選択的(optional)とすることができる。

【 0 0 7 9 】

- 動作(operation): 図3bに示すように、目標ノードの情報をプッシュ(push)する、もしくはプル(pull)する、送信動作および受信動作をそれぞれ実現する。

【 0 0 8 0 】

- 論理モード(logical mode): ノード“AはB AND (および) Cに従属することができる”、または、ノード“Aは、B OR (または) Cに従属することができる”。論理ORモードを用いることによって、代替の経路をマーク付け(marking)することができ、それ以外は、ノードAは、すべての目標ノードに従属する(図3c参照)。

【 0 0 8 1 】

ノード状態 (Node States)

ここで、各ノードは、ノードデータの有効性をマーク付けする状態を有する。これは、ノード情報が更新される必要があるかどうかを判断するために用いられる。すなわち、このノードに結びつけられた視覚ルーチンを実行する必要があるか否かが判断される。基本的に2つの状態があり、データが、有効(valid)であるか、もしくは無効(invalid)であるか、である。最初に、すべてのノードは、無効データを含む。更新後、すなわち視覚ルーチンから情報を受け取った後、ノードのデータは有効となる。ノード状態の“無効”への遷移は、時間により、もしくは何らかの他の基準により決定されることができる。有効データを備えるノードに遭遇したときに動作の数を動的に減らすのに、ノード状態をどのよ

10

20

30

40

50

うに使用するかについては、後述される。

【 0 0 8 2 】

システムの記憶のレイアウト (System Memory Layout)

異なるリンクタイプ、動作、修正子およびノード状態を説明したので、ここで、該システムで使用される実際のプロトタイプの記憶（メモリ）パターンを示す。

【 0 0 8 3 】

図 4 の上の部分は、対象物構造についてのビュー(view)を示す。該対象物の特性は、様々な視覚ルーチン（左上に示されている）に結びつけられる。図 4 の下の部分には、従属性パターンが示される。この図は、単に、記憶のコンテンツ（内容）についての異なる 2 つのビューを示すにすぎない点に注意されたい。両方の表現は、同じノードを用いて同じ記憶内に共存している。図に見られるように、ノードの直接的な従属性のみを定義しており、全体的なツリー（木）は定義していない。これは、設計プロセスを容易化し、システム構造を管理しやすいものにする。完全な従属性のツリーは、後で説明する解析アルゴリズムを用いて生成される。

【 0 0 8 4 】

従属性解析(Dependency parsing)

上記では、該システムが、世界およびその内部の機能的な従属性の両方についての知識を表現する手法が示された。ここでは、該システムの周辺についての感覚情報の要求駆動による取得を実現するために該知識がどのように使用されるかを示す。図 4 の底部には、直接的な従属性のみが定義されている。対象物の 3 次元位置（world location：世界位置）のような、対象物の特性を更新するためには、そのノードの従属性を解決する(resolve)のが必要とされる。図 5 における、該世界位置について該解決された従属性のグラフは、必要なステップを表している。

【 0 0 8 5 】

再帰解析 (Recursive parsing)

世界位置(図のWorldLocation)を受け取るというこの例において（図 5 のステップを参照）、これは、網膜像の位置(図のRetinalLocation)の計測（受信）（図の 1）および対象物の距離(Distance)の計測（受信）を必要とする。なお、該対象物の 3 次元位置は、奥行き推定(depth estimation)アルゴリズムによって計算されることができる。

【 0 0 8 6 】

しかしながら、たとえば網膜像位置の計測自体は、空間変調マップを送信すること（図の上のSpatialModulationMap）に従属している（図の 2）。この従属は、選択的である。網膜像位置を、該変調の入力を有することなく計測することもできるからである。空間変調マップの送信自体は、空間変調マップの取得（受信）（図の下のSpatialModulationMap）に従属する（図の 3）。変調情報は、それが使用される前に取得される必要があるからである。

【 0 0 8 7 】

従属性グラフを解析することを、再帰的な問題として定式化することができる。よって、後述の擬似コードで示されるように、解析アルゴリズムは、再帰関数として実現される。

【 0 0 8 8 】

この例は、さらに続き、従属性が、さらにもう 1 ステップ追跡される（図 6 を参照）。空間変調マップの計測は、対象物マスクの計測および該対象物の網膜像位置の計測に従属する（図の 4）。これらの 2 つの情報は、正しい形状で正しい位置に、空間変調マップを作成するのに必要である。しかしながら、網膜像位置ノードは、すでに、それよりも前に訪問されている。ここで示されているのは、ループ、すなわち循環型の従属性(circular dependency)であり、これは、システムがそれを扱う手段を持たないと、デッドロック状況を生じさせることとなる。

【 0 0 8 9 】

第 1 の重要な点は、グラフにおいて訪問したノードをマーク付け(印付け)することによ

10

20

30

40

50

って容易に行われることのできる、このような循環型の従属性を検出することであり、該ノードが、そこに入る前に既にマーク付けされているかどうかを確認する。第2の重要な問題は、循環型の従属性を検出したならば何をすべきか、である。ここで、上記に述べた従属性タイプが、重要な役割を果たす。循環型従属性を検出した後、親ノード（空間変調マップ）に戻り（図の4）、該従属性が、強制的(mandatory)なものか選択的(optional)なものかを確認する。従属性が選択的なものならば、それで終わりであり、つまり、この点における該ループを、該アルゴリズムを破壊することなく単に切断する。これは、欠けている該情報が、該アルゴリズムが動作するのに本質的なものではないからである。しかしながら、もし該従属性が強制的なものならば、該システムは、現在のノードの従属性を解決することができない。この図の例では、後者のケースがあてはまる。空間変調マップが、網膜像位置を知ることが必要とするからである。こうして、該システムは、他のステップに戻って（図の3）、従属性グラフの親の動作を実行することができるかどうかを確認する（このケースでは、空間変調マップの送信）。図6に示されるように、これは、このケースには当てはまらない。空間変調マップを送信することは、まずそれを受け取ることに厳密に従属しているためである。再び、従属性経路は、1ステップ戻る必要がある（図の2）。これにより、網膜像位置の受信にまで戻ることとなり、これは、空間変調マップを送信することに選択的に従属している。この時点で、ループに至る完全なブランチ（枝）を切断（カット）することにより、循環型従属性を“解消”することができる。循環型従属性を取り扱う手順を、以下にまとめる：

- 1) 循環型従属性を検出する、
- 2) 従属性ループに至る現在のリンクが選択的なならば、これを切断し、循環型従属性を含む枝全体を取り除く、
- 3) そうでなければ、ルート（根）ノードに既に辿りついたかどうかを確認する。その場合、従属性を解決することはできず、よってエラーが返される。まだルートにたどりついていなければ、従属性経路を1ステップだけ戻り、上記のステップ2)を続ける。

【0090】

既に取得された知識の再使用 (Reusing Already Acquired Knowledge)

機能的従属性を柔軟にモデリングすることへのこのアプローチの最大の利点の一つは、システムが有する知識を再使用することができるという事実である。そのため、ノード状態が、上記のように取り入れられた。該ノード状態は、ノードが更新を必要としているかどうか、すなわち、ノードが、従属性グラフにおいてその親によって必要とされる動作を実行することを必要としているかどうか、あるいは、ノードが、すでに有効なデータを保持しているかどうかを、グラフ解析アルゴリズムに知らせる。ノードが既に有効なデータを有していれば、システムは、該ノードより下の従属性サブツリー（sub-tree）全体を実行する必要が無い。ここで、網膜像位置（該データは、なお有効である）が既に計測されており、対象物の世界位置を更新することが、現在所望されているとする。これにより、図7に示されるような縮小したグラフとすることができる。このグラフを、図5のオリジナルのものと比較すると、結果としてグラフが小さくなっているのがわかる。これは、従属性グラフの構造が、システムの知識によって決定されることを意味する。これは、固定したグラフについてのみ作用する参考文献（4）、（8）、（6）のような、以前に提案された方法に対する主要な相違点である。

【0091】

最終的に、グラフは、システムの知識を取り入れることによって小さくなり、より効率的で負荷の小さいシステムにすることができる。

【0092】

代替の経路 (Alternative Pathway)

認知視覚システムのロバスト性は、冗長性によってしばしば向上させることができるので、或る対象物の特性を計測する代替の方法が存在することが、該システムにおいて望ましい。これは、特性を判断するための様々なアルゴリズムが、データ、結果を計算する方法、該システムの持つ速度、精度および脆弱性等について行う想定に、違いがあるかもし

れないからである。したがって、そのような代替の経路を扱うような方法が加えられる必要がある。図 8 に示される例では、3 つの異なるセグメント化アルゴリズムがあり、顕著性 (saliency) マップを用いたシンプルなサイズ (大きさ) 推定 (詳細は文献 (9) を参照)、領域拡張法 (Region-Growing method, 文献 (10) を参照)、およびレベルセット法 (Level-Set method, 文献 (11) を参照) である。代替経路をモデリングするため、論理 OR モードが、上記のように取り入れられた。図 8 に見られるように、“対象物マスク (Object Mask)” ノードがマーク付けされ、これは、“OR ノード”を示す。OR ノードは、グラフパーサ (解析部) によって、“これらの従属性のうちの 1 つのみが必要とされる”と解釈される。対象物マスクを計算するため、これらのルーチンのうちの 1 つが、開始される必要がある。しかしながら、異なる複数のアルゴリズムが見つかることとなり、これらのアルゴリズムは、速度、初期的な要件、および精度が異なるものと見ることができる。網膜像の大きさ (Retinal Size) の推定は、非常に速く、初期値として対象物位置のみを必要とし、それほど精度が高くはない。領域拡張法は速く (しかし、網膜像の大きさの推定よりは遅い)、初期値として対象物位置のみを必要とし、少なくとも同種の構造化された対象物についての精度は良い。他方、レベルセット法は、他の 2 つのアルゴリズムに比べて比較的遅く、開始するのに初期セグメント化を必要とするが、構造化された対象物についても非常に精度が高い。上記述べた特性のうちの 1 つの結論は、対象物マスクを初期的に推定するにはレベルセット法を用いることはできない、ということである。なぜならば、該レベルセット法は、実行するのに初期マスクを必要とするからである。さらに、該システムは、必要に応じた精度を持ち、なるべく速いアルゴリズムを選択することができる。したがって、必要とされるのは、現在のシステムの状態 (たとえば、必要とされる精度および使用可能な時間) および該システムの知識 (たとえば、初期の対象物マスク) に依存した決定である。最も簡単な実現形態では、パーサ (解析部) が、従属性のうちの 1 つが解決されるまで、従属性を解決しようとの試みを継続的に行うことである (ノードは、その初期状態が満たされなければ、その実行を拒否できる)。

【0093】

いずれの従属性も解決されなければ、循環型従属性において上記述べたトレースバック (後戻り) を実行することができる。従属性の解決可能性とは別に、解析アルゴリズムの拡張バージョンは、これらの異なる経路のコストおよび精度を考慮することができる。

【0094】

グラフ解析アルゴリズムの擬似コードは、問題が再帰的であるので、再帰的な本質を持っている。該アルゴリズムは、動的に、要求された特性から開始する従属性グラフを生成する。また、循環型従属性の検出とその取り扱いを考慮する必要がある。更新手順は、以下のとおりである。

【0095】

< 手順 (UpdateNodeValue) >

(a) ノードの実行する能力をチェックする。

(1) 有効なデータについて、現在のノードをチェックする。該ノードが既に有効なデータを持っていれば、動作をスキップして、「成功」を返す。

(2) 既にセットされた訪問済みフラグによって示される、循環型従属性をチェックする。循環型従属性を検出したならば、対応するエラーを、該ノードの親に渡す。

(3) 該現在のノードについて、訪問済みフラグをセットする。

(b) 従属性の更新

(1) 現在のノードについての全ての従属性についてのリストを取得する。

(2) 各従属性 (子ノード) について、以下を行う：

(2.1) 子ノードに対し “UpdateNodeValue” を呼び出す、

(2.2) 循環型従属性のエラーについての該呼び出しの戻りコードをチェックする。そのようなエラーを受け取り、かつ該子ノードについて強制的従属性を持つ場合には、該エラーを、さらにその親に伝える。選択的従属性に対するエラーについては、該リストの次の従属性を処理することを続ける。

(2.3) 論理ORノードであるならば、少なくとも1つの従属性が満たされるので、該ループを去って(c)を続ける。

(c) 現在のノード動作の実行

(1) 親によって要求される、現在のノードの送信または受信動作を実行し、選択的に、センサデータを局所的に格納する。

(2) データの有効性フラグをセットする。

(3) 訪問済みフラグを取り除く。

【0096】

実験 (Experiments)

上記のアルゴリズムおよび図4に示される構造的な定義を用いることによって、概念証明型(proof-of-concept)のシステムが実現される。該システムの記憶内容は、図9に見ることができる。図9の“プロトタイプ(prototypes)”セクションにおいて、対象物の構造のみが定義される。すなわち、どの特性が対象物を構成し、該特性がどのように関連しているかが定義される。概念証明型のシステムにおいては、“hasProperty”リンク(黒)のみが用いられる。“Sensory Interface(感覚インターフェース)”セクションにおいて、該対象物構造は引き継がれ、さらに、従属性定義および視覚ルーチンに対する結合が加えられる。従属性構造は、図4に示すような直接的従属性の定義から生じたものである。

【0097】

長期記憶(Long Term Memory)は、該対象物構造を、“Sensory Interface”から引き継ぐ。ここで、“hasProperty”リンクのみが、読みやすさを維持するため示されている。図からわかるように、すべての特性が、対象物毎に具象化(インスタンス化)されるわけではない(カラーおよびサイズのみが、ここでは選択される)。しかしながら、網膜像位置または距離のような他のノードを、必要に応じて具象化することができる。

【0098】

長期記憶(long-term memory)における特性ノードは、感覚表現(sensory representations)を格納し、これらは、これらにリンクされる対象物について安定的なものである。他の変化しやすい対象物情報は、ここでは格納されないが、具体的なシーンにおいて計測され、短期記憶(short-term memory)に格納される。短期記憶の内容は、その構造を、長期記憶から引き継ぐ。要約すれば、対象物構造は、プロトタイプの定義から始まり、様々な記憶インスタンスを介して、短期記憶へと、継承によって伝達される。

【0099】

この構造的な情報の他に、表現は、センサへの結合および互いに従属するノードをリンクするグラフによって、質的に向上される。該結合および従属性グラフは、“Sensory Interface”層において取り入れられる。この点まで、何の実値(real value)も、特性ノードには満たされない。これは、1または複数の対象物にリンクされることのできる長期の安定したセンサデータについて、長期記憶内で生じる。

【0100】

このメモリ構造に基づいて、ノードについての更新プロセスがどのようなものが示される。対象物1(obj_1)の色特性についての更新プロセスが、図10に示されている。

【0101】

従属性解決プロセスおよび後続の情報伝達プロセスが、非同期に実現される点に注意されたい。特性を更新する第1のステップは、それを具象化(インスタンス化)することである(図10a参照)。こうすることにより、プロトタイプの定義にまで特性は継承される。それらの特性のうちの1つは、ノードの従属性である。対象物(object)の色(color)を計測するため、空間変調マップ(spatial modulation map)が必要とされ、これは、さらに、対象物の網膜像位置(retinal location)を必要とし、さらに、これは、顕著性(saliency)についての重み係数(weight factor)を必要とする。この伝達プロセスの結果が、図10bに示されている。

【0102】

10

20

30

40

50

顕著性の重みは、さらなる従属性を持たないので、それらは、速やかに送信されることができる。そうした後で、網膜位置ノードのすべての従属性が満たされ、これは、データのためにその視覚ルーチンを要求する(図10c参照)。網膜位置ノードの視覚ルーチンを起動した後、該プロセスは、空間変調マップにおいて継続する。前に定義したように(図4参照)、該変調マップは、処理されるべき対象物マスク(object mask)を必要とする。対象物マスクはORノードであり、これは、代替の3つの計測プロセスが存在するからである(網膜像の大きさの推定、領域拡張法、およびレベルセット法)。これらの視覚ルーチンの1つのみが、実行される必要がある。

【0103】

図10dに示されるように、対象物マスク(object mask)ノードは、最初に、レベルセット計測(level set)を起動しようと試みた。しかしながら、前述したように、このアルゴリズムは、実行するのに初期マスクを必要とする。そのようなマスクを持つことの代替は、領域拡張(region growing)法を用いることである。領域拡張ノードは、最後の葉ノードであるので、従属性は、ノード動作を実行しながら従属性経路を後戻りする(トレースバックすることによって解決される。これが、図10e~gに見ることができる。最後に、すべてのマーク付けされたノードが、データを配信するようそれらの視覚ルーチンを起動した。空間変調マップは、データがそれらを送信するのを待つ。

【0104】

視覚ルーチンからデータが到着したとき、それらは、従属ツリーに沿って上方へと伝達される。図11において、網膜像位置および領域拡張アルゴリズムのマスクは、(ほぼ)同時に到着する。領域拡張マスクについての情報は、その後、上方に向けて、該処理を終了することのできる対象物マスクまで進む。有効な対象物マスクおよび対象物の網膜像位置で、空間変調マップを計算して、その後に送信することができる。空間変調マップを送信した後、色(color)ノードの視覚ルーチンが実行され、最終的に、色(カラー)を返す(図11の底部参照)。

【0105】

何らかの時間が経過した後、いくつかのノードのデータが再び無効になったと想定する。そのようなケースが、図12の一番上に示されており、網膜像位置、領域拡張、および対象物マスクのノードのデータが無効化されている。さらに、システムは、対象物1(obj_1)の距離(distance)を知る必要があると想定する。

【0106】

距離を受け取ることは、空間変調マップを送信することを必要とする(図4参照)。典型的には、このマップを送信することは、該マップの抽出を必要とするが、このケースでは、空間変調マップのデータはなお有効である。こうして、更新する必要がないので、該プロセスは、該情報を送出し続けることができる。この手順を、空間変調マップを送信することが対象物マスクおよび網膜像位置の枝(ブランチ)全体を起動したものとなっている図10の手順と比較すると、計算的な手間が劇的に低減されている。最後に、図12の底部に示されるように、該距離は、システムによって要求されて受け取られる。

【0107】

暫定結果 (Interim Result)

上記に提示されたシステムは、機能的な従属性についての知識および世界についての知識の両方を整合性あるやり方で表現するのに、グラフ構造を用いている。このシステムにおける関係意味記憶(リレーショナル・セマンティック・メモリ)が、ノード間の任意の数のリンクパターンを表現することができるという事実は、さらに、視覚ルーチンを、そのノードに結合することができる。従属性リンクのモデリングが記述され、該リンクパターンについてのいくつかの修正子を取り入れられ、これにより、視覚システムについての重要なケースをカバーすることが可能となる。選択的および強制的な情報、情報フローの方向をモデリングする異なるノード動作、および代替経路のケースが、説明される。従属性リンク構造に基づく解析アルゴリズムが設計され、これは、所定の環境下における循環型従属性を検出して“解決”することができる。これとは別に、該解析アルゴリズムはま

10

20

30

40

50

た、以前に取得した感覚情報を効率的に再使用して、該システムの全機能を維持しつつ、計算負荷を低減することができる。この実験は、このフレームワークにより、オンデマンドで（要求に応じて）データを取得し、その処理チェーン（鎖）を柔軟に適応させることのできるシステムを構築するのを可能にする、ということを示している。

【0108】

また、該システムが所定のアクションのコストを推定することが可能である。従属性ノードの数は、コスト関数として用いられることができる。しかしながら、或るアクションが要する時間を計測することを考慮し、これを、コスト関数として用いることもできる。

【0109】

この情報を用い、該システムは、時間的な制約が適用されるとき、どのアクションを取ることができるかを学習する。同じように、該システムは、さらに、経路がどれほど正確で信頼性あるかを学習し、良好な精度の情報が必ずしも必要でない場合には、速いけれども粗い(coarse)関数を用いることができる。これをさらに推し進めて、該システムは、自身で従属性を見つけることを試みて、“最適”な処理待ち行列を学習するようにしてもよい。

【0110】

従属性構造を推定することとは別に、参考文献(4)のようなコンピュータサイエンスからのまとまった知見を容易に採用して、処理経路を最適化および並列化するようその後動作することもできる。そこで見つけれられるアルゴリズムは、基礎となる構造が似ているため、容易に適用されることができる。

【0111】

本発明のコア（中核）

本発明は、さらに、視覚的入力を選択的に分析する視覚ルーチンの、アクティブで半順序(semi-sequential)な補充(recruitment)および調整(modulation)によって、視覚サブタスクの解決を含む内部タスクによって駆動されるコンピュータ・ビジョン・システムを提供する。この補充および調整は、実行時に動的に起こり、視覚的なアイテム（項目）およびプロセスについての長期知識だけでなく、視覚分析の直前のステップ中に編集された視覚シーン、そのアイテムおよびその特性について現在利用可能な短期情報にも基づいている。長期記憶および短期記憶は、前述したように構成されることができる。

【0112】

その主要な構成要素を備えたシステムが、図13に示されている。この図の底部において、視覚的入力(visual input)が、非タスク固有(non-task-specific)の一連の事前処理ステージにおいて取得され、分析/事前処理される。視覚的入力は、1つ以上の画像（たとえば、ステレオまたはマルチカメラ機構のように）を含むことができ、また、グレースケール、カラー、もしくは他の視覚的領域（たとえば、赤外線のような）を用いたものであることができる。事前処理ステージ（図13のA）は、シーンおよびその対象物についての知識を必要としない特性（すなわちそれらは特定のではなく一般的である）を用いる、たとえば正規化、コントラスト強調、エッジおよび特徴抽出および同様の処理を含むことができる。事前処理動作のタイプは、この発明には関係がないので、この点についてさらなる詳細は述べない。

【0113】

事前処理ステージからの出力は、図13のBに示される“視覚ルーチン(visual routine)”によって使用される。これらは、対象物に固有(object-specific)の、またはコンテキストに固有(context-specific)の視覚プロセスであり、視覚シーンについて何が探索されるか、および（または）何が既知であるか（すなわち、現在の感覚コンテキストについて短期記憶に何が記憶されているか）に依存している。対象物またはコンテキストの固有（特定）性は、トップダウンの接続を介して（図13の右上の点線矢印）視覚記憶（メモリ）から届けられ、それは、特定の対象物またはコンテキストの記憶から抽出されたパラメータを用いた、視覚ルーチンの選択、操作(steering)、および調整(modulation)を含む

10

20

30

40

50

。

【 0 1 1 4 】

視覚ルーチンの例として、対象物の存在を示す特定の特徴を見つけることによる対象物候補の検出、対象物に固有の追跡、位置に固有のセグメント化等がある。

【 0 1 1 5 】

視覚ルーチンの固有の特性は、該視覚ルーチンが、それらに割り当てられた基本の視覚サブタスクの実現のために固有に起動されることができるということであり、こうして、通常の動作において、視覚ルーチンの一部のみ（実行時で動的に変化するけれども）が各時間ステップで起動される。

【 0 1 1 6 】

操作可能な(steerable)視覚ルーチンは、汎用の視覚システムにおいて計算資源をインテリジェントに管理するための方法を提供する制御プロセスおよび視覚記憶と共に、知識ベースの選択的視覚分析の核となる。

【 0 1 1 7 】

視覚ルーチンは、図 1 3 の C に示される記憶ステージから操作され、その結果を記憶ステージに送る。ここで、様々な視覚ルーチンからの感覚情報は、視覚シーンについての既存の情報に関して編集され、出力される。記憶ステージは、以下のような 3 つの主要な構成要素部分を持つ。

- 1) 概念記憶(conceptual memory) (長期および短期の両方) 、
- 2) 視覚プロセスの記憶 (メモリ) 、
- 3) 視覚ルーチン、概念記憶、および視覚プロセスの記憶の間を調停する制御モジュール。

【 0 1 1 8 】

概念記憶は、感覚入力(sensory input)に現れる特徴パターンであって、たとえば視覚テンプレート、物体、視覚的な周囲の状況 (“ 部屋 ” のような) 、または特徴的なシーン (室内、室外等) に関連づけることのできる特徴パターンを表現するのに用いられる。これは、特定の感覚概念を構成するものの情報を記憶するので、概念的と呼ばれる。

【 0 1 1 9 】

概念記憶に格納される概念 (コンセプト) は、或る存続期間 (ライフタイム) の間は有効であり、一般的に有効と考えられ時間と共に安定する概念 (“ 長期 ”) から、感覚対象物についての一時的な計測結果または仮定を表現する概念に至るまで、連続して広がっており、また、短期の時間スケール (短期記憶) に対しては変化することが期待されるものである。

【 0 1 2 0 】

長期の概念記憶は、たとえば、特定の空間的な構成で配置されるべき構成要素として車輪を有するというような、視覚的に車を定義するものの情報を含む。これに対し、短期記憶は、たとえば、感覚入力から検出され、3次元世界の座標および感覚空間で動く、特定の時間ステップで特定の車の情報を表現するものである。この場合、感覚要素 “ 車 ” の短期記憶は、その特性の多くを継承するが、対象物関連の感覚情報を取得して更新する際に係わる現在の計測プロセス (視覚ルーチンに結合される) に関連する情報だけでなく、実際に計測された車の感覚特徴に関連する情報を付加的に持つという意味で、長期概念の “ 車 ” を特化したものとなる。

【 0 1 2 1 】

視覚プロセスの記憶は、或る感覚的なものを計測する方法、具体的には、概念記憶からの視覚的なアイテム (項目) の特性を計測する方法の情報を表現するのに用いられる。

【 0 1 2 2 】

車の例に戻ると、視覚プロセスの記憶は、適切な配置で車輪が存在するかどうかを見出す方法 (これは、そのシーンにおいて車が存在することをシステムが推論することを可能にする感覚パターンの存在を確かめるものである) を定義する。

【 0 1 2 3 】

低い感覚レベルにおいて、視覚プロセスの記憶は、視覚ルーチンへの該記憶からのリンクを提供し、明示的に、視覚ルーチンにアクセスしてそれらの結果を抽出するのに必要なすべての関連情報を含む。こうして、視覚プロセスの記憶は、システム自体の感覚装置および該装置上で視覚ルーチンのレベルで取ることのできる可能なアクションの表現を備える。

【0124】

概念記憶および視覚プロセスの記憶の両方は、前述したように、階層的なグラフの形態で表現され、概念のハブ（中心）として動作するノードおよびノード間の関係を示すいくつかの異なるタイプの直接リンクを含む（図14のA参照）。

【0125】

概念記憶の場合、リンクは、たとえば、或る視覚特性すなわちサブパーツ（たとえば、車の車輪）の存在が、より包括的なパターン概念（車）の良好な標示である、ということを示す。

【0126】

視覚プロセスの記憶の場合、包括的な概念（車）に結びつけられた異なるタイプのリンクは、システムが車の存在を推論できる前に車輪が検出されなければならないことを示し、これは、たとえば、車のパーツの検出の視覚ルーチンを補充する（recruit、使用する）ことによって生じる。記憶の両方のタイプが、異なるリンクを用いているが、同じノードを用いた単一のグラフィカルな構造内で混ぜ合わせられること、すなわち、それらは、図14のBに示すようにノードを介して通信する2つの分離したサブネットとして視覚化されることに注意されたい。

【0127】

ノードは、さらに、視覚プロセスおよび視覚ルーチンを駆動するのに必要なパラメータのみならず、現在の感覚計測に関連する内部パラメータ（たとえば、対象物の位置、色等）を、信頼性(confidence)と共に含む。

【0128】

図15は、概念記憶および視覚プロセスの記憶のサブネットに格納された内部視覚記憶に従う視覚的入力により詳細な概要を示す。

【0129】

概念記憶は、階層グラフにおいて、たとえばライト、車輪、ウィンドウといった視覚的な外観について、車がどのようなパーツを持つべきかを指定することによって、車の“概念”を含む。視覚プロセスの記憶は、たとえば特化された車輪検出モジュールのような、対応する視覚ルーチンにアクセスすることによって、これらのパーツの存在を計測する方法をコード化する。

【0130】

さらに、概念化された車は、たとえば、所定の車のパーツの発見が期待される相対的位置、もしくはどの特徴が該パーツの検出を容易にするか（たとえば、リアライトについての赤色）、というような情報を含むことができる。この情報は、その後、視覚ルーチンに選択的に渡され、該視覚ルーチンは、自身の機能を制約したり向上させるように、それを使用することができる。たとえば、車の前輪がすでに検出され、よって或る領域で後輪の発見が期待されるならば、該後輪を検出するための視覚ルーチンを、この領域においてのみ探索するよう拘束することができ、これにより、探索の負担をより軽くしつつ、結果をよりロバストにすることができる。

【0131】

システムの長期記憶および短期記憶の両方とも、実行時に動的に更新される。該更新は、グラフ構造の作成、修正および破棄だけでなく、グラフのノードにおいて格納されるパラメータ値を含む。新しいグラフ構造は、たとえば新しい感覚アイテムが短期記憶に含まれるときに生じ（たとえば、静止した背景に対する移動体のような、それが非常に顕著であるがためにアイテムのポップアウト（飛び出してくること）を視覚ルーチンが示すならば）、もしくは、既に検出された感覚アイテムが、より大きい概念の混合物を形成するよ

10

20

30

40

50

う結合されるときに生じる（たとえば、2つの車輪が検出されて、それが、車の感覚検出を表す新しい概念ノードに結合されるとき）。

【0132】

グラフ構造を、それらの感覚的な有効性がもはや与えられないとき、たとえば、アイテムが、そのシーンから消滅して再び現れるのが期待されないとき、破棄することができる。

【0133】

制御モジュールは、概念記憶、視覚プロセスの記憶、および視覚ルーチンの間の情報フローを制御する。それは、更新ないし計測されるべきパラメータを備えたノードのセット（組）を連続的に識別する。

10

【0134】

視覚プロセスの記憶により、制御モジュールは、パラメータを、それらを計測するのに必要なプロセスにリンクすることができる。したがって、パラメータを計測する目的は、或る対象物がシーンに実際に存在しているか否か、あるいはその位置、色等は何であるか、を見いだすというような、視覚タスクと等価である。要約すれば、制御モジュールは、現在利用可能な概念記憶と共に、プロセスの記憶を使用し、必要なパラメータを計測する方法を決定する。

【0135】

視覚ルーチンの選択的なアクセスは、すでに収集された、長期記憶、より重要には短期の感覚記憶に含まれる概念情報に従って、処理リソース（資源）の効率的な利用を可能にする。処理リソース（計算コスト、エネルギー消費、動作時間、メモリ資源）は、複雑な視覚シーンについて、主要な問題となる。なぜなら、複雑なシーンについては、調べる可能性のある特徴の空間が指数関数的に増大するためである。したがって、単に信号駆動のやり方で、すべての必要な特徴分析ステージを見越すことは不可能である。記憶に基づく選択方式は、資源問題を緩和し、仮説駆動型(hypothesis-driven)の案内される探索プロセスとして、ビジョン（視覚）を前提のものとする。グラフに基づく記憶構造によって提供される階層的な表現は、処理リソースを低減することができ、結果として、コアース・トゥ・ファイン（粗から微細へ）の探索となる。さらに、所与のタスクに必要なこれらの視覚ルーチンおよびパラメータの形態のみに対する制約は、計測結果のロバスト性および信頼性を増大させる。

20

30

【0136】

制御モジュールは、様々なヒント（手がかり）を用い、視覚タスクに必要な視覚ルーチンの選択、操作(steering)および調整(modulation)を行う。制御モジュールは、視覚プロセスの記憶に関連づけられるリンクに格納された情報を用い、特定の感覚事象を実行するときに、期待される情報のゲインおよびコストをチェックすると共に、感覚事象（図14のBの底部の、点線矢印）と視覚ルーチンに対するそれらのリンク（図15の底部を参照）との間の従属性についてチェックする。

【0137】

本発明の特定の実施形態において、従属性リンクは、異なる修飾子(qualifier)/リンクタイプと共に用いられ、これが、処理順序および情報フローの異なる結果となる。使用されるタイプは、強制的(“mandatory”)、選択的(“optional”)、多数のうちの1つ(“one-of-many”)、であった。

40

【0138】

第1のケースでは、厳密な順番が、2つの接続されたノード間で課され、これにより、1つのノードの処理は、最初に、従属ノードの有効な結果を必要とし、こうして、従属ノードが最初に処理されるという必要性を形成する。

【0139】

第2のケースでは、処理ノードは、従属ノードが有効な情報を使用できる（有効な情報へのアクセスを持つ）のであれば、該従属ノードの結果を使用することができ、そうでなければ無視することができる。

50

【 0 1 4 0 】

第3のケースでは、従属性は、いくつかの従属ノードのうちの1つが有効な結果を達成し次第、解決される。

【 0 1 4 1 】

従属情報は、既に編集された感覚情報の現在の状態と共に使用され、視覚サブタスクの処理順序を決定する。さらに、期待されるゲインおよびコストに関する情報を考慮して、制御モジュールが、たとえばより低い処理コストで近似した結果を得るのか、それともより高い処理コストで正確な結果を得るのかを決定することができる。不確実性情報が利用可能であれば（プロセス記憶および感覚計測に結びつけられる）、グラフィカルな構造に対して作用する確率的手法をここで含めることができる。感覚事象を表現する各ノードは、その結果の有効性を追跡する。この有効性を、たとえば、計測の信頼性、その取得したデータの品質、あるいは最後の計測からの経過時間によって判断することができる。

10

【 0 1 4 2 】

制御モジュールは、視覚サブタスクの従属性が自動的に解決されるのを可能にする。これは、図16にプログラムフロー図として示され、結果の有効性と共に、視覚プロセスの記憶からの情報、期待されるゲインとコストに従って、グラフをインタラクティブに進行するステップを含む。概念記憶、特に短期感覚記憶は、何が既に検査されたかを格納するのに使用されると共に、コンテキスト（すなわち概念記憶）に依存する視覚ルーチンのためのパラメータとして作用するのに使用される。或る点でのインタラクティブなグラフの進行は、視覚ルーチンにリンクされたノードに到着し（図15の底部の右側に示される）、ここで、特定の視覚分析を起動する。視覚ルーチンが結果を配信するとき、その有効性は、結びつけられた記憶ノードによってチェックされ、さらに、それを呼び出した先行ノードに通信される。このようにして、必要な感覚プロセスは起動され、視覚タスクの従属性は解決される。

20

【 0 1 4 3 】

本発明を、以下のように要約することができる。

【 0 1 4 4 】

I. その情報処理フロー、その処理リソース、およびそのパラメータを、知覚可能な世界、その対象物とその特性、現在の視覚タスク、感覚的コンテキスト、および知覚装置自体に関する明示的知識を用いて適応させるコンピュータ・ビジョンのための方法であって

30

a. 視覚的に知覚可能な対象物のような感覚事象の記憶を、機能的な関係を表す直接リンクによって他のノードに接続される、計測可能な感覚パラメータを備えたノードエンティティとして表現するステップと、

b. 現在の視覚的な感覚コンテキストの記憶を、感覚事象の接続されたグラフで表現するステップと、

c. 空間サブグラフおよび該サブグラフのノードとリンクに結びつけられたローカルプロセスにおいて、上記Iaの感覚パラメータにアクセスして計測する方法に関する視覚サブタスク情報の記憶を表現するステップと、

d. ステップIa～Icにおいて取り入れられた記憶からの情報を用いることによって、実行時に処理リソースを動的に割り振って情報フローを編成するステップと、

40

e. 上記のリソース割り振りおよび情報フローを、ステップIa～cからのグラフ構造に従って最適化するステップと、

f. グラフの対応するノードおよびリンクのエンティティにおいて、感覚計測から取得した推定値を記憶し、これにより、IaおよびIbからの情報を更新するステップと、

g. たとえばさらなる視覚対象物についての情報、および一般的にはIa、Ibからの一般的な感覚コンテキストにおける追加および修正についての情報を表す新しいグラフ構造を漸進的に(incrementally)適応して作成するステップと、

h. ステップIdに戻り、知覚された世界および自身の感覚プロセスについての知識の表現を改良するために、ステップIa～Icから、自身の処理リソースを柔軟に適応すると

50

共に、記憶を改善（リファイン）するシステムで、ステップI d ~ I hを繰り返すステップと、を含む。

【0145】

II. 上記のIで記述された方法であって、ステップI eの最適化は、I cからの視覚サブタスク情報に加えて、たとえば、時間対精度の調停、リソースの衝突（同じ感覚デバイスを要する異なる計測）、順番および機能的な従属性（どの視覚サブタスクが、他より先行しなければならないか、どれが他からの情報を用いるか）、および、以前の動作中における経験から編集されたサブタスク情報（期待される精度、情報のゲイン、結果を得るのに消費される処理コスト/時間）のような、タスク固有のゲインおよびコストの要件を考慮することによって行われる。

10

【0146】

III. 上記のIまたはIIに記載された方法であって、最適化は、期待される情報フローおよび現在の情報フローに関連する不確実性情報を考慮することによって、および、最新の確率的手法を用いることによって行われる。

【0147】

IV. 上記のI ~ IIIのうちのいずれかに記載の方法であって、ステップI eで使用される視覚サブタスク情報は、サブタスク動作からの実行時（ランタイム）の統計を収集することによって、とりわけ経過時間および結果の精度を収集することによって、漸進的に調節され、この情報を、上記の最適化ステップI e、IIおよびIIIで利用できるようにする。

20

【0148】

V. 上記のI ~ IVのうちのいずれかに記載の方法であって、視覚事象間の従属性は、次ような方法で、ステップI cのサブグラフに格納される。

a. 従属性情報は、感覚事象を表現するノード間のリンクとして明示的に格納される（たとえば、ノードAの結果は、ノードBの結果に従属し、よって、ノードBは、ノードAよりも前に処理されなければならない）。

b. 従属性情報は、既に編集された感覚情報の現在の状態と共に用いられ、視覚分析中に視覚サブタスクの処理順序を動的に決定する。

c. 従属性リンクは、異なるタイプであることができ、これにより、処理順序および情報フローの異なる評価を可能にする。使用されるタイプは、たとえば、強制的（“mandatory”）、選択的（“optional”）、および多数のうちの1つ（“one-of-many”）、である。第1のケースでは、厳密な順序が、2つの接続ノード間に課され、これにより、1つのノードの処理は、従属ノードの有効な結果を必要とする。第2のケースでは、処理ノードは、従属ノードが有効な情報を使用できるならば、該従属ノードの結果を使用することができ、そうでなければ、それを無視することができる。最後のケースでは、従属性は、いくつかの従属ノードのうちの1つが有効な結果を達成し次第、解決される。

30

d. 感覚事象を表現する各ノードは、その結果の有効性を追跡する。この有効性は、たとえば、計測の信頼性、取得したデータの品質、最後に計測してからの経過時間、によって判断されることができる。

【0149】

40

VI. 上記のI ~ Vのいずれかに記載された方法であって、視覚サブタスクの従属性は自動的に解決される。これは、次のステップを含む。

a. 先行する視覚事象の結果を表現するノードの従属性リンクを得ること。

b. 従属性リンクが接続する各先行ノードについて、

i. 先行ノードのデータが有効かどうかを、ステップIV dに従ってチェックすること。これが有効なら、ステップV bの次の従属性へ続ける。

ii. ステップV aで開始し、その後続ノードの場合と同じやり方で先行ノードの従属性を解決すること。

c. 先行ノードのデータを取得すること。

d. ノード自身の感覚プロセスを、先行ノードからのデータを用いて実行すること。

50

- e . 実行の結果の有効性を評価すること。
- f . 解決された従属性にマークを付け、後続ノードに戻る。
- g . ノードの結果が無効ならば（これは、その先行ノードが既に無効なデータを含むか、もしくは、データの品質が所定の基準を下回ったかを示す）、視覚サブタスクを再び呼び出す時を、高レベル上で決定すること。

【 0 1 5 0 】

V I I . 上記の I ~ V I のうちのいずれかに記載の方法であって、従属性の解決は、非同期になされる。

【 0 1 5 1 】

V I I I . 上記の I ~ V I I のうちのいずれかに記載の方法であって、ローカルプロセスの呼び出し、サブタスクの呼び出し、および従属性の解決は、並列になされる。

10

【 0 1 5 2 】

I X . データ駆動型でリソースが最適化される視覚動作について、要求に応じて提示される視覚計算を用い、これにより、所定のタスクに関連する視覚の側面に選択的に専念する、ロボットや車両のような可動のプラットフォームに接続される、上記の I ~ V I I I のうちのいずれかに記載の方法。

【 0 1 5 3 】

上記は、本発明の実施形態に関連するものであり、多くの変更および修正を、特許請求の範囲で記述した本発明の範囲から逸脱することなく行うことができる。

【 0 1 5 4 】

20

参考文献

- [1] Julian Eggert, Sven Rebhan, and Edgar Köörner. First steps towards an intentional vision system. In Proceedings of the 5th International Conference on Computer Vision Systems (ICVS), 2007.
- [2] Jack B. Dennis. First version of a data flow procedure language. In Proceedings of the Colloque sur la Programmation, volume 19 of Lecture Notes in Computer Science, pages 362-376, London, UK, 1974. Springer-Verlag.
- [3] Jack B. Dennis. Data flow supercomputers. Computer, 13(11):48-56, November 1980.
- [4] Jeanne Ferrante, Karl J. Ottenstein, and Joe D. Warren. The program dependence graph and its use in optimization. ACM Transactions on Programming Language and Systems, 9(3):319-349, July 1987.
- [5] Robert Cartwright and Matthias Felleisen. The semantics of program dependence. In Proceedings of the ACM SIGPLAN 89 Conference on Programming Language Design and Implementation, pages 13-27, 1989.
- [6] Per Andersson. Modelling and implementation of a vision system for embedded systems, 2003.
- [7] Florian Rohrbain, Julian Eggert, and Edgar Köörner. Prototypical relations for cortex-inspired semantic representations. In Proceedings of the 8th International Conference on Cognitive Modeling (ICCM), pages 307-312. Psychology Press, Taylor & Francis Group, 2007.
- [8] Robert A. Ballance, Arthur B. Maccabe, and Karl J. Ottenstein. The program dependence web: A representation supporting control-, data-, and demand-driven interpretation of imperative languages. In Proceedings of the ACM SIGPLAN 90 Conference on Programming Language Design and Implementation, volume 25, pages 257-271, New York, NY, USA, 1990. ACM.
- [9] Sven Rebhan, Florian Rohrbain, Julian Eggert, and Edgar Köörner. Attention modulation using short- and long-term knowledge. In A. Gasteratos, M. Vincze, and J.K. Tsotsos, editors, Proceeding of the 6th International Conference on Computer Vision Systems (ICVS), LNCS 5008, pages 151-160. Springer Verlag, 2008.

30

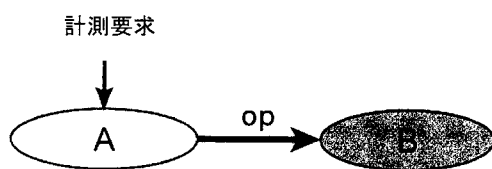
40

50

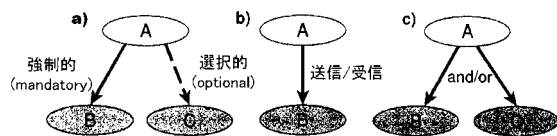
- [10] Milan Sonka, Vaclav Hlavac, and Roger Boyle. Image Processing, Analysis, and Machine Vision. Thomson-Engineering, 2 edition, 1998.
- [11] Daniel Weiler and Julian Eggert. Multi-dimensional histogram-based image segmentation. In Proceedings of the 14th International Conference on Neural Information Processing (ICONIP), pages 963-972, 2007.
- [12] Clouard R., Elmoataz A., Porquet C. & Revenu M.: Borg: A Knowledge-Based System for Automatic Generation of Image Processing Programs. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1999. Vol. 21 (2).
- [13] Shekhar C., Moisan S., Vincent R., Burlina P. & Chellappa R.: Knowledge-based control of vision systems. Image and Vision Computing. 1999. Vol. 17.
- [14] Sridharan M., Wyatt J. & Dearden R.: HiPPo: Hierarchical POMDPs for Planning Information Processing and Sensing Actions on a Robot. In Proceedings of the International Conference on Automated Planning and Scheduling. 2008.
- [15] Sridharan M., Dearden R. & Wyatt J.: E-HiPPo: Extensions to Hierarchical POMDP-based Visual Planning on a Robot. In the 27th PlanSIG Workshop. 2008.

10

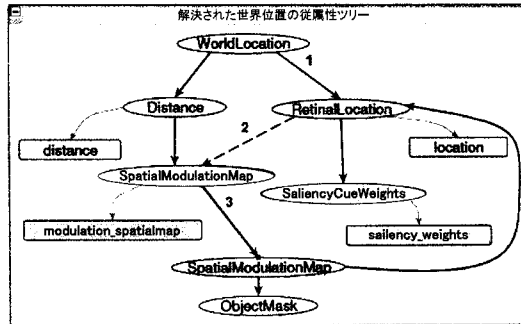
【 図 2 】



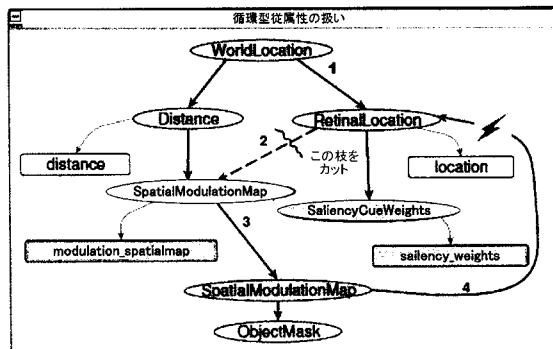
【 図 3 】



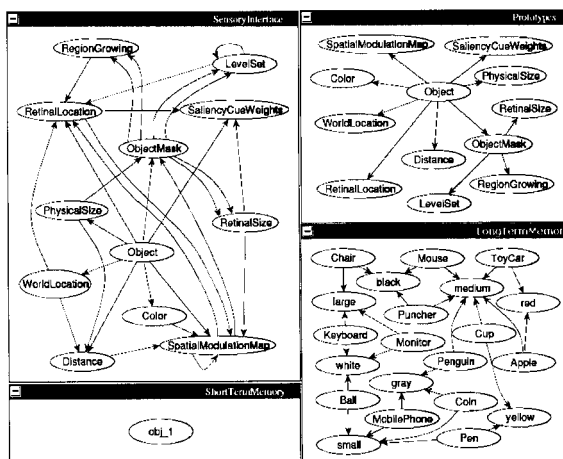
【図 5】



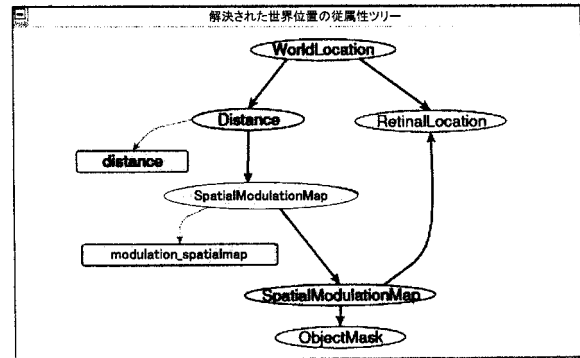
【図 6】



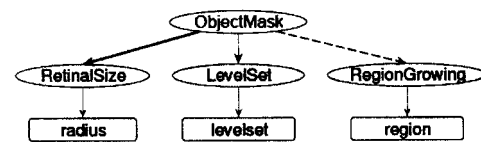
【図 9】



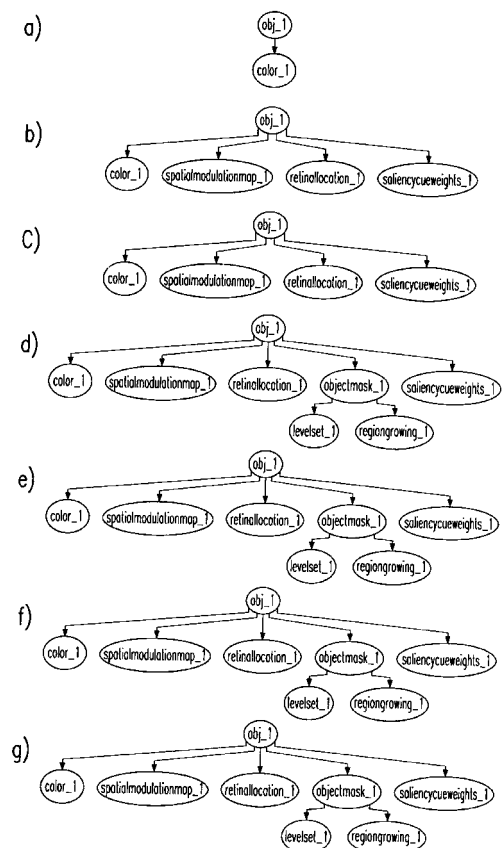
【図 7】



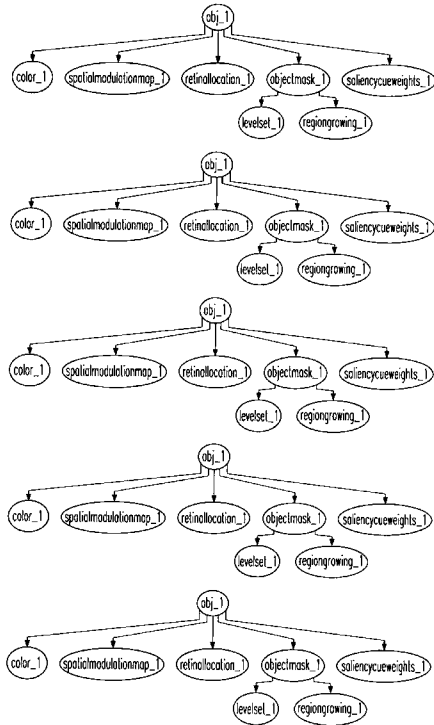
【図 8】



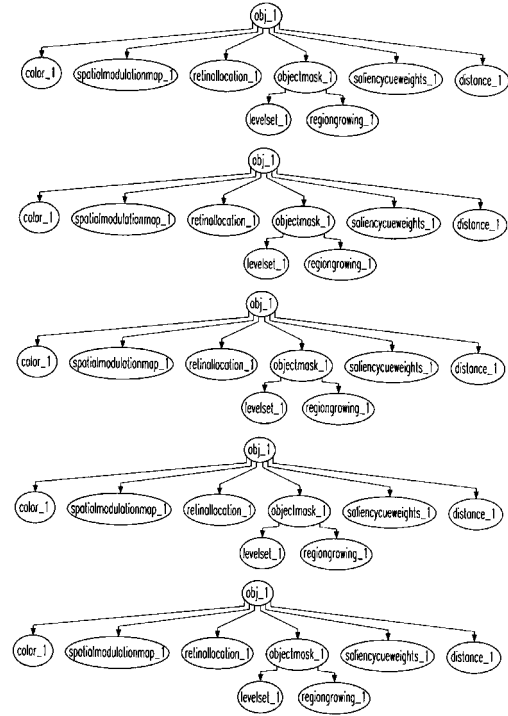
【図 10】



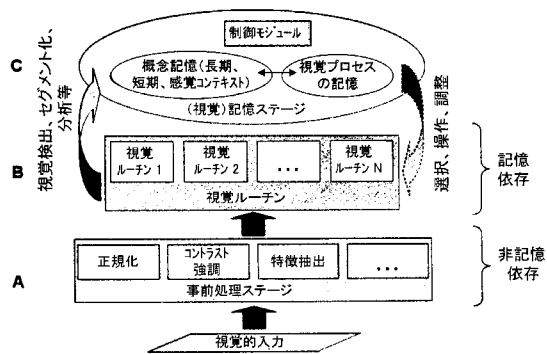
【図 1 1】



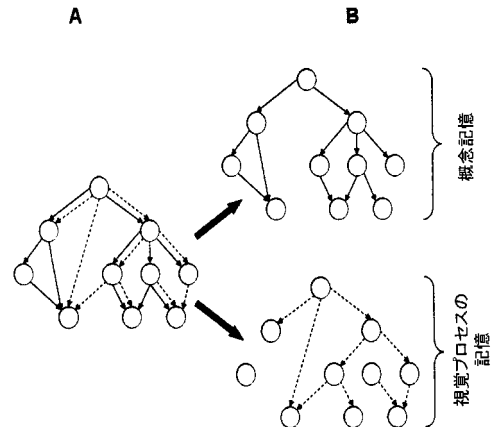
【図 1 2】



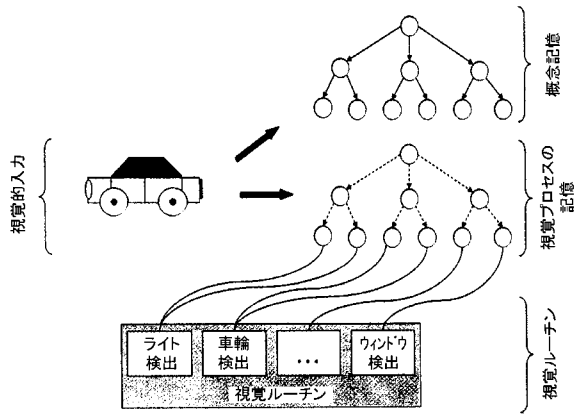
【図 1 3】



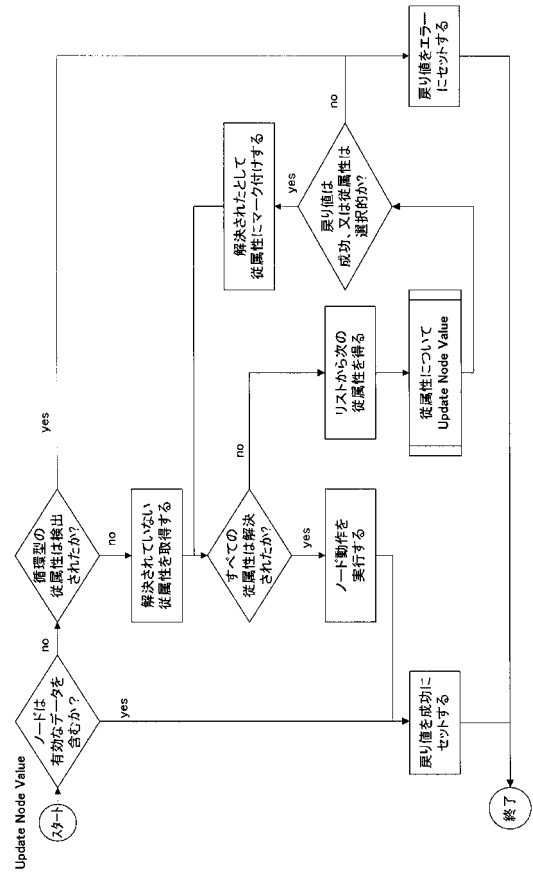
【図 1 4】



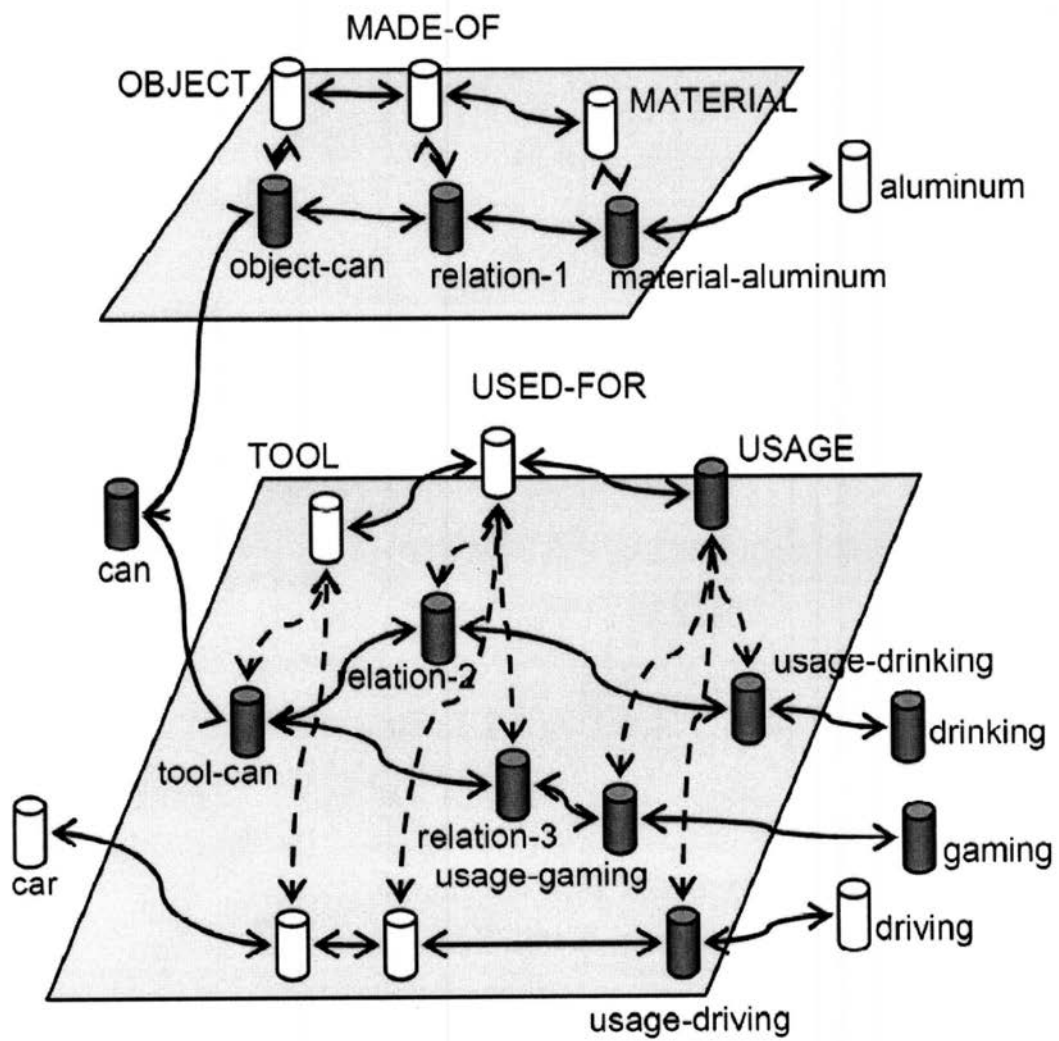
【図 15】



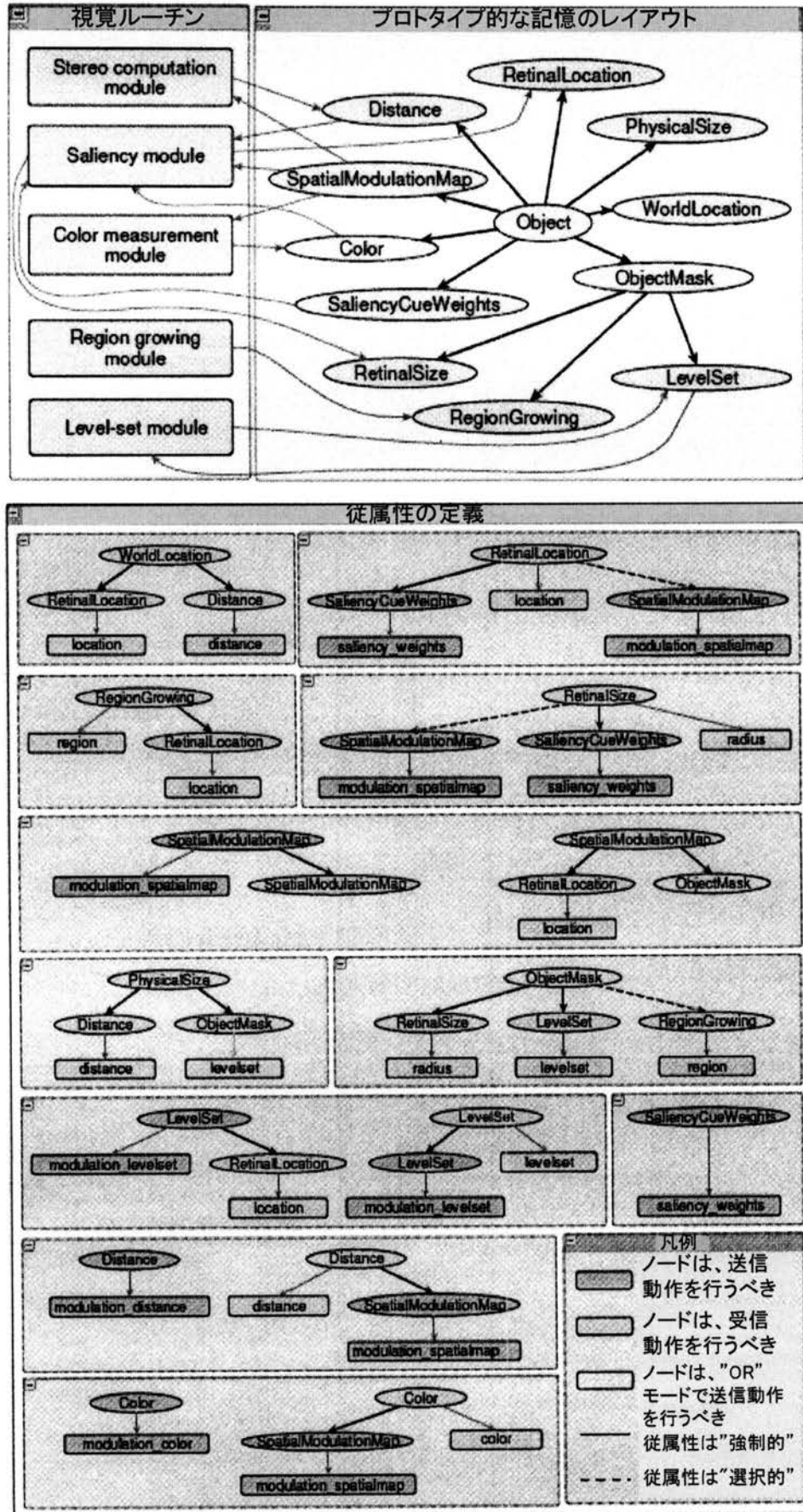
【図 16】



【図1】



【図4】



フロントページの続き

(72)発明者 スベン・レブハン

ドイツ国 6 3 1 7 9 オーベルトシャウゼン、ミュールフェルドシュトラッセ 3

審査官 新井 寛

(56)参考文献 特開平 0 8 - 1 4 5 7 3 8 (J P , A)

Sven Rebhan, et al., Attention Modulation Using Short- and Long-Term Knowledge, Proceedings of the 6th international conference on computer vision systems (ICVS 2008), 2 0 0 8 年, p.151-160

Julian Eggert, et al., First steps towards an intentional vision system, Proceedings of the 5th international conference on computer vision systems (ICVS 2007), 2 0 0 7 年

(58)調査した分野(Int.Cl., D B 名)

G 0 6 N 5 / 0 4

G 0 6 T 7 / 0 0